AN INQUIRY INTO ARCHITECTURAL SPACE IN COMPUTATIONAL DESIGN PRACTICE

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Architecture

by
Kadir ÖZTÜRK

July 2023
İZMİR

We approve the thesis of **Kadir ÖZTÜRK**

Examining Committee Members:

Prof. Dr. Fehmi DOĞAN
Department of Architecture, İzmir Institute of Technology

Prof. Dr. Mustafa Emre İLAL
Department of Architecture, İzmir Institute of Technology

Prof. Dr. Mine ÖZKAR KABAKÇIOĞLU
Department of Architecture, İstanbul Technical University

Prof. Dr. Aslı Ceylan ÖNER
Faculty of Fine Arts and Design, İzmir University of Economics

Prof. Dr. Şebnem YALINAY ÇİNİCİ
Department of Architecture, İstanbul Bilgi University

20 July 2023

Prof. Dr. Fehmi DOĞAN
Supervisor, Department of Architecture,
İzmir Institute of Technology

Assoc. Prof. Dr. Altuğ KASALI
Co-Supervisor, Department of Architecture, İzmir Institute of Technology

Prof. Dr. Koray KORKMAZ
Head of the Department of Architecture

Prof. Dr. Mehtap EANES
Dean of the Graduate School of Engineering and Sciences

ACKNOWLEDGMENTS

It is impossible to mention all of them, but I would like to remember some people here who contributed to the emergence of this thesis.

First, I would like to thank Fehmi DOĞAN for his influential guidance, optimistic persistency, and sincere friendship. This thesis would not have been possible without his leadership, patience, and support. He will keep his place in my life as a role model and I will try to remember the values he represents throughout my life.

I am also grateful to Altuğ KASALI for his insightful guidance and motivation throughout my research. His dignified and comprehensive way of thinking made many difficulties easy within his subtle sense of humor. I am thankful to his insights, critics, and feedbacks especially on research methods. It was an unequaled opportunity to work with him in this research.

I am indebted to Mustafa Emre İLAL whose careful and precise detections strengthens thesis monitoring process. He always contributed my research with his valuable interpretations and initiative visions. I am fortunately have taken the first steps in programming and coding with him.

I would also express my gratitude to Mine ÖZKAR KABAKÇIOĞLU, her unique experience and vast knowledge in computational design thinking and theory gave me confidence throughout the thesis monitoring process. She gave always progressive and enlightening criticisms which are always took research one step further.

I was honored to have two other inspiring academicians on my committee. I appreciate Aslı Ceylan ÖNER whose valuable comments and directions which I will definitely keep in my mind for my future studies.

I am fortunate to have Şebnem YALINAY ÇİNİCİ who accepted kindly to be on my committee. The ideas she introduced of Arch 574 at Bilgi University unconsciously prepared me for this journey.

I would like to express my gratitude also to all colleagues, teachers, and academicians whose names I can't mention one by one, but from whom I learned a lot while walking on this path. Another special thanks to the Design Cognition Research Group for the friendship, motivation and team feeling. I would like to thank also my thesis colleagues, Livanur and Batuhan, who patiently answered and helped me with all my questions, no matter how nonsense.

I would like to acknowledge all the interviewees who joint this research from around the world. This thesis came about thanks to their sharing of unique experience gained throughout their professional life. Thank you again for the precious time you have spent.

I would like to thank the Izmir Metropolitan Municipality, the institution where I worked professionally during the preparation of this thesis, and its valuable employees, especially my roommates Gizem, Alkın and Özden.

Also, I would like to acknowledge the help of my wife's parents and my relatives ÖZTÜRK, ERTÜRK, KIROVA, and VATAF families. They made things possible by supporting me many times during my thesis projects.

Finally, I express my profound gratitude to my parents Sibel and Mustafa ÖZTÜRK. I always felt they would be there whenever I needed them. Special thanks to my sister Selda who always trust me and always cheer me up with her unique positive point of view.

Words cannot express my gratitude to my dear wife Senem. She always kept me fresh and whenever I fell into despair, stood me up. Her efforts that keep children and family together are priceless. This dissertation would not have been possible without her love and patience.

This thesis is dedicated to my dear daughters Güneş and Deniz.

ABSTRACT

AN INQUIRY INTO ARCHITECTURAL SPACE IN COMPUTATIONAL DESIGN PRACTICE

Computational design, an overarching theme pervading architectural practice and research, has instigated profound transformations in our perception of space. Although computational design practices have gained significant prominence in architectural production, it is essential to thoroughly analyze their consequences on our understanding and imaginations of space. This research endeavors to explore the alterations that have occurred in our conceptualization of space within computational design practices, while also scrutinizing what aspects are retained, adapted, or disregarded from our understanding of space. This study seeks to comprehend how designers conceptualize space by conducting interviews with experienced practitioners with a specific research agenda in mind. The resulting interview data is subjected to qualitative thematic analysis. The research focuses on two key inquiries: How architectural space is imagined in computational design practice? How is architectural space considered as a multi-layered, loosely defined, and ineffable concept in computational design approaches in which procedures are precise, descriptive, and prescriptive? The infiltration of computational terminology into architectural discourse appears to exert influence not only on designers' verbal expression but also on their cognitive processes concerning architectural space. It is argued that computational tools impact designers' thought processes and spatial imagination, as well as the parameters they consider. The underlying assumption is that space, as a multi-faceted and multi-significant concept, encompasses both computable and non-computable aspects, warranting exploration into how computational practice accommodates these dimensions. While computation may offer as a appropriate medium for contemplating and representing the quantifiable aspects of space, the ineffable and ambiguous aspects emphasizes the limitations of computability. Findings imply that the topic of space and its relationship with computational design tools is not a primary focus or a significant area of attention in the current context.

ÖZET

HESAPLAMALI TASARIM PRATİKLERİNDE MİMARİ MEKAN ÜZERİNE BİR ARAŞTIRMA

Çağdaş mimari uygulama ve araştırmalarında başat temalardan biri olan sayısal tasarım, mekana ilişkin düşünme biçimimizde köklü değişikliklere yol açmıştır. Sayısal tasarım uygulamaları, mimarlık üretiminde giderek daha baskın hale gelse de mekan hakkında düşünme ve hayal etme biçimimizdeki yansımalarına yönelik derinlemesine çalışmalar hala gerekmektedir. Bu araştırma, sayısal tasarım uygulamaları ile birlikte mekanı kavramsallaştırma biçimimizde yaşanan değişiklikleri sorgulayarak modern mekan anlayışımıza ilişkin nelerin korunduğunu, uyarlandığını veya ihmal edildiğini tartışır. Sunulan çalışma ile sayısal tasarım araçlarını kullanan deneyimli uygulayıcılarla gerçekleştirilen görüşmeler üzerinden mekanla ilgili kavramsallaştırmaların farklı boyutlarının anlaşılması amaçlanmaktadır. Görüşmelerden elde edilen veriler nitel tematik inceleme yöntemiyle analiz edilmiştir. Bu araştırma iki ana soru etrafında şekillenmektedir. Hesaplamalı tasarım pratiğinde mimari mekan nasıl hayal edilir? Prosedürlerin kesin, tanımlayıcı ve kurala dayalı olduğu hesaplamalı tasarım yaklaşımlarında mimari mekan çok katmanlı, tanımlanması ve ifade edilmesi zor bir kavram olarak nasıl ele alınır?

Mimari kelime dağarcığına aktarılan sayısal terminoloji, tasarımcıların sadece sözlü ifadelerine değil, aynı zamanda mimari mekan hakkındaki düşüncelerine de hakim olmaktadır. Sayısal tasarım araçlarının, tasarımcıların göz önünde bulundurdukları parametrelerin yanı sıra mekanı hayal ederken onların düşünme biçimleri üzerinde de etkisi olduğu tartışılmaktadır. Buradaki varsayım, birden fazla katman ve anlam barındıran bir kavram olarak mekanın hesaplanabilir değişkenlerinin yanında hesaplanamaz yönleri de olduğu ve sayısal tasarım pratiğinin bu iki yönü nasıl ele aldığını araştırmanın keşfetmeye değer olduğudur. Sayısal düşünce, mekanın ölçülebilir yönlerini düşünmek ve temsil etmek için uygun bir ortam sağlarken, mekanın tarif edilemez ve belirsiz yönleri de hesaplanabilir olanın sınırlarını ortaya çıkarabilir. Mekan kavramının sayısal tasarım teknolojileriyle nasıl yeniden düşünüldüğüne ilişkin bulgular irdelenmeye değerdir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

Computational design has become one of the most popular topics in contemporary architecture. The use of computer technologies and computational thinking in architecture has been more pervasive since the 1960s and its transformative implications have yet to be fully discussed in architectural theory. When looked at from a broad perspective we are witnessing many changes in the practice in the way of architectural space is imagined, represented, and produced. For instance, architectural vocabulary is populated with terms borrowed from computational disciplines like parameters, algorithms, input/output structures, data oriented numerical models, scripts, codes (Coates, 2010). In addition, new ways of making of architecture like computer-numerically controlled (CNC) machines, robotic production and file-to-factory solutions has replaced conventional tools and methods. Besides, design as a way of thinking is exposed to a considerable transformation by computers and all designers individually deal with this transformation while composing their own cultures (Burry, 2011, p.8). However, some constitutive elements of architecture, especially the concept of space, have not been emphasized among computational design circles as it was earlier. Since space has been one of the main assets of architecture since at least Modernism, this research investigates changes and continuities in the understanding of the concept of the space within computational community.

The definition of space in architecture evokes notions from physical, mathematical, philosophical, and social theories. Scientific definitions of space are mostly based on theories of physics and mathematics. Space in philosophy considers cognitive and intangible dimensions. Yet, space is a constant value of architecture and different meanings are assigned in any period. It is an uncontested concept in architectural theory; space is also one of the most discussed ones. Different architectural movements with different perspectives have repeatedly defined space. Especially in Modernism, space has been the main tenet of architecture. Each definition highlights different values of space, which reflects the characteristics of contemporaneous architectural thinking

(Moholy Nagy, 1928). Space is a polysemic term that includes many meanings and interpretations.

Space has no single, fixed definition. It evolves continuously based on contemporary conditions. In this explorative research, it is aimed to elucidate the concept of space in the era of computational design. Here, the position is that space possesses both computable and non-computable features, and some of these features will be better represented through computational design tools. The choice of design tools significantly interacts with the representation and definition of space.

It is known that the design process is not linear and often open to unexpected surprises, but computational process is linear and actually based on precision and prediction. Computational procedures need to be explicit, transparent, and traceable in programming language. Space is embedded in and emerges from these algorithmic procedures and computable parameters in computational design practice. However, instead of being a constitutive tenet of the design process, space might have become a residue and an after-the-fact product in computational design approaches of contemporary architectural practice.

Architectural discipline as an intellectual activity had to convey ideas through visual representations. Representational systems are the tools of expressing designers' thoughts, which have transformed through history of art and architecture. These tools help transmit designers' ideas to the real world or even more strongly they help designers from their ideas. Every tool, every representational system we create simultaneously opens and closes different ways of thinking, which in turn define who we are. As often attributed to McLuhan (1967), "We shape our tools and then our tools shape us." The limits of production are the limits of representation tools, and the limits of representation tools are the limits of imagination.

In some instances, consideration of architecture only as tangible objects led to a lack of concern for space, which is more intangible than tangible. It is worth questioning how computational design today reckons with space as a multi-meaningful, undefinable, perpetual, and ineffable concept. The concept of space as an ephemeral concept has not yet been scrutinized in the computational paradigm. This inquiry will explore the concept of space as it is re-constituted in the paradigm of computational design practice. The assumption is that space has both computable and incomputable aspects and that it is worth exploring how computational practice treats both aspects.

1.2. Problem Definition

The terminology of computation, now being imported into to the architectural vocabulary, seems to dominate not only designers' verbal articulation but also designers' thinking over architectural space. It is argued that computational tools have an impact on designers' way of thinking while imagining space, alongside the parameters that they consider. The assumption here is that space as a multi-layered and multi-meaningful concept has both computable and non-computable aspects and that it is worth exploring how computational practice treats both aspects. While computation may provide an appropriate medium for thinking about and representing the quantifiable aspects of space, the ineffable and ambiguous aspects of space highlight the limits of what is computable.

1.3. Aim and Scope of the Study

This dissertation is focused on the conceptualization and imagination of the architectural space within computational design practice. For this purpose, the experiences and opinions of people who are authority in using computational design tools are consulted. The research employs qualitative research methods and follows the principles of the thematic analysis. In this context, possible participants were identified by conducting research focused on computational design practice. The questions were formulated to explore how participants approach and frame the concept of space in their own computational design practice. Interviews are conducted regarding architectural space and computational design practice. Interview questions are revised and refined throughout pilot interviews and feedback from a panel of academicians who experts in their field are. The data evaluated within the scope of the thesis is obtained from the written online remote interviews.

This survey was conducted with eighteen interviewees. Interviewees are selected from different professional experience levels of computational practice provided that they had completed at least one project with computational design tools. Each participant's unique architectural space approaches are examined by relating them to their computational design practice. It is revealed that, though they are experts of computational tools, their backgrounds, thinking approaches, and utilization of design tools leads to different inferences regarding their considerations of architectural space.

In this research, findings related to architectural space in computational design practice are presented in six chapters including the introduction. Chapter 2 presents the literature review and describe the background of computational thinking and architectural space. Design and the methodology of the research is explained in Chapter 3 by expressing the changes and refinements in the research process. Emerging concepts, themes, clusters, and categories are reported in the results as Chapter 4. In Chapter 5 revealing categories are related and discussed by relating them with practice and literature. It is concluded in Chapter 6 via summarizing the findings of this dissertation and recommending future works.

1.4. Research Questions

This research is shaped around two main questions:

1. How architectural space is imagined in computational design practice?

2. How architectural space is considered as a multi-layered, loosely defined, and ineffable concept in computational design approaches in which procedures are precise, descriptive, and prescriptive?

The first question is oriented to reveal the processes, factors and actions that related to the conceptualization and imagination of architectural space in computational design practice. The second question investigates the consistent and contradictory points while evaluating the concepts of space and computation from an architectural point of view.

CHAPTER 2

LITERATURE REVIEW

The understanding and exploration of architectural space have undergone significant transformations in parallel to developments in computational design practice. This chapter reviews the literature of the diverse perspectives and conceptualizations of architectural space, delving into its constitution, different conceptual frameworks, computationally constituted manifestations, and its limits beyond computability. By considering the role of design tools in shaping our perception and manipulation of space, this review aims to provide a comprehensive understanding of the complexities and implications inherent in the integration of computational approaches within architectural practice. The purpose is to establish a foundational background that connects the key issues within the research scope.

2.1. Constitution of Architectural Space

Space as a fundamental concept of 19th century architecture is constituted consciously since 1890 (Forty, 2004) many times by philosophers, scientists, architects, sociologists, and by its inhabitants. Its constitution encompasses physical, mental, and social dimensions (Lefebvre, 1991). As an ephemeral and deliberated topic, space does not possess a single and uniform explanation, but rather encompasses many conceptions. This section examines the development of the contemporary concept of architectural space, tracing its origins from the late 19th century.

Forty in their Words and Buildings (2004) draws the history of constitution of the concept of space. They date back the use of the term space in the vocabulary of architecture through the end of the 19th century. According to Forty, architectural 'space' is originated in the work of two philosophers Gottfried Semper and Immanuel Kant. Semper's theory was based on the enclosure which is explained via Caribbean hut structure shown in Figure 1 (Semper, 2004, p.14). Forty claim that the enclosed space concept was based on Hegel's Aesthetics. Hegel's understanding of enclosure was a feature of architecture's purposiveness (Forty, 2004). Through the maturation period of Modernism, understanding of space as an enclosure has affected many architects including Adolf Loos, H. P. Berlage, Peter Behrens, and Camillo Sitte. Forty consider Kant as another important source in the constitution of the concept of space. Forty, from Kant's Critique of Pure Reason which published in 1781, quote "space is not an empirical concept which has been derived from outer experiences". Kant declares that space does not represent any property of things in themselves, nor does it represent them in their relation to one another. Space is thought as a priori, ego centrist and pure intuition in the mind (Forty, 2004).

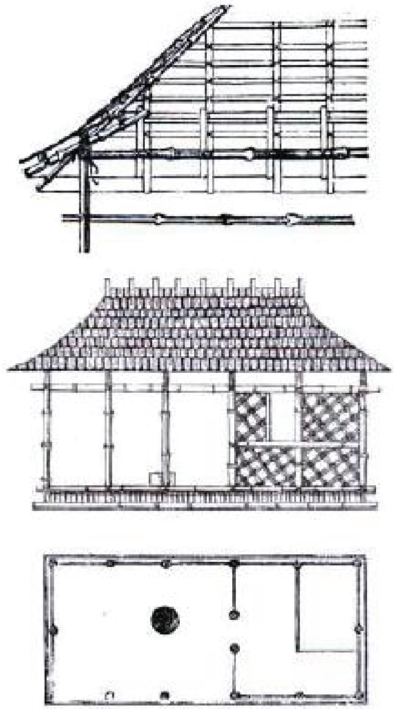

Figure 1. Semper's Caribbean hut.
(Source: Semper, 2004, p.14)

In addition, there is yet another approach offered by August Schmarsow (Schumacher, 2008). Schmarsow in his essay The Essence of Architectural Creation (1893) declared that the aesthetic of architecture does not lay in its material components. Schwarzer and Schmarsow (1991) distinguish Schmarsow's understanding of space from other theorists in highlighting the bodily movement rather than a stationary perception of form in the conception and experience of space, which constitutes the essence of architecture. Besides its scientific properties, Schmarsow developed a cultural perspective of vision. The origin of his theory of space, which suggests that 'spatial ideas are not a priori intuitions but the fusion of the impressions of bodily movements in mind', goes back to the empiricist theories of George Berkeley and William Wundt (Schwarzer & Schmarsow, 1991). Schmarsow departed from Kantian aesthetics and 'detranscendentalized the concept of architectural space in a cognitive process'. Spatial images were bounded on each other by the time through the movement.

Another theory that enforces the experience of space and Schmarsow's space concept is based on the theory of empathy. The theory of empathy is envisioned by Theodor Lipps (1897) drawing attention to the relationship between object and subject which creates pleasure when it is happened (Schwarzer & Schmarsow, 1991, p.53). Lipps defined empathy as an emerging pleasure at the juncture of "conscious mutual belonging between the soul and the thing being perceived".

Reminiscently, Le Corbusier in his Modulor points out the concept of ineffable space. The experience of ineffable space describes a point where body and mind meet while experiencing a specific emotion and individual understanding, which reminds the Lipps' concept of empathy. Lipps thought that there were two kinds of seeing (Forty, 2004). First one is optical seeing and this is concerned with matter. Second one is aesthetic seeing and this considers what was excluded after matter. Lipps defends that the spatial form can exist purely, unmaterialized as it is the case in the art of abstract representation of space (Forty, 2004).

The discussion could be extended to Dionysian and Apollonian dichotomy. Nietzsche suggest that the conflict between Apollonian and Dionysian spirits creates culture (Forty, 2004). Forty assigns Dionysian spirit the bodily understanding of space similar to Lipps' theory of empathy. However, Dionysian spirit of Nietzsche also glorifies the haptic senses of individual through the understanding of space.

Schmarsow's understanding of space developed a different point of view in the history of architecture and art. According to him, the history of architecture means the history of sense of space (Forty, 2004) and from ancient Egypt to Greece different historical periods of architecture could be reframed from this perspective (Schwarzer & Schmarsow, 1991, p.55). In addition, painting, relief, and sculpture which are visual creative arts, are all related to senses of the body. However, it is just architecture that spatial intuition emerges from the two essential sensations; first is the bodily inner values and second the visually outer values. Standing body senses, the three dimensions; first is the verticality by its height, second is the horizontality by its symmetry, third is the depth by its movement (Schwarzer & Schmarsow, 1991, p.54). It is how the experience of space as bodily extension constituted.

Space became a highly established term in the vocabulary of architecture through the 1920s. Forty (2004) indicates three points that the concept of space was accepted as the symbol of Modernism. First, new architecture of modern perception is identified with the concept of ‘spatiality’. Second, space as a new quasi purified concept of architecture, left all weights that concerning physical, philosophical, and historical backgrounds. Third, the concept of space elevated the ground of architectural labor through a mental activity in addition to its materiality. Later on, in the frame of Lefebvre’s tripartite conceptualization of space, spatiality became the core of architecture.

Lefebvre’s space is incorporated by the social actions of individuals and is immediately perceived (through the social relations of everyday life), conceived (by thought), and lived (as bodily experience) at the same time. Physical, mental, and social notions of Lefebvre’s space concept are inseparable, yet also have their own existence. ‘Space of architects’ (conceived - mental) becomes ‘architectural space’ (perceived - physical), and when it is experienced becomes ‘viable space’ (lived - social) that could reproduce itself. Lefebvre’s concept of space suggests that space is constituted mutually and consecutively by inhabitants/users, architects and by itself. According to Lefebvre, space is not pure, homogenous, and objective. Space is already contaminated by political impressions, modes of production, cultural issues, designer interferences, and/or inhabitant patterns. Adrian Forty mentioned critiques of Lefebvre’s against ‘abstract’ space from the points of unneutrality of given space, pre-learned qualities of educated architects, tool based non-objectivity of architects, priority of visuality in architecture, modernism’s homogenous and reductionist concept of space.

This study looked at space on an architectural scale, but there are also studies on an urban scale, which is another dimension. John Frazer, in collaboration with colleagues and students from the Architectural Association in London, was tasked with displaying the potential of an advanced, interactive, and adaptable computer model designed for a sustainable urban environment. This model aimed to empower the citizens of Groningen, allowing them to actively participate in and influence the development of their city. Prior to this initiative, Frazer introduced the concept of the "Intelligent Urban Tool" and emphasized the transformative nature of what he called "New Tools" in computing (Frazer, 1997). According to Frazer, these tools weren't just passive aids for tasks like drawing and modeling; they represented active tools that fostered innovative thinking and novel approaches, enabling what was previously impossible or inconceivable. The team developed a "what if?" planning model for Groningen, emphasizing urban-scale evolution and bridging different scales within the city. At the core of this model was the notion that computer programs could simulate real-life planning activities, facilitating a deeper understanding of suggestions and enabling proposals for potential future scenarios. Notably, this project initially employed a "computation without computers" mindset to shape its fundamental structure and objectives.

With such a diverse and rich debate on space and its significance in architectural design, this dissertation trace and explore how the concept of space is constituted in leading designers' computational design practice.

2.2. Different Concepts of Space

When Adrian Forty compares Heidegger and Lefebvre with respect to their concepts of space, he declares that "what both these writers make clear is that the space of which architects talk is not space in general, but an understanding of it quite specific to their own métier – it is a category invented for purpose of their own." (2004, p.275).

Laszlo Moholy-Nagy, known as one of the most influential and provocative figures of the Modernist era, presented his views on space in his work 'The New Vision.' He outlined a comprehensive understanding of space by describing forty-four adjectives that capture its various manifestations. According to Moholy-Nagy, space is not a standalone entity but rather a dynamic concept that is influenced by time and culture. It intersects with human cognition and is grounded in sensory experience (Moholy-Nagy, 1947). For Moholy-Nagy, sensory experience is a fundamental biological function that can be cultivated and developed. He emphasized that space is not a purely visual or intellectual construct but is grasped through the body and the senses. Sight, hearing, equilibrium, and the means of equilibrium were identified as the primary faculties through which space is perceived. Moholy-Nagy declared that the technically feasible hollow bodies could not compensate the exhilarating experience of articulated space. Moholy-Nagy keeps the space creation much more important than the other issues like financial, technical, structural, and social aspects. For him, the real architectonic conception is the space creation, and the dwelling means a life in a space that is a psychological need. Moholy-Nagy says that “the root of architecture lies in the mastery of the problem of space; its practical development lies in technological advance." (1928, p.60). Articulating masses, bodies and volumes are matters of classical architecture, but modern architecture articulates spaces with spatial patterns and originating relations between slabs and planes (Figure 2). Space creation is not volumes of building materials that are visible, measurable, and well-proportioned but is the nexus of spatial entities (Moholy Nagy, 1928). Continuity, fluid boundaries, and transparency are fundamental aspects of space creation, according to Moholy-Nagy. He emphasizes that space is an intangible, invisible, yet ever-present element in architecture.

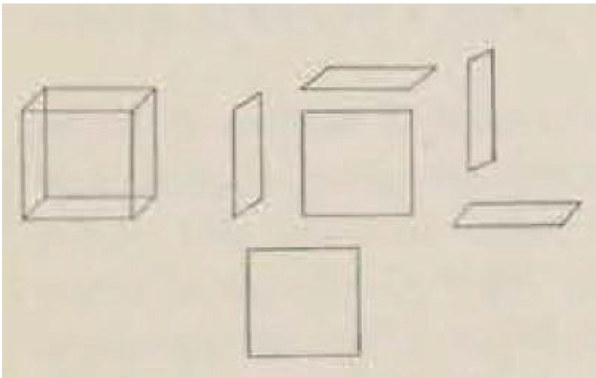

Figure 2. Volume and space relationships of Moholy-Nagy.
(Source: Moholy Nagy, 1928)

Adrian Forty grouped Moholy-Nagy’s space conceptions into roughly three different categories which are space as enclosure, space as continuum, and space as extension of the body. ‘Space as enclosure’ was established by Semper, developed by Berlage and Behrens, eventually incorporated in the volumetric Raumplan of Loos. ‘Space as continuum’ concept was appreciated by some groups from De Stijl to Bauhaus according to which inside and outside is thought as a whole continuous infinite continuity.

‘Space as extension of the body’ concept is conceptualized by Schmarsow that means body’s imagined extension within a volume.

Semper’s ‘space as enclosure’, Schmarsow’s ‘space as bodily extension’, Lipps’ ‘space as empathic pleasure’ and Lefebvre’s ‘space as social product’ are referred above. In conclusion, the exploration of different concepts of space reveals that the architectural understanding of space is not a general concept, but a specific construct tailored to the needs of the architectural approaches. From the late 19th century onwards, the term "space" in architecture has evolved through various perspectives, including enclosure, continuum, and extension of the body. These concepts, as exemplified by Semper, Schmarsow, Lipps, and Lefebvre, provide distinct frames for understanding space within architectural discourse. By recognizing these diverse conceptions, we could gain a deeper understanding of the multifaceted nature of architectural space and the ways it is shaped by different architectural way of thinking.

2.3. Space as Computationally Constituted

The main focal point of this dissertation is contemporary designers’, specialized in computational design tools, way of thinking on the concept of space. Computational design thinking is considered as a paradigm shift and the new thinking system of our era (Perez-Gomez & Pelletier, 1992). It is encouraged to think that this shift is a revolution, and the computational forms of these methods are technologically, formally, and conceptually divorced from architecture’s past (Keller, 2012). Pervasive nature of this kind of thinking requires reconsidering the architecture discipline (Bundy, 2007). Moreover, some has declared that computational design is the mainstream of architecture from now on (Schumacher, 2009). Computational design thinking caused radical alterations in the conception of space, representation of space, and image of space in our mind (Burry, 2011, p.8). Sean Keller states that there is a genius in artists to possess the complexity and the unity of nature. According to him, the role of the genius is replaced by algorithm, therefore the architect has lost his/her role (2012, p.47). As the argument goes, computational technology has decreased the intense relationship between design and designer (Keller, 2012, p.47). By rendering the object of architectural design as a matter of algorithms, computational design naturally quantifies, solves, and computes space using parameters. However, the concept of space is not given sufficient emphasis as the primary constituent of architecture.

William Mitchell (1990), in his Logic of Architecture, presented the fundamental relations of design, computation, and architecture without bypassing the concept of space. Although many definitions have been brought to the concept of computation, Mitchell's definition, which expresses “design as computation”, is one of the simplest and accepted. Mitchell states:

> “The process of finding a solution to a design problem is a trial-and-error one of applying rules to generate candidate solutions, then computing predicates to determine whether candidate solutions are acceptable solutions.” (p.179)

Computational design thinking is not a homogenous strategy and is referred to in ‘a variety of labels like algorithmic, parametric, genetic, nonlinear, nonstandard, intricate, animate, blob etc.’ but their commonality is ‘altering the architect’s authorial role and the understanding of designing in singular and static manner.’ (Keller, 2012, p.42). There are also some other classifications and different fractions of computational design.

Rivka Oxman (2006), for instance, groups the digital design models according to “various relationships between the designer, his/her conceptual content, the design processes applied, and the design object itself” (2006, p.245). Another approach is provided by Toni Kotnik (2010) who classifies “contemporary digital design methods and understanding of their logical relationship by their levels of digital computability” (2010, p.2).

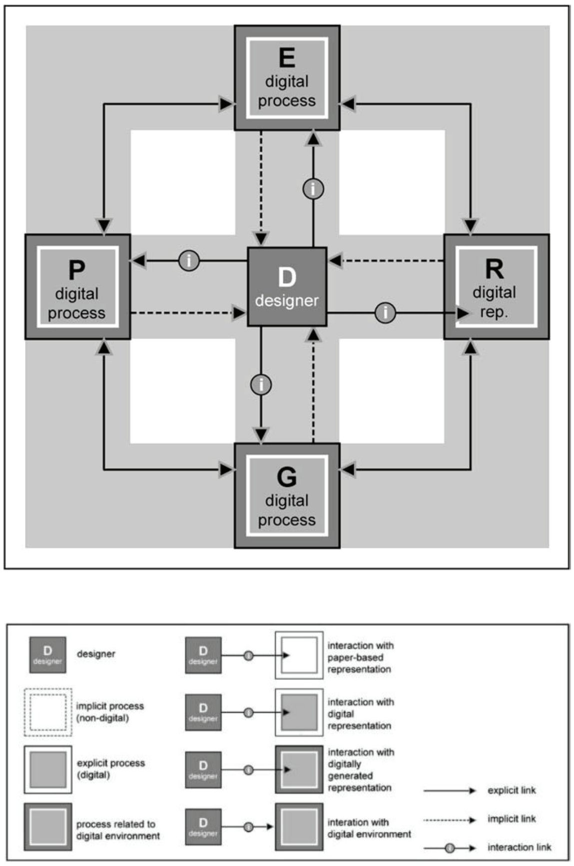

Figure 3. Oxman’s schema: symbols, boundaries, links, and compound model.
(Source: (Oxman, 2006)

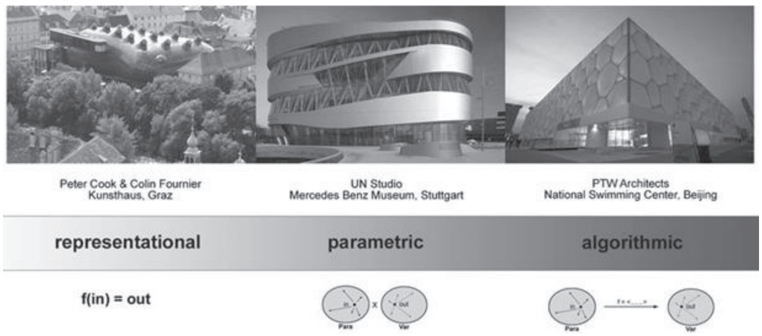

Figure 4. Kotnik’s diagram of levels of digital computability.
(Source: Kotnik, 2010)

Knight and Stiny (2001) distinguish computational design as classical / non-classical along two aspects of computation that are representation and process. It differentiates the ways in which computation could be addressed in the discipline of architecture.

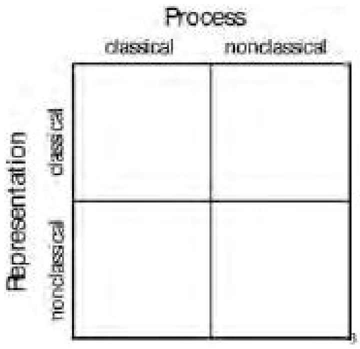

Figure 5. Knight and Stiny's categorization of computation.
(Source: Knight and Stiny, 2001)

The computational design approaches, encompassing various labels and classifications, have fundamentally altered the perception, representation, and understanding of space in the minds of designers. While parameters and algorithms have transformed the architectural design process, the concept of space itself has received relatively less attention as a vital constituent of architectural practice within computational design. The computational design approach, with its diverse strategies and perspectives, challenges the traditional authorial role of the architect and embraces a dynamic and non-static approach to designing. By examining the relationships between designers, conceptual content, design processes, and the design object itself, as well as considering computability, a deeper understanding of computational design and its implications for architectural practice can be acquired.

2.3.1. Space as Mathematics

Although, the impact of computation is extensively scrutinized in architecture, the effects of mathematics in computational revolution are not obviously seen and discussed (Legendre, 2011). Legendre (2011) reminds that architects, designers, historians, and engineers have common roots, so they think and practice in similar ways but throughout the digital design era the impact of mathematics is neglected. Constitution of space in architecture deeply correlates with the developments in the mathematics. Mathematics is 'shifted and fluctuated' in history but in any condition, it was a key to understand the sensible world (Dahan-Dalmedico, 2011, p.19). From the ancient mathematics of immobile objects to the early calculus studies of the 17th century by Descartes and Newton, mathematics has evolved into a dynamic and generative discipline in the 21st century. It has transitioned from mere observation and calculation to active emergence and computation. The mathematical developments lead to groundbreaking advancements in various fields.

In reference to Picon, Legendre (2011) asserts that there is an estrangement between architecture and mathematics, which happened when arithmetic and geometry were displaced by the calculus. According to Picon (2011), using geometry and arithmetic instead of calculus create stronger and deeper spatial intuition in designer's mind. The new world of calculus is "no longer provided averages but firm boundaries" and "mathematics was about setting limits" to "design regardless of its fundamental intuitions" (Picon, 2011, p.34). Computer gave another chance to architecture to reconcile with calculus by facilitating geometric flows, transform forms. As Forty declares that "mathematics appears neither as foundational nor as tools". There are two reasons for this, first that mathematics is veiled behind the computer screen and the second is that designers deal with relations, parameters, scripts, and algorithms abstracted from spatial intuition (Picon, 2011). Picon finally declares that mathematics and architecture should come together ethically "when to use power and when to adopt restraint" (2011, p.35).

2.3.2. Space as Human-Centric

In the realm of exploring space as human-centric, Derix merged the notions of space as Schmarsow's "impressions of bodily movements in the mind", Lipps' "empathy theory," and Chomskian "linguistic based theories." This amalgamation of ideas emphasizes the significance of understanding space from a human perspective, considering how it is perceived, experienced, and interpreted by individuals. By incorporating these diverse perspectives, Derix introduces human-centered approaches in thinking of space. (Derix, 2014b). Especially Coates' applied algorithmic studies, which views architects as system designers, and Hillier's space syntax studies, which are based on visual senses and spatial relationship diagrams, initiated how computation and practice could be brought together. Some designers constituted their understanding of space on the basis of occupant/user-oriented studies. Two of them are Christian Derix and Asmund Izaki (2014), who study on developing a design strategy that uses computation "as a vehicle to embed human-centric concerns in spatial systems" (Derix, 2014a, p.7). Occupant/user-oriented studies' main concern is to integrate user experiences (Izaki & Helme, 2014) and behaviors (Derix & Izaki, 2014) in the early design stages even as "a design driver" impulse (Derix & Jagannath, 2014). Derix criticizes the contemporary design computation's automation and optimization. He claims that this approach resembles the machine analogy of Modernist thinking (Derix, 2014b, p.15). According to him, qualitative user-centric spatial issues and user behaviors will become more important for the future of the computation agenda. In addition, Izaki and Helme present a project of Computational Design Research group at Aedas|R&D in which specifically developed algorithms are used "to calculate, visualize and stimulate human-centric architectural conditions". Three topics considered throughout the project are "movement and networks, visibility and space, and behavior and experience" (Izaki & Helme, 2014).

However, human-centric computational design strategy creates a different constitution of space, Leon van Schaik (2014) raises concerns in relation to the overemphasis of computation that could alienate and distance architects from spatial qualities in design. Schaik's research underlines the relationship between coding, spatial awareness, and the architectural design process. He explores how architects shape their mental space to foster creativity and drive intellectual progress in their designs. According to van Schaik, despite the widespread discussion on this topic, there is a need for concrete evidence to substantiate the claim that coding can assist architects in gaining insights into their personal spatial experiences (van Schaik, 2014, p.141).

2.3.3. Space as Generative, Evolutionary, and Responsive

Schwarzer (Schwarzer & Schmarsow, 1991) interpreted Schmarsow's concept of space "as living amalgamation of human impulses, created perceptually by its creator and its users" (1991, p.55). This interpretation basically reminds the generative, evolutionary, and responsive attitudes of contemporary computational design practice. Designers do not design end-products, they design systems that will generate end-product(s). They prefer to focus on inner logic instead of external form. Gordon Pask (1969) refines this point and accepts architects as first cyberneticians because of their profession's focus on organization of abstract systems.

John Frazer in his book 'Evolutionary Architecture' suggests that the nature is an inspiring analogy for architecture with its forms, structures, and inner logic of morphological processes (1995, p.10). Frazer looks for "a coded set of responsive instructions” that could conceptualize the “genetic language of architecture”. Frazer quoting Lionel March “logic has interests in abstract forms. Science investigates extant forms. Design initiates novel forms. A scientific hypothesis is not the same thing as a design hypothesis. A logical proposition is not to be mistaken for a design proposal” (1995, p.12). Frazer declares they are not after a theory of explanation but a theory of generation (1995, p.12). Frazer criticizes W. J. Mitchell’s generative approaches because of endless potential solutions and unmanageable quantity of permutations possibilities (1995, p.16).

Frazer also reminds us Nicholas Negroponte’s idea of ‘Soft Architecture’, which is an evolutionary design process (1995, p.17). It is a responsive system includes hardware and artificially intelligent software. Negroponte proposed the idea of presenting the design process, which he viewed as evolutionary, to a machine also capable of evolution. This mutual interaction between the design process and the machine was intended to result in training, resilience, and growth. Frazer points out the Negroponte's hopes for computer hardware and later for software with artificial intelligence, but neither of these elements yielded the desired outcomes or solutions in the circumstances of the day (1995, p.17).

Gordon Pask in his ‘The Architectural Relevance of Cybernetics’ (Pask, 1969) also states that “responsible architect must be concerned with evolutionary properties; architectural designs should have rules for evolution built into them if their growth is to be healthy rather than cancerous”. Cybernetics with an appreciable predictive power is deeply integrated with responsive environments.

The earliest and the most inspiring project of this line of thinking in architecture is Cedric Price’s Fun Palace, started in 1962 but never finished (Mathews, 2006, p.39). Fun Palace was as a social experiment in new ways of building, thinking and being. It was shaped in the social and political context of 1960s and was technically supported by cybernetician Gordon Pask. Fun Palace was a living experiment for cybernetics theory because its design concept was based on “the inevitability of change, chance, and indeterminacy by incorporating uncertainties as integral to a continuously evolving process modeled after self-regulating organic processes and computer codes” (Mathews, 2006, p.40). Mathews specified that Price considered spaces and events differently, so

Price’s understanding of architecture is parallel with Heidegger’s concept of space underlying “the site of human activity and meaning rather than as structure or enclosure”. Price, Littlewood, and Pask thought Fun Palace as permitting multiple events, adapting to change, learning, and planning future activities with biological, social, and mechanical systems but these powerful attempts just concretized in architectural discourse.

2.3.4. Space as Language, Grammar, and Structure

Constitution of space is also considered through linguistic terms. Linguists have a different kind of 'systems view'. Some of them establish their theoretical structure on texts and algorithms, some of them on shapes as a base unit. In any way, linguists structure a system of rules that outputs end product(s).

Bill Hillier’s theory of space syntax, for instance, is derived from basically the theory of linguistics. Hillier thought that buildings are spatial configurations – machines – rather than physical matters that absorb and generate social information by their configurations (Forty, 2004). Space syntax is a configurational analysis tool that detects patterns and quantitative expressions of space in case to answer the question of “how things are put together” (Hillier, 2007, p.1). According to Hillier, human mind handle configuration unconsciously and intuitively like it handles the grammatical and semantic structures of language intuitively (2007, p.3). Hillier, in his 'Space is the machine', states that architecture is analogous to language because both of them have conceptualized meanings, a set out vocabulary, syntactic rules, and grammar. The structures of language and architecture restrict the combinatorial possibilities and these restrictions construct meaningful outputs (2007, p.8). He adds that there are limits of utterable statements and within these limits creativity and individuality emerge (Hillier, 2007).

Paul Coates in his 'programming. architecture' summed up Chomsky's generative grammar theory as parsing and breaking down of the text(s) until to the basic phonemes of the utterance that also could be inverted from atomic particles by combining them in a correct sentence (Coates, 2010, p.2). Generative grammars could produce syntactically correct sentences, but its syntactical correctness does not guarantee semantic values. On contrary, syntactically incorrect sentences may have meaningful statements (Coates, 2010). Coates claims that using generative grammars helps to structure deeper relations with extended designs / ideas and provide flexible and abstract forms of representation.

There are two basic advantages of language, first is the capability of "generating an infinite number of syntactically correct sentences" and the second is the availability of recursively parsing that allows "encapsulation, multiple branching and large combinatorial possibilities" (Coates, 2010, p.3). According to Coates, there are natural and artificial languages. Natural languages have anonymous syntax sources, and its lexicon is naturally developed. In artificial languages, on the contrary, syntax is explicit, and lexicon is well-defined. Computer code is an algorithm based artificial language. Coates references Kittler who distinguishes computer code as "the only text that can read itself" (2010, p.3). It is accepted that "the computer languages are based on the fundamental operations of the Turing machine". A piece of data can be an instruction, and an instruction can generate a piece of data in Turing machines (Coates, 2010, p.3). Coates states that the inbuilt reflexivity of computers can bootstrap themselves by using code. His main point of view is that it is possible to generate large set of algorithms by coding which extends the limits of research space with huge amounts of output capacity.

In his fundamental book 'Shape: Talking about Seeing and Doing' George Stiny (2006) intentionally use of calculation instead of computation to constitute a theoretical ground in digital design theories. Main concern of his discourse is to explain designing and calculating as similar activities. The reason why he prefers calculating rather than computation relates to the fact that computation intuitively suggests the use of computers, which is in itself restrictive given the fact that computation in general could be possible without computers as well. Stiny considers calculating as more down to earth, ordinary, and suitable for art and design (2006, p.6).

Stiny clarifies the division between ordinary and visual calculating as follows. He adds one square to another mathematically, which equals to two squares but in visual computation they do not add up to two. This relates to peculiarities of seeing; the sum of two squares visually could be many things including two L shapes, three squares, or four triangles as presented below Figure 6. Stiny calls this embedding (2006, p.8) which includes loops of seeing and doing over shapes. He says that points, lines, curves, planes etc. are constituents of shapes (Stiny, 2006). According to him, designing starts with seeing because when the constituents of shapes fuse and generate new shapes it means that the designer will see it with a different point of view. However, Stiny specifically mentions the disturbance of aggrandizing of making in art and architecture in his conversation (Stiny & Gün, 2012). He is actually distant to the usage of computers as a representation tool or a building performance analysis and simulation tool. The use of computers as a representation tool sounds annoying and misses the core of design for him. His point is that even though it is a visual and calculable process, parametric design is missing the intelligence of seeing resembling a visual kick of sponge with a defined colors, components, and combination (Stiny & Gün, 2012, p.10).

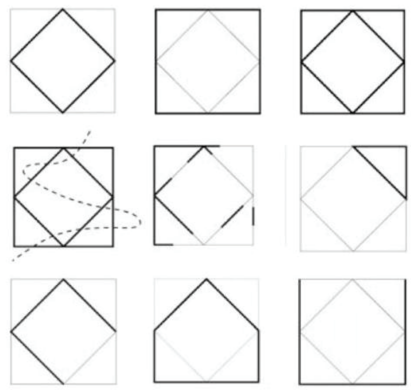

Figure 6. Knight and Stiny's visual embedding diagrams.
(Source: Stiny, 2006)

The acceptance that individual interrelations between the world and the mind are established through language has led to philosophical inquiries (Aysever, 2003, p.127). Wittgenstein presented two aphorisms that pointed out the limitations and incapability of language. Firstly, he stated that if we cannot speak about the unsayable, then we must remain silent. Secondly, he asserted that the boundaries of our world and mind are defined by our language. According to Wittgenstein, there are aspects of the world that go beyond what can be expressed in language (Aruoba, 2006). However, acknowledging the limitations of language does not imply that all sayable things have been said and that language is incapable of going beyond them (Güvenç, 2013). Ultimately, if there are limits to what can be said, there may also be limits to what can be computed. It can be inferred that abstract and subjective aspects such as aesthetics, beauty, emotion, social values, experience, and more, could be considered as beyond the realm of computability when evaluating the composition of space.

Inquiring into the indescribability/indefinability of space within a computational approach opens up new perspectives for understanding the topology of design space through the programming of artificial languages. Merely adhering to grammatical correctness does not guarantee meaningfulness in architecture or language. Meaningfulness is of greater importance alongside syntactic accuracy. Considering an artifact as meaningful extends beyond adherence to grammatical rules alone. Computational design techniques generate numerous artifacts as potential design solutions that conform to the structured rules of generative systems, from which the designer selects one to further develop. However, instead of focusing solely on generating and selecting random solutions from a limited search space, it is valuable to engage in inquiry. Contemporary computational design practices should aim not only to express but also to comprehend. By understanding the constitutive components of space, architecturally meaningful artifacts can be produced.

Computational thinking, rooted in computer science, establishes communication between computers and humans through artificial programming languages. Similarly, to natural languages, programming languages have their own grammar rules. However, recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have started to blur the distinction between natural and artificial languages. This blurring of boundaries has paved the way for computational design approaches, allowing the analysis of the constitutive elements of architectural space within the realm of computational thinking.

In the context of understanding architectural space, natural languages possess a metaphorical and nonlinear semantic richness that allows for complex interpretations. On the other hand, programming languages used in computers are relatively younger, more standardized, and adhere to strict grammar rules. Despite this, artificial programming languages possess high processing capacities due to advancements in AI and NLP. There is potential for advancements and alterations in the understanding of architectural space by leveraging these capabilities.

Through computational design approaches, the application of artificial languages and intelligence could offer new perspectives and insights into architectural space. The analysis of architectural elements using computational thinking allows for a systematic exploration of spatial relationships, proportions, and configurations. It provides a structured framework for understanding and manipulating architectural space, potentially leading to innovative design solutions and a deeper comprehension of the impact of spatial arrangements on human experiences. Therefore, the integration of artificial languages and intelligence in architectural practice has the potential to reshape our understanding and exploration of architectural space.

2.4. Space as Beyond Computable

There are many discourses in the architectural literature about the dimensions of space beyond the computable, which actually built the sanctity of architectural space throughout Modernism. Almost all the leading figures that come to mind when Modern Architecture is mentioned have developed discourses that make the architectural space otherworldly.

Le Corbusier (1948), for instance, discusses the concept of "ineffable space" in his book "New World of Space”. He explores the notion of space and its experiential qualities, including the idea of ineffability—the idea that certain aspects of space are beyond verbal or visual description (Kiyak, 2008). His concept of ineffable space explains a moment of enlightenment that he experienced. He states that there was a wall at his home with certain lighting and dimensions where he was painting and hanging his works. Then, one day, at a very precise moment he saw what he calls the ineffable space that came into being before his eyes when the wall with its picture lost its limits and became boundless. Actually, it is an emotional and intellectual experience when feeling and conceiving space emerge in a creative way. The experience of ineffable space describes a point where body and mind meet while experiencing a specific emotion and understanding individually. Le Corbusier's thoughts on ineffable space and its importance in architectural design are as follows:

> “Then a fathomless depth gapes open, all walls are broken down, every other presence is put to flight, and the miracle of inexpressible space is achieved. I have not experienced the miracle of faith, but I have often known

the miracle of inexpressible space, the apotheosis of plastic emotion." (Corbusier, 2004, p.32)

Tobi Stöckli (1992) in his article 'The Measurable and the Unmeasurable or From Form to Design to Existence' quotes L. Kahn, another important character of Modern Architecture, to explain how design process is nonlinear and mystical. Kahn describes the design process with many steps and jumps that makes it nonlinear and discontinuous progress (Stöckli, 1992, p.274). The building comes into being through the relationship of the form, design, and existence through his cycle of unmeasurable and measurable. Stöckli quotes Kahn in describing the unmeasurable:

> "A great building must, in my opinion, begin with the unmeasurable, must go through the measurable in the process of design, but must again in the end be unmeasurable. The design, the making of things, is a measurable act. ... What is unmeasurable is the psychic spirit. The psyche is expressed by feeling, and also by thought, and I believe it will always remain unmeasurable..." (Kahn, 1930, p.11)

Despite the intellectual and formal conceptions that computational design thinking and tools have introduced to architecture, there are inherent challenges or unexplored dimensions that require further attention and investigation in order to fully grasp the concept of space within the realm of computational design. Not only unpredictability of social life parameters but also uniqueness of bodily spatiotemporal experiences makes it harder if not impossible the formulation of well-defined input values to be computed in computational process.

Norberg-Schulz and Loci's perspective highlights the distinction between the everyday life-world and the tools and abstractions we create to understand and manipulate it. They argue that while concepts like atoms, molecules, numbers, and data serve specific purposes in scientific and technological domains, they do not fully capture the essence of the lived experience. According to them, places are complex qualitative phenomena that cannot be reduced solely to their spatial relationships or analyzed through scientific concepts.

> 'Everything else, such as atoms and molecules, numbers, and all kinds of "data", are abstractions or tools which are constructed to serve other purposes than those of everyday life. Today it is common to give more importance to the tools than our life-world. … A place is therefore a qualitative, "total" phenomenon, which we cannot reduce to any of its properties, such as spatial relationships, without losing its concrete nature of sight. … Being qualitative totalities of a complex nature, places cannot be described by means of analytic, "scientific" concepts. As a matter of principle science "abstracts" from the given to arrive at neutral, "objective" knowledge. What is lost, however, is the everyday life-world, which ought to be the real concern of man in general and planners and architects in particular. Fortunately a way out of the impasse exists, that is the method known as phenomenology.' (Norberg-Schulz & Loci, 1980, p.6-8)

Norberg-Schulz and Loci criticize the tendency of modern society to prioritize tools and abstractions over the genuine concerns of human existence. It is suggested that by fixating on scientific abstractions, losing touch with the everyday life-world is risked. In their view, science tends to abstract from the concrete realities of lived experiences in order to attain objective knowledge, but in doing so, it overlooks the essence of our daily lives.

Computers as the main tool of computational design action are based on abstract logic systems whose closest ancestor is Turing machine which is a one-dimensional pseudo infinite cellular tape. Computers are fundamentally established on machine codes that execute instructions strictly. These instructions operate algorithmic procedures that are composed of input / output structures (Kotnik, 2010). Computability theory is a field in mathematics related with Entscheidungsproblem and halting problem (Brey & Søraker, 2009). When associated with computability concept, it could be speculated that the computational design process may not be capable of processing ill-defined or incomputable inputs as in architecture to generate outputs.

In addition to the mathematical debates on computability, the issue has also been discussed in the discipline of architecture. Parisi (2013) defines that there are aspects of thinking within the context of programming that cannot be fully understood or explained by traditional computational models, and these aspects deserve further attention and investigation. It is not possible to put into process and get an output from non-numerical, uncountable, and unparameterizable values in computational systems (Parisi, 2013, p.XVIII). Another dimension to discuss the computability in architectural design process is Lefebvre's approach that considers the social practice as an additional third fact which completes and reproduces space again and again as an alive generative concept (Lefebvre, 1991). Scientific capabilities of computational design approaches are still insufficient against the uncomputability / undecidability of spatial (spatiotemporal) experience. Besides computable and describable parameters of space, it should be remembered that there are many worldly and unworldly synesthetic values that could not be transformed into or represented easily (Frascari, 2011).

2.5. Considering Space via Design Tools

Marco Frascari (2011) in his "Eleven Exercises in the Art of Architectural Drawing" investigates drawing as the main tool of designers. Non-trivial drawings of early design process are his starting point to explain the interactive design process. He explores early design ideas and their transformation through design tools especially aggrandizing drawing which according to him enhances the imagination of designers. He emphasizes that unity between mind and hand is possible with the facture of drawing.

> "Architecture is not a work of art, but the art that makes the work. Its embodiment formulates factures, because architecture makes things, rather than accepting things that make architecture." (Frascari, 2011, p.11)

Frascari describes his work as not a counter-revolution, but he states that he aims to gain a critical understanding of the processes at the base of architectural design for better handling of computer technology. His points could be a start line to understand how spatial imagination could be traced through the design process to inquire the relationship between computational design tools and designers' spatial imagination. Frascari criticizes speedy, sterile, and meaningless drawings and sometimes directs his criticisms at the digital tools' effect on designers' imagination. He accepts that non-trivial architectural representations have not algorithmic or digital potentials, but they possess materiality, sensuality, and meaning of imagination that not any tools could trace.

Frascari considers digital products (renderings, drawings, images, etc.) as archeiropoietic (not-handmade) icons. For him, digital images legitimize themselves by other-worldliness with holy, authentic, and divine view. Through a metaphor between building and cookery, he suggests that ingredients, recipes, and personal eating experience need to come together for cooking by conceiving of ideas. He refers to Michael Polanyi as he explains that the knowledge of the theoretical physics of balance cannot replace the practical knowledge of how to ride a bicycle. Human knowing activity is not just a mental and abstract one but also a bodily and experiential process. Frascari quotes Polanyi that tacit knowledge includes combinations of many sensorial and conceptual information to deduce meaning, so people know more than they can tell.

Synesthesia is Frascari's other fundamental concept that he repeatedly mentions through explaining the design process by drawing. Synesthesia technically means the crossing of the senses; it is an emotional experience. Synesthete people could smell colors, see sounds and feel tastes bringing together reason and emotion. Frascari mentions that synesthetic perception supports interconnections between ineffable, invisible, and impalpable things that describable, but which are beyond words.

Frascari also explains the concepts of slowness and fastness in architecture in reference to fast-food vs. slow-food following his cooking metaphor. According to him, good architectural drawings mean slow food for architects' thinking but bad architectural drawings equal fast food. Kundera's novel La Lenteur is reminded by Frascari in which slowness means remembering and speed means forgetting. He explains slow architecture with festina lente:

> "Festina lente aims at a highly efficient and effective method for practicing long-term thinking within architecture. The notion of festina lente has major implications for the pace of drawing because it prevents individuals from acting in the heat of the moment and avoids swift and arbitrary decisions by fulfilling a slow building up of careful work." (Frascari, 2011, p.30)

Nontrivial drawings emerge when abstract ideas turn into representable fact, so this moment is a critical threshold. Frascari says that architect's drawings are continuous alternation between representable and not representable.

When drawing is investigated in depth, tools became primarily important. One of the eleven exercises of Frascari draws attention to awareness of tool. This exercise requires designers to design their own drawing tools. For him, tools are replicable, reusable, and encapsulate information and behavior. In addition, they can connect and interact with each other. He claims that tool-making would raise mind, body, and reflexive activity to a different level.

It is defined that mathematical abstract thinking transformed geometry into a mere algebraic and digital understanding (Frascari, 2011). Teaching geometry moved from the level of sensual perception to cerebral discernment. According to Frascari, this move from approximations to analytically intelligible precision risks the delightful architecture because every part and detail of architecture is all about approximation.

The next distinction that Frascari mentions is the typographic performance and chirographic culture. He describes chirographic culture with the traces of drawing, in direct physical interaction with work and material play. However, if the process goes with digital input and without any bodily integration, this is typographic progress. For him, typographic architecture performs with screens, mouse, keyboards which is described as harmful for architectural imagination.

Marco Frascari's approach needs to be considered before starting to evaluate architectural imagination and computational design tools. His exploration of architectural drawing as an interactive tool that enhances imagination and his critical examination of digital tools' impact points out the significance of non-trivial drawings, synesthetic perception, and the rhythm of architectural practice. His emphasis on tools, the distinction between chirographic and typographic cultures, provide valuable insights for understanding architectural imagination and the use and design of computational design tools.

The different dimensions of the concept of architectural space in the context of computational design practice is reviewed in this chapter. The formation of the concept of architectural space in Modern Architecture was examined and its origins were explained. Then, the chapter reviewed the notion of space as computationally constituted, exploring the role of programming languages and computational approaches in shaping architectural space. The concept of space as beyond the realm of the computable is presented with acknowledging the abstract, experiential, and elusive qualities that resist computational representation. Finally, the chapter considers the role of design tools in relation to architectural space, clarifying how designers engage with space through various design methodologies and approaches.

The insights gained in this chapter will provide a strong foundation for our exploration of the transformative impact of computational design tools on the conception, creation and thinking of architectural space.

CHAPTER 3

METHODS OF STUDY

This chapter introduces the research design and qualitative methods of the study. After the different method trials experienced, the thematic analysis method followed in the research to analyze qualitative data which are obtained from online interviews. It is aimed to identify patterns or themes within the qualitative data while providing a detailed and systematic exploration of participants' perspectives and experiences. To ensure the reliability of the data, a detailed description of the key participants is provided, highlighting their expertise and involvement in the study. Various methods of data collection and interview trials were conducted, followed by the transcription, coding, and analysis during the study.

3.1. Thematic Analysis Method

This study posits that the practice of computational design has led to transformations in the understanding of architectural space. The aim is to explore this process by gathering insights from practitioners who have undergone and experienced this computational transition. Given the limited availability and diverse locations of suitable participants, the interview process will be conducted in a written and online format. Employing thematic analysis through online written surveys offers a systematic and rigorous approach to comprehending complex and multifaceted aspects of current conceptualization of architectural space in the practice and in architectural theory. Interviews enables to capture participants' nuanced experiences, identification of emerging themes, and acquisition of valuable insights. Thematic analysis has its origins in Grounded Theory and Phenomenology which have similarities in theory and practice and is rooted in the criticism of Interpretivism expressed against Positivism (Guest et al., 2011, p.11).

> “Applied thematic analysis as we define it comprises a bit of everything—grounded theory, positivism, interpretivism, and phenomenology—synthesized into one methodological framework. The approach borrows what we feel are the more useful techniques from each theoretical and methodological camp and adapts them to an applied research context. In such a context, we assume that ensuring the credibility of findings to an external audience is paramount, and, based on our experience, achieving this goal is facilitated by systematicity and visibility of methods and procedures.” (Guest et al., 2011, p.15)

Thematic analysis with a bottom-up approach involves deriving knowledge directly from the data without being constrained by pre-existing theories or predetermined categories. It allows themes to emerge organically from the data itself, ensuring that the analysis remains grounded in participants' perspectives and experiences (Terry et al., 2017, p.17). Researchers engage with the data, such as interview transcripts or open-ended survey responses, by reading and re-reading them multiple times. This process helps researchers become familiar with the content, gaining a deep understanding of the nuances and intricacies of the data (Braun et al., 2021).

Initially the analysis starts with open coding, which involves systematically identifying and labeling meaningful units of text (Corbin & Strauss, 1990, p.12). At this stage, the aim is to capture the essence of the data either in descriptive or interpretive format. The focus is on capturing the explicit and implicit content of the data without a predetermined approach. When the coding progress matures, the analysis continues with comparing and grouping similar codes based on their shared characteristics or meanings. This process involves examining the relationships between codes and identifying potential patterns or connections to determine superordinate categories from the clustered codes.

The analysis follows a hierarchical organization starting with codes at the very bottom, continuing with clusters, categories, and superordinate categories to organize data (Strauss & Corbin, 1998, p.163). Categories refer to groups of codes that share commonalities or themes within the data. These categories represent meaningful units of analysis and help to organize the data into coherent groups. Superordinate categories are overarching or higher-level categories, and broader conceptual categories that encompass multiple subcategories or clusters. They are derived directly from the data and are not predetermined. The aim is to capture the richness and diversity of participants' perspectives and experiences.

Outputs are reviewed and refined by examining their coherence, relevance, and significance. The focus is on assessing whether the superordinate categories and categories accurately represent the dataset and on considering the fit with the research questions. Categories and superordinate categories are revised, combined, or split into clusters as needed. They are named descriptively to reflect their content and meaning and explained with clear and concise definitions that capture the essence of the data.

3.2. Interview Design Process

Qualitative data was collected through interviews with a specific group of individuals who are actively involved in computational design practice. Various interviewing methods were employed, including both informed and uninformed one-on-one conversations, online face-to-face discussions, and written exchanges, which will be described in detail below. Finally, it is decided to send the questions to the selected interviewees by e-mail to be answered by filling an online form for each question on internet or writing their comments and return the form back. It was expected that the personal experiences shared by the interviewees would yield noteworthy themes, significant statements, sentences, or quotes. It is thought that by the evaluation of these responses the imagination of architectural space in computational design practice might be at least partially enlightened.

At first, three pilot interviews were conducted following the structure of the first prepared Questionnaire 1, which is provided in Appendix C: Preliminary Interviews. These interviews consisted of four sections: personal background information of the interviewees, questions about architectural space, questions related to computational architectural design, and general topics for discussion. The interviews were semi-structured, conducted in person, and the questions were not shared with the participants beforehand. The three pilot interviews were conducted with three participants. In addition to these interviews, two interviews were conducted with two architectural students with experience in computational design tools. After the completion of the interviews, the outcomes were evaluated, and certain questions were removed. Firstly, it was determined that detailed personal background information was not the primary focus of the research, as it took up a significant amount of time during the conversations. Therefore, this section was eliminated. Additionally, some questions were found to be ineffective in eliciting spatial concerns from the interviewees, and they were also removed. The aim was to maintain the interviewees' focus on spatial thinking and computational design practice.

Afterward, the subsequent Questionnaire 2 was prepared, consisting of three sections: personal design process and approach, personal understanding of space, and the concept of space in computational design, comprising a total of 18 questions. Various interviewing methods were employed during this phase. For one interview, the questionnaire and introduction paper were sent to the interviewee in advance, allowing them time to reflect before conducting a face-to-face conversation on Skype. Another interview was conducted solely through written correspondence, where the documents were exchanged via email. Another one was conducted in person, without sharing the questions beforehand. Face-to-face interviews were carried out with three participants, while remote interviews were conducted online with two participants. Among these methods, the most effective and concise approach proved to be sending the introduction paper and questionnaire prior to conducting one-on-one interviews. However, based on feedback from the interviewees and critics during the research progress, it was noted that the questions needed further refinement, as some were unclear, terms were ill-defined, and indirect speech was used. Therefore, the questionnaire was revised once again, this time focusing on organizing clearer, shorter, and more direct question components, accompanied by informative paragraphs to provide framing.

Revised Questionnaire 3 was developed based on feedbacks, consisting of an introduction and four sections: design process, design approach, understanding of space, and the concept of space in computational design. This questionnaire comprised a total of 16 questions. Recognizing the value of obtaining feedback from relevant experts from the field, their input was sought to prior to sending the questionnaire to the interviewees, with the aim of identifying any overlooked aspects of the interview process. The revised questionnaire was shared with Prof. Dr. Zeynep Mennan, Prof. Leon van Schaik, Prof.

John Peponis, and Prof. Şebnem Yalınay, and their responses highlighted certain areas for improvement. It was noted that the questions remained too general, open-ended, ill-framed, indirect, and lacking a clear focus. The advisors suggested that a smaller number of more directed and targeted questions would be preferable. However, the introduction part received positive feedback from all the advisors, as it effectively addressed the intended purpose.

Lastly, Questionnaire 4, which is presented below, was specifically designed to explore the aforementioned feedback and concepts. It commenced with an introductory section and was organized into three sections: tools and design process, tools and spatial thinking, and the understanding of architectural space in computational design approaches. A total of seven consecutive questions were included in these sections. The questionnaire received responses from a total of eighteen participants, who actively contributed to the study by providing their insights and perspectives. Questionnaire 4 is presented in Figure 7.

Introduction: The use of computer technologies and computational thinking in architecture has been more pervasive since the 1960s and its transformative implications have yet to be fully discussed in architectural theory. Design as a way of thinking is exposed to a considerable transformation. Architectural jargon has become associated with parameters, algorithms, relational input/output structures, data-oriented models, digital tools and production systems. However, some constitutive elements of architecture, especially the concept of space, have not been emphasized as they were earlier.

Space is one of the main tenets of architecture since Modernism. Architectural education has been formulated primarily around the concept of space, which as an uncontested concept in architectural theory, is also one of the most discussed ones. Space is a polysemic term which includes many meanings and interpretations in one word.

Although the computational process might sometimes cause unexpected surprises, it is actually based on precision and prediction. Computational procedures need to be explicit, transparent and traceable in programming language. Space is embedded in computational design practice through algorithmic procedures and computable parameters. However, instead of being a constitutive tenet of the design process, space might have become an after-the-fact product in computational design approaches of contemporary architecture. Architectural object is usually composed of physical concrete elements such as surface, shell, mass, module, piece, system details etc... in computational design practice. Yet, the descriptive and prescriptive approach in computational design practice mainly deals with the "How" questions about the production and realization of architectural form. In some instances, consideration of architecture as tangible objects led to a lack of concern for space, which is more intangible than tangible. It is worth to question how computational design today reckons with space as a multi-meaningful, undefinable, perpetual, and ineffable concept.

Computational design proved itself by feasibility and constructability in architectural practice but the concept of space as an ephemeral concept has not yet been scrutinized in the computational paradigm. This inquiry will explore the concept of space as it is re-constituted in the paradigm of computational design practice. The assumption is that space has both computable and uncomputable aspects and that it is worth exploring how computational practice treats both aspects.

Interview Questions

1. Tools and Design Process: Architects have been using sketching and physical models during the design process since at least Renaissance. Both of these representational tools, which are themselves spatial, allow designers to think about space through spatial devices. With the prevalent use of computational tools, we see the use of new representational tools such as scripting, algorithms, and parametric formulations which are essentially non-spatial representations.

1.1. What are the reasons why you use computational tools in your design process? Is it more efficient, does it facilitate the generative process, does it help in visualizing and spatial thinking, does it enhance creativity etc…

1.2. In your personal design process, how do you benefit from or use these conventional spatial representations and the newly emerging computational tools? Do you sketch something first, or do you do modelling / coding / scripting first or do you use all of these simultaneously? If you prefer certain tools explain how they are used during your design process and if you prefer more than one tool how and why you go back and forth between different tools?

2. Tools and Spatial Thinking: Every tool, every representational system we create simultaneously opens and closes different ways of thinking, which in turn define who we are. As often attributed to McLuhan, "We shape our tools and then our tools shape us." Following McLuhan's idea, our imaginations of space are at least partially shaped by the tools we use.

2.1. Could you describe the importance of computational tools for you while imagining architectural space with regards to your design philosophy, formal repertoire, building technology etc.? Do computational tools help you conceive an image that you cannot think, draw, produce without them or do computational tools help you enrich your imagination process?

2.2. When do your spatial ideas and images emerge in your design process? Could you describe the process, relations, and negotiations between the computational tools and your understanding of space? How do the computational design tools foster spatial thinking? Could you illustrate it through examples of your own work?

2.3. What is the importance of the quality of space in your design approach and how do computational design tools ensure the quality of space?

2.4. Could you describe spatial concerns that are personally significant to you, and could you describe how your personal spatial concerns and anticipations are fed into the computational process? Please illustrate through examples of your own work.

3. Understanding of Architectural Space in the Computational Design Approaches

3.1. Which spatial qualities, concepts, values, and norms are best represented and examined through computational tools and which ones are not? What are some emerging aspects of architectural space and some alterations in our understanding of space which might have been triggered by the computational design approach? Please illustrate with examples if possible?

Figure 7. Questionnaire 4.

The thesis primarily focuses on the data collected through the final questionnaire, while acknowledging the valuable insights obtained from all previous interviews. However, it is important to note that the data obtained through different methods were not combined during the analysis phase.

3.3. Participants

Geertz (1973) in The Interpretation of Cultures, argues that to achieve a genuine understanding of a subject, one must closely examine the actions and methods employed by practitioners in that field. The active involvement of the participants in this study ensures that the research remains rooted in the real-life experiences and perspectives of those under investigation. Geertz (1973) states:

> "If you want to understand what a science is, you should look in the first instance not at its theories or its findings, and certainly not at what its apologists say about it; you should look at what the practitioners of it do." (p.5)

While the respondents of the questionnaire are experts in their fields, their backgrounds and practices offer a wide variety. The participant group consists of respondents from different age groups, genders, and countries. The main criterion of participant selection was the level of expertise in computational design determined in this research based on completion of several design and production processes with computational design tools. Details for a sample of participants are presented below without importance and priority order.

Michael Hansmeyer, one of the respondents, is an architect and a programmer who works in different practices. He has many studies in different academies and has many publications on CAAD. Kas Oosterhuis, another respondent, is an academician at TU Delft Faculty of Architecture and director of Hyperbody research group, in addition he has many projects in Studio ONL. Sabri Gökmen, a third participant, defines himself as a computational designer and researcher with expertise in computational geometry, parametric modeling, generative algorithms, digital fabrication, and software development. Ulla Hell is a guest lecturer of design courses at the University of Innsbruck and one of the partners of Plasma Studio known with works like houses of Esker, Paramount-Alma, Cube, Dolomitenblick to Xixian Restaurant, Creativity Pavilion and Datong Twin Towers. Giulio Piacentino is an architect teaching at TU Delft as a guest lecturer, organized workshops who also develops computational tools by scripting. Burcin Nalinci who is the founder of the Studio bits2atoms and is interested to extend computational tools into physical space through robotics, machine design, and material engineering. Toni Österlund is an architect who has many publications on digital design and one of the editors of eCAADe 2016 Complexity & Simplicity. He also realized many projects in workshops and in practice. All participants expose the skills of dealing with computational design tools and complex geometries that provide indispensable and unique computational design experience.

Table 1. Interviewees in alphabetical order.

Axel Kilian	Kas Oosterhuis	Sebastian Gatz
Burcin Nalinci	Marios Tsiliakos	Thom Faulders
Davide Del Giudice	Martin Tamke	Toni Österlund
Davide Mariani	Michael Hansmeyer	Tuğrul Yazar
Efilena Baseta	Rocky Hanish	Ulla Hell
Giulio Piacentino	Sabri Gökmen	Yannis Zavoleas

Eighteen participants, whose names are presented in the table above in alphabetical order, contributed to the study and their contributions examined by the thematic analysis method. The call to participate to the research was sent out to about 110 architects in two rounds of calls via personal correspondents or via e-mail. The study was described and explained in the invitation to participate in Appendix F, and participants gave their consent to participate. Out of 30 potential respondents, 18 agreed to provide answers to the questions. The number of respondents were decided to be sufficient when the answers were exploratory enough of the field and when additional efforts to increase the number became more and more futile. Although the participants of the face-to-face and pilot interviews conducted within the scope of the thesis are not specified in the table and are not included in the thesis, they are recorded for post-thesis evaluations and studies. Selected quotations of the participants are cited in the dissertation with their own words. However, quotations of interviewees are anonymized to avoid discussions about anyone. In addition, contributions of the participants were corrected as much as possible with their approval to clarify the content of their statements.

3.4. Data Collection

Qualitative data were obtained from online written interviews consisting of open-ended questions. Including open-ended questions in surveys allows respondents the freedom to express their thoughts and approaches about a specific topic, enabling the collection of exploratory data that might uncover unexpected opportunities, issues, or noteworthy quotes. These quotes or examples carry impacting and powerful statements. According to Braun et al. (2021) online surveys provide a valuable method for conducting qualitative research. It is demonstrated that online surveys are as an effective and flexible tool that keeps the richness of qualitative data, benefiting both researchers and participants. In this case the online written format survey is conducted via email exchange or using Survey Planet web page platform. The online application also automatically recorded data such as location, date, time spent on the answering process. Based on 10 participants using an online application, it is seen that the average time spent for participation is 35 minutes.

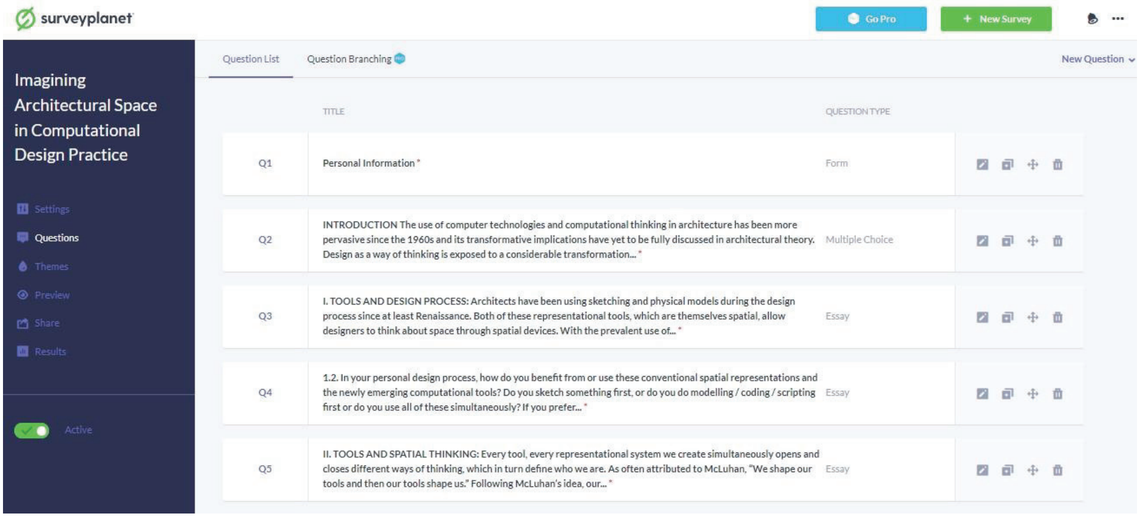

Figure 8. Survey Planet interface screenshot.

Examining the participant independently of others requires an ethnographic approach, which was not followed in this study. However, the responses of the participant below are presented as a cohesive whole, intended to be read with a holistic understanding rather than being treated as individual quotations, as is done in the thesis. Interview of the participant P15 is attached in the Appendix B.

3.5. Data Analysis

In the analysis phase, the research adopted thematic analysis method which "is a rigorous, yet inductive, set of procedures designed to identify and examine themes from textual data in a way that is transparent and credible" (Guest et al., 2011, p.15). According to Guest et al. (2011, p.15) thematic analysis, grounded theory, and phenomenology are all inductive analysis approaches that share similarities. However, thematic analysis stands out due to its broad scope, not limited to theory-building like grounded theory or subjective human experience like phenomenology. Thematic analysis offers flexibility in theoretical frameworks and analytic tools, accommodating both positivist and interpretive approaches. Its systematic and transparent process is in harmony with interpretivism, emphasizing the absence of inherent conflict between the two.

3.5.1. Coding Qualitative Data

According to Charmaz (2006, p.42) the coding is the beginning of bringing analytical perspective to the qualitative data. It involves the systematic organization and categorization of data uncover and interpret meaningful patterns within unprocessed qualitative data. It is emphasized that coding is not only description but involvement through the identification of concepts, patterns, and relationships within the data. (Strauss & Corbin, 1998, p.55)

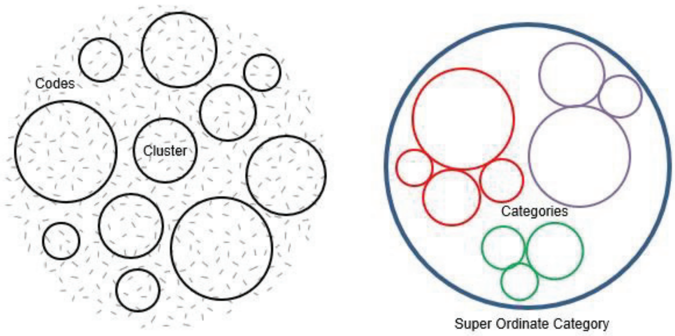

Figure 9. Bottom-up coding hierarchy.

The study employed three primary coding procedures: open coding, selective coding, and axial coding. Open coding served as the initial step, involving the identification of concepts and exploration of their properties and dimensions within the data. Axial coding, which focused on establishing relationships between clusters, categories, and their superordinate categories, was used in the second stage. This coding process, referred to as "axial," revolved around the axis of a category, facilitating the linkage between different categories. Lastly, selective coding was implemented to integrate and refine the data, providing a comprehensive perspective.

> "In open coding, the analyst is concerned with generating categories and their properties and then seeks to determine how categories vary dimensionally. In axial coding, categories are systematically developed and linked with subcategories. However, it is not until the major categories are finally integrated to form a larger theoretical scheme that the research findings take the form of theory. Selective coding is the process of integrating and refining categories." (Strauss & Corbin, 1998, p.143)

In this study, since the data were obtained from online written interviews, the text was directly adapted to open coding without any alterations. In the open coding phase, codes and clusters started to form both within the individuals themselves and with the comparative evaluations between the responses from all participants. In open coding, all the prominent elements are marked without any discrimination or elimination, in fact, this stage is the first step of the possibility of turning the data into information.

MAXQDA is utilized as qualitative data analysis software tool, which offers effective management of large amount of interview scripts. The software also effectively facilitated managing the taxonomy of emerging clusters, categories, and superordinate categories. In the figure below (**Hata! Başvuru kaynağı bulunamadı.**), a sample of analysis i nterface presented. On the left of the interface are the codes, clusters, and categories which are color-coded according to the questions, and on the right are the responses of the participants associated with the codes.

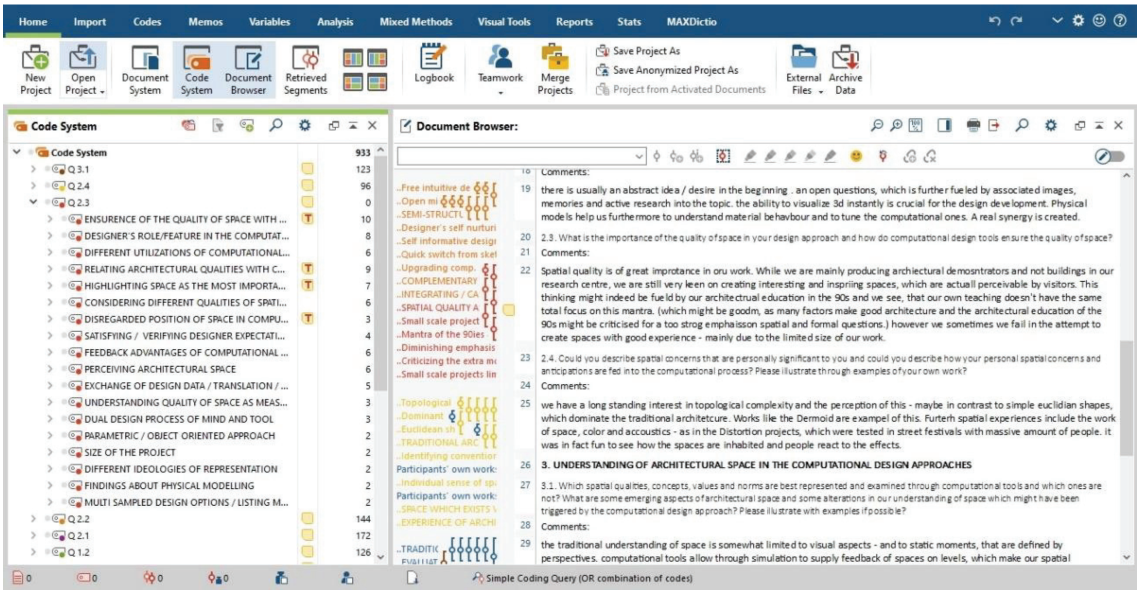

Figure 10. A screenshot from the qualitative data analysis software.

All generated codes were distilled through axial and selective coding stages. **Hata! Başvuru kaynağı bulunamadı.** shows the emergent categories and super ordinate c ategories established at the end of the research.

Table 2. Emergent categories and superordinate categories.

Superordinate Categories	Categories
Preference reasons of computational tools	Imagination related reasons
	Pragmatic reasons
Using different design tools together	Modes and combinations of design tool use
	Chronology and order of use of design tools
	Capacities of design tools
	Design of the design tool
Representation with design tools	Mind mapping medium
	Externalization of emergent design ideas
	Transformation and transfer of design data
Refinement of design and tool	Evaluation capabilities
	Design tool feedbacks
Increasing design capacity	Offering catalog of options
	Expanding and exploring design search space
States of the design process	Compatibility and consistency of tools
	The transformation of the design process
Changes in the profession of architecture	Changes in the profession of architecture
	Changing modes of relations
	Outsourced concepts in architecture

3.5.2. Categories and Super Ordinate Categories

The coding guide used in this research is presented in APPENDIX A in detail. Additionally, the guide includes segments from the qualitative data to illustrate each category. Here, superordinate categories and their subsets categories are listed while providing definitions for the set of 18 categories that were developed through the described analysis procedures.

1. Reasons for preference of computational tools
   a. **Imagination related reasons:** The reasons for using computational tools which are related to the creativity, freedom, and novelty that computational tools offer in terms of new, unique, and unconceived designs.

b. **Pragmatic reasons:** The reasons for using computational tools which are related to the production, fabrication, and construction processes of architecture in terms of material, time, and labor advantages.

2. Using different design tools together

a. **Modes and combinations of design tool use:** Alternately using digital or analog design tools, computationally, or not, with different modes and purposes. The combinations of tools are either complementary, combinatory, integrated, or simultaneous. For instance, Analog + Analog, Digital + Analog, Digital + Digital tools are utilized. Different ways of using the tools are possible for instance analog tool in computational use, analog tool in noncomputational use, digital tool in computational use, or digital tool in noncomputational use.

b. **Chronology and order of use of design tools:** Determining which design tools are used in which phases of the design process for which purposes or services.

c. **Capacities of design tools:** Instances of the limits and constraints of different design tools whether digital or analog, computational, or not which could be related to why designers customize and combine tools.

d. **Design of the design tool:** Accepting the design tool as one of the active actors of the design process which is shaped by designer (customization / hybridization) and shapes designer (fixed / predetermined).

3. Representation with design tools

a. **Mind mapping medium:** Mapping of intangible thoughts, abstract data, or ill-structured initial design ideas to get closer to visual representations of architectural space via digital or manual, computational, or non-computational tools.

b. **Externalization of emergent design ideas:** Capturing and following-up of an emergent design idea while keeping the designer intuitions through the design process.

c. **Transformation and transfer of design data:** Capturing and following-up of an emergent design idea while keeping the designer intuitions through the design process.

4. Refinement of design and tool
   a. **Evaluation capabilities:** Testing, simulation, and optimization facilities of computational tools through the design process.
   b. **Design tool feedbacks:** Continuous evaluation, clarification, and improvement of the design decisions through the design process until it is refined. Nurturing design outputs with the iterative and informative feedback loops (between different design tools computational or conventional).
5. Increasing design capacity
   a. **Offering catalog of options:** Availability of listing all various possible design options with computational design tools.
   b. **Expanding and exploring design search space:** Computational design tools expand the design search space and change the direction of exploration.
6. States of the design process
   a. **Compatibility and consistency of tools with idea, design, and production:** Compatibility and consistency of tools with idea generation, design development and production which includes phases such as capturing emergent intuitive design idea, conceptual development, representation, and complete production.
   b. **The transformation of the design process:** Changes and transformations in the design process with the use of computational design tools.
7. Changes in the profession of architecture
   a. **Changes in the profession of architecture:** Changes in the profession of architecture in terms of issues such as organization, working practice, way of working, and in the role of the architect in terms of expectations, services, naming themselves, etc.

b. **Changing modes of relations between design, tool, and designer:** Identifications of the changing and conserved relationships between design, tool and designer while using computational design tools.
c. **Outsourced concepts in architecture:** The outsourced (spolia) concepts of architecture which especially originated in and derived from mathematics.

Table 3. Conceptual analytical graphic.

CLUSTERS
Imagination related reasons
Pragmatic reasons
Architectural space and spatiality
Integrated use of different design tools
Sketching in the Computational Design Process
Design of the design tool
Refinement in the design
Spatial Concerns in Computational Design Practice
Augmented Capacities
Bottom-up and Top-down Strategies
Feedbacks
Consequences of Incompetent Use of Computational Tools
Phases of Mental, Representational, and Bodily Experience of Space
Directions of Thinking and Designing
Vocabulary and Language of Architectural Space Descriptions
Data-oriented Definition of Architectural Space
Design data exchange
Interpreting Design Tool
Ensuring the quality of space
Quality of space as an integrative phenomenon
Relating Architectural Qualities with Computational Quantities
Designers Considering Architectural Background
Individual experiences in the formation of the architectural space
Fragmented and piecemeal consideration of architecture and space
Externalization from idea to representation
Bodily senses as the interaction tool of perceiving and making
Changes the profession of architecture
The transformation and new state of the design process
Reflections on computability and noncomputability
Emerging issues and changes in the understanding of space
Ways to improve computational design tools

CATEGORIES
Imagination related reasons
Pragmatic reasons
Modes and combinations of design tool use
Chronology and order of use of design tools
Capacities of design tools
Design of the design tool
Mind mapping medium
Externalization of emergent design ideas
Transformation and transfer of design data
Evaluation capabilities
Design tool feedbacks
Offering catalog of options
Expanding and exploring design search space
Compatibility and consistency of tools
The transformation of the design process
Changes in the profession of architecture
Changing modes of relations
Outsourced concepts in architecture

SUPERORDINATE CATEGORIES
Preference reasons of computational tools
Using different design tools together
Representation with design tools
Refinement of design and tool
Increasing design capacity
States of the design process
Changes in the profession of architecture

3.6. Validity

To ensure higher levels of reliability and validity in qualitative research, various strategies could be conducted. This study would follow inter-rater reliability method. Inter-rater reliability evaluation involves independent coders who follow coding and interpretation methods to reach a level of concordance, thereby enhancing the rigor of the research (Creswell, 2008, p.176). The level of concordance indicates the similarity of codings among multiple coders and accepted proof of reliability. Appendix D would be used as a guide or base in the interview with the inter-rater coders to explain and discuss coding protocol. The objective of this process is to determine the level of agreement between independent reviewers, which serves as an indication of high inter-rater reliability. When the data reliance is between 64% and 81%, the level of agreement is considered strong, while a data reliance of 82% to 100% indicates an almost perfect level of agreement (McHugh, 2012, p.279).

In the interrater reliability study, the responses of a single participant were analyzed. The answers were transmitted to the coder as they were given. The first round of the interrater coding was conducted by delivering the aforementioned coding guide to the coder understanding and interpreting it on his/her own and completing the coding process. Although the researcher coded 27 codes on this participant, interrater coder identified 13 codes in the first round. The level of concurrence obtained in the first round was 54%. The first study coded by the interrater coder is presented as APPENDIX E. Then, a meeting was held with the interrater coder and explanations were made regarding the points where the coding guide could not enlighten. The coder also stated that when the coding guide is examined, it is supposed that answers should be coded as blocks without dividing sentences. This misunderstood was resolved in the interview. In the second round of the coding, the concurrence rate increased to 67%, which thought to be developed. In the third turn, the approaches, and ways of thinking about the researcher's coding and categorization system were explained to the coder. The coder was convinced of the researcher's approach in some categories. The final concurrence level of the research is calculated as 77%. Once a considerable level of concurrence was achieved concerning the authorities accepted, interrater reliability study has been finalized.

CHAPTER 4

RESULTS

In this chapter the contributions of the participants are presented and examined in detail. The sequence of the examination continues in parallel with the order of the interview structure which consists of seven questions in three parts. The discussion provides detailed narrative on design tools, design thinking and architectural space by filtering the prominent approaches, concepts, and relationships within the answers obtained. The results are presented in a format in which quotations from the participants are discussed in relation to prominent concepts from the related literature. Wording and spelling errors in the quotations from the participants were reviewed and adjusted to refine the content of their statements.

4.1. Introduction to Interview Section 1

In the first section of the interview, there are two questions that ask participants to elaborate their design processes and the use of tools. A general approach to shape the questions, which is articulated to the participants at the outset, involves the developing representation tools within architectural design processes and the influence of these tools, if there is any, in reframing the meanings concerning the concept of architectural space. The questions inquire the logic behind participants' orientation towards certain computational design tools and how the participants specifically employ them in the design processes.

Tool, as a concept, is one of the key phenomena in explaining the evolution of human cognitive abilities (Tomasello, 1999). The tool use is one of the main subjects whose continuity being examined in anthropological studies. There are many observations conducted on both human beings and apes to understand cultural and cognitive evolution of tool development and use. Humans understand and make sense of the world with and through tools that have traces of human cognitive capacities and have reflections on human intelligence and culture. The use and refinement of tools is a parallel and concurrent process with human evolution (Tuttle, 2023). Homo species are differentiated from others by making and using tools in their mundane practices. This differentiation created anatomical reflections on individuals and social and cultural reflections on groups in the evolution process. Tools that have developed from prehistoric times to the present have accompanied civilizations. The emerging needs and development of humanity are driven by the mutual interaction of tools and people. The human evolution is an ongoing process from the first rock carved as a tool to sophisticated artificial intelligence technologies. One of the most sophisticated levels of instinctive collaboration with tools are experienced in our time. Human beings add a new link to the evolution chain with the computers they have invented.

The origins of architecture are based on the human instincts to create a shelter. There is an evolution from the instinct of shelter to the discipline of architecture and the tools whether these are for construction or manufacturing, production, representation, or design have always existed and accompanied throughout the ages. Architects have been using drawing and physical models during the design process since the Middle Ages. Both of these representational tools, which are themselves spatial, allow designers to think about space through spatial devices. With the prevalent use of computational tools, we see the use of new representational tools such as scripting, algorithms, and parametric formulations which are essentially non-spatial representations. Therefore, in the first set of questions, the participants are asked to introduce their design tools as they engage in exercises of space-making.

4.1.1. Examination of Question 1.1

First question inquired into the reasons why designers prefer computational design tools in architectural design. Typically, the decision about which tool to use is made at the outset of design. Even just the tool selection is a decision that is taken at the beginning but has effects on the whole design process. This decision may also lead to the replacement or support of the design tool with another one as design work progress. Participants were encouraged to refer their experiences and observations within different stages of design, concerning mostly the intellectual and representational issues in space making. The question 1.1 is as follows:

1.1. What are the reasons why you use computational tools in your design process? Is it more efficient, does it facilitate the generative process, does it help in visualizing and spatial thinking, does it enhance creativity etc…?

In participant responses, there is a multiplicity of reasons for using computational tools which was approached and evaluated through various categories. Although the participants did not form a homogeneous view, they frequently expressed their opinions in reference to the concept of creativity. The concept of creativity has also been brought to the agenda by addressing different issues like freedom, newness, visualization, and making the impossible possible with computational design tools.

Beyond creativity, issues related to manufacturing and construction were frequently mentioned by the participants. Additionally, production related processes were also evaluated by including different concepts like complexity, efficiency, precision, and standards of the practice. All the participants who mentioned the different dimensions of production had emphasized the value of preferring computational design tools to achieve better results in the quality of building construction.

Another prominent reason suggested by the participants is the capacity to make alternative design options available for architects. This is related to expanding the design search space and generating a catalogue of options which is a highlighted advantage of employing computational tools in design.

One other finding to mention is the developing terminology in computational practices which is imported into architectural vocabulary. In addition, there are also some comments on the role of the designers and their relationships with design tools. Although few, there are also opinions about architectural space and spatial concepts.

Following the qualitative analysis, the reasons for employing computational tools are presented in two categories, namely imagination related and pragmatic reasons.

Imagination Related Reasons

Eight out of eighteen participants explained their reasons for preferring computational design tools primarily by addressing issues related to the “architectural” imagination. The imagination-based reasons include the creativity and the freedom and control clusters. Here, imagination is considered as an intellectual activity especially at the early design phase triggered by representations supported by computational design tools. Some of the imagination-based reasons are in the scope of the mental activity of design process while some of them are related to changing capacities of design tools on representation or production. It is revealed that the participants considered the reasons related to imagination with different dimensions when choosing computational design tools.

Creativity is the most highlighted concept of imagination-based reasons and is considered in relation to novelty, design search space, serendipity, and abstraction. Increasing and transforming representation and production capacities along with computational tools are also interpreted with reference to creativity. Another notable issue is the changing positions and expectations of designers in the demand of freedom and control while using design tools.

Creativity

Creativity is a concept that has multiple definitions in the literature. It is one of the universal parameters in which designs are evaluated. When it is asked about the reasons of preferring computational design tools the creativity is mentioned by nine participants from different point of views that make it one of the most intensely reacted concepts. The views on the effect of computational design tools on creativity were examined and the approaches were divided into two. The first group stated that computational design tools positively support creativity. On the other hand, the second group stated that they preferred computational design tools for reasons other than creativity. Participants explained that computational design tools offer different advantages rather than just creativity related ones and that designer's competence and authority is becoming ever more important with the increasing use of computational tools.

Five participants stated that computational tools have positive effects on creativity. For example, it is claimed that computational tools make possible the forms which were unseen and impossible before. For instance, Participant P9 states:

> P9: *The goal is newness. To create architectural form that has hitherto not been seen, that could not have been produced, that could not have been conceived of. Computational tools to expand the mind and imagination.*

P9 claims that computational design tools enrich imagination and make it possible to conceive an image which cannot be thought, drawn, or produced without these tools. P9's statements offer an emphasis on new or increasing capacities of design tools to enhance creativity in creating architectural forms.

Participants' opinions concerning the value of computational tools supporting creativity highlight the capacity of presenting many options within defined set of design rules. The computational tools offer the use of algorithmic models to generate a list of probable solutions that satisfy preset rules and conditions. Algorithmic models in which design decisions are embedded list all probabilities that comply with the rules and satisfy the conditions. When designers set up their specific design rules, the computational process might output unanticipated possibilities. In this sense, participants P17 and P7 articulated their approaches as follows:

> P17: *Having an algorithmic design process, computational design tools enable the visualization of complex geometric relationships based on simple mathematical relationships. The result is never a master-piece but a catalog of options that often surprise the initial logic.*

> P7: *… it opens up multiple avenues for how design could be carried because the designer gains the advantage to look at multiple variants of a system and choose an end result that satisfies a design problem.*

The formulation of enhanced creativity by expanding the design search space is also related to the 'unanticipated emergence' (Knight & Stiny, 2001). While designers carry out the computational design process as a self-possessed, restrained, and controllable process, they welcome the unpredictable outputs as their individual design products and adapt them. These kinds of ordered accidents are welcomed and accepted as creativity triggering conditions in the design process. The concept of welcomed unpredictability is emerged from the participant's contributions. Computational design outputs might surprise designers and deviate them from their expectations because of the unpredictability of complex, generative or algorithmic design process. This concept matches with the ‘happy accidents’ definition which is explained as ‘unintended results become favored design decisions’ (Chaszar & Joyce, 2016, P.168). Accident and unanticipatedness are frequently cited concurrent phenomenon with different statements. For instance, Rocker (2006) quotes Lynn as follows:

> Lynn: I started to learn the software by experimenting, but after a happy accident it only makes sense that you practise, master and integrate the unanticipated result into a technique. I love the moment when I discover some new potential in software, but once I find it I try to turn this into a technique. (p.90)

Greg Lynn also admits that he welcomes the unanticipated output and spend an effort on it. Additionally, he claims that “the era of the happy accidents is over in architectural design using computer-based design mediums” and in-depth understanding, customization and manipulation of design medium needed.

The roots of the unanticipated and serendipitous discovery in the design process is also discussed by John Frazer (1995) through the concept of ‘intentionality’ via referencing nature and evolution. He opens the topic as follows:

> “To us the connotations of the term 'design' are very different from the norm: when we 'design', we are clear in our intentions, but perhaps 'blind' to the eventual outcome of the process that we are creating. This 'blindness' can cause concern to those with traditional design values who relish total control. It can alarm those who feel that what we are proposing might get out of control like a computer virus. However we remain convinced that the harnessing of some of the qualities of the natural design process could bring about a real improvement in the built environment.” (p.12)

In Frazer’s conceptualization, the intention is an executive motivation in the design process and ‘blindness’ is suggested as a positive concept for designers. Designers shape their designs through tools. For example, artists also use tools, but they are closer to their sculpture or painting and interact with their artifacts more directly than architects. Actually, artists produce their artifacts via tools, but architects initially produce the representations of their artifacts. Artists also tune and consider their tools, however, the importance and the effect of the tools on the artifact has different dimensions to consider in the architectural design activity. One of them is the distance between work and designer which is named here as indirectness (Aish, 2005, p.11; Yazar, 2009, p.67). Similarly, the profession of architecture is performed through representations undergoing through transformations. All representation systems contain indirectness anyway and computational method offers a new set of representations but the distance or the depth between idea, designer, tool and the design is wider. Designer controls the design tools and outputs indirectly as usual in architectural design but in computational design tools the designers are at one step back and more isolated from the representation when it is compared with the customary design tools. In the distance between the design and the designer, some design processes are delegated to the computational design tools. Delegation of design tasks in computational process is matched with 'blindness' and it is affirmed by Frazer (1995).

In addition, when the bodily expression of the process is considered, for example, the pencil and paper duo of drawing has recently left its place to computers. The current transformation can be expressed as the delegation of the drawing process through scripted codes which are structured and formulated around specific design ideas. The knowledge of having a curved masonry system brick wall in height 300 centimeters of a line drawn on paper with a pencil in hand or the conceptual meanings of any abstract sketch that cannot be understood without explanation are re-evaluated with computational design tools.

Carpo (2017), in "The Second Digital Turn", explains a paradigm shift by highlighting the threshold between sorting and searching concepts which is basically due to the increasing capacities of computational power. Sorting means listing all existing or anticipated options as arborescent diagrams within the capacity of human prescience but searching is another level beyond organic intellect. While sorting is about finding a suitable place for a thing in an already existing or suggested classification system, searching is looking for a thing without a priori mental or physical set up as Carpo states:

> "As I have argued, this posthuman logic is already ubiquitous in our daily lives and embedded in many technologies we use. Nothing represents the

spirit and the letter of this new computational environment more than the "search, don't sort" Google tagline: humans must do a lot of sorting (call it classifications, abstraction, formalization, inferential method, inductive and experimental science, causal laws and laws of nature, mathematical formulas, etc.) in order to find and retrieve things—both literally and figuratively. Computers are just the opposite: they can search without sorting. Humans need a lot of sorting because they can manage only few data at a time; computers need less sorting—or, indeed, no sorting—because they can manage way more data at all times. To sort, humans must have a view of the world—regardless of which one came first, the worldview or the will to sort; computers need neither." (p.96)

The process of search may disclose the unanticipated relations for designers. Designers could demarcate their anticipated design search space by design parameters, yet computing algorithms might generate and reveal unanticipated design options. Since it is not possible to categorize an unexpected or unknown possibility/option, searching as a way of thinking in contemporary era overlaps as a compatible and complementary tool.

Computational design tools added a new dimension to abstraction in the creative process. The concept of creativity is related to abstraction and any representation contains a certain level of abstraction. Sketching is one of the familiar tools in which the early design ideas are abstracted and represented in the design process (Doğan & Nersessian, 2010). Representation methods and design tools embody certain notions of abstractions (Root-Bernstein, 1991). In customary design processes, the focus is basically on the design object itself as an expected end product. However, while using computational tools creative effort of the designers not directly deal with the design object but the design parameters which are expected to lead to a design solution. Data and information are inputs of the management and manipulation as a creative process. Regarding this, one of the participants, P15 states that:

P15: *Computational thinking allows the designers to work with data and information. The management and manipulation of information, in my opinion, is the proper contemporary creative process.*

New representation systems have transformed both the physical representation tools/agents and the way designers think. While the designers who focus directly on the design idea or object in the sketching process with pencil and paper, the final product is indirectly composed with digital tools and algorithms in the computational design process. The level of indirectness in the representation process defines the distance between the original idea in mind and the representation which is a product of the tool. The level of indirectness could be considered in terms of number of tools between mental image and represented image or number/amount of design knowledge transfers between different tools.

In contrast, three participants stated that they did not prefer or prioritize computational tools as creativity enhancer. The main reasons for the participants who think that computational design tools do not strengthen creativity are generally related to superior representation and production capacities. From their point of view, computational tools are not evaluated as creativity enhancer, or they are not more creative and/or more superior compared to other design tools in terms of creativity.

Participants who do not primarily attribute creativity to computational design tools find these tools advantageous in terms of their visualization, generation, simulation, evaluation, and construction capacities. Representing design ideas visually and evaluation of generated design alternatives could be categorized as creativity supporting but participants did not put it as such. The statements of the participants P2 and P14 could exemplify such an approach:

> P2: *We use computational tools because sometimes it is more efficient and helped in the generative process, also it helps visualizing, we use it a lot for testing variations on possible solutions, also in the constructive process, not necessarily to enhance creativity.*

> P14: *We do not believe that computational tools enhance creativity but rather extend the designers skills in visualizing and evaluating his/her ideas and subsequently materialize complex geometries, which is not possible without computation.*

Although the participants did not identify the reason for choosing computational design tools as creativity-related, the other reasons they stated are concepts that feed or trigger creativity. Visual representation of design ideas while dealing with complexity then generating and testing alternative design options while considering production is a complicated decision-making process related to creativity. In a follow-up interview, these participants may be asked what their definition of creativity is.

In addition, participants who use computational design tools thinking that these tools do not contribute to creativity, in a way positions the designer as the central control agent of the design process. It strengthens the role of the designer in the computational design process. Designer as a central figure of the design dominates the process.

Another prominent expressed reason is that computational design tools are just as effective and ineffective on creativity as other design tools. In terms of creativity, it is suggested that every design tool is equal, but it is also commonly accepted phenomenon that different design tools will lead to different results. As other design tools, the computation tools are also a means to an end. For example, participants P5, P8, and P1 express their views on this subject as follows:

> P5: *Like sketching and physical models, per your prompt above, computation tools are a means to an end. I became an architect before the use of desktop computing and drawing. As such, I am used to thinking about figure and form, space and atmosphere, material and affect, without feeling the need to "go to the computer" to explore what is attainable. That said, we (in my office and in my teaching) continually rely on computational tools and scripts to explore new possibilities, in ways that simply might not be possible to image without these tools. Design intent directs the tool, but the tool also has the power to challenge initial ideas, or to open up new directions.*

> P8: *Finally, I would say that no tool in particular will enhance spatial thinking or creativity. It is still the mind we rely on to do those things. Deciding to use a C-clamp, a press, a 2H pencil, or a parametric program will not inherently change your thinking - although changing tools will create a feedback loop that might tell you something interesting.*

P1: *They are mediums used along with other ones to express, process and communicate ideas and as such they may help think in different directions. Moreover, they offer great computational power, which allows to perform alternatives through parametric and generative processes. In effect, they can have a great influence in design concerning creativity. Still, I don't believe that they should count as criteria to assess architecture, as no other tool or method would either. Architectural criteria are of humanistic nature, whereas tools and media are more on the technical aspect, as any other medium. This should be clear from the start, as for example a text or a photograph are also media, but as such they don't direct content and the same applies with computational tools.*

The opinions above actually abstain from taking a position in stating nothing beyond that the design tool influences the designer. It is an agreed opinion that changing the design tool also changes the designer. This line of thinking could be evaluated and paralleled with the understanding of designer's central position in the computational process. Whatever the tool the designers use, it can change their thinking system or how they design, but the position of the designer keeps its presence in the design process.

Creativity is the most popular concept that participants commented on when the reasons of computational design tools is asked. It is evaluated that it supports creativity positively in terms of newness, revealing the unanticipated design search area and enhancing representation capacities. Besides that, it is described that tools are means to an end and the importance of the designers' competence and authority highlighted.

Freedom and Control

Freedom emerges as a contradictory concept when the explanations of participants are examined because the participants who stated that computational design tools provide more freedom in formal exploration often associated it with the concept of control. This is expressed in the statements of participant P16:

P16: *They are both. I mean it is really important to have the complete control of the geometries and to give freedom to the designer, so I can say* *computational design tools are now a basic standard tool when we work in complex geometries field.*

While designers welcome unexpected outputs, especially in the early design stages, expectations of freedom and flexibility are dominant. Freedom is often associated with complex form, and control might be interpreted as a flexible activity which helps to explore ill-defined design problems via experimental design moves. In the subsequent manufacturing and construction stages of the design process, the demands for control and precision are decisive. Computational design tools meet the expectations of designers at different stages of design by offering different services. Considering the criteria evaluated when choosing a design tool, the adaptation capability and the wide range of service options are some of the fundamental features.

All things considered, it is seen that the capacities that computational design tools offer in representation and productivity are the most attractive factors when they are compared to customary design tools. Awareness of the distinction between tool and the designer has been clarified, and the creativity is considered as the responsibility of the designer.

Pragmatic Reasons

Another category relates to pragmatic reasons, which is highlighted by eight out of eighteen participants. Some of the reasons for participants to use computational design tools are explained in reference to production, digitization, evaluation, visualization, complexity, and impossibility which are prominent clusters of pragmatic reasons. The common point of this group is that computational design tools are associated with a utilitarian approach to the solution of problems. All the participants, who highlight pragmatic-based reasons, affirm computational design tools in this respect.

Emerging concepts listed in the pragmatic-based reasons are basically in parallel with the literature. Making or production related issues deal with efficiency, precision, and feasibility. Efficiency is considered multidimensional in terms of material, time, and labor considerations. Production with a high standard of precision at the level of craftsmanship has brought new insights into the practice of architecture and triggered a new way of revivalism. One of the strengths of computational design was to convince the community concerning the feasibility of production, no matter how complex is the architecture is. The only condition for this was digitization. One of the new benefits is that anything digitized could be subjected to different evaluations even before production. Visualization, which has become a way of justification in new production methods, is the representation of complex and abstract design ideas by digitization in computational platforms. Complexity is considered as sometimes a cause and sometimes an effect for designers. When computational design tools are considered in terms of production-based reasons, as one participant stated, they are the standard of the practice of contemporary architecture.

Production and Making

The production phase has been the most convincing and attractive area of computational design tools. Designers and construction industry has proved that computation works. The advantages of computational design tools and production methods were met with the favor of designers. For example, efficiency has been one of the reasons for preference frequently mentioned by the participants. Participants discussed the concept of efficiency with its different dimensions. In addition to efficient use of labor and time, there is also productivity advantages in terms of material use. The other prominent reason of production is precision involving two issues in design. The first is related to the quality in different scales of physical production and the second refers to the exact production systems of algorithms. Feasibility, which is the final stage of the transformation from mental idea to physical object in architecture, can be discussed at different levels in computational design practice in terms of application. There are forms of realization at different scales and with different sensitivities like generating plan layout options, tessellation of any geometry for façade solution or an optimization of a canopy structure, etc. Participants P11 and P7 express their views on the subject as follows:

> P11: *The primary reason for implementing computational methods on everyday practice is time efficiency. A simple parametric model can provide a vast amount of design options. Even if the method is not exactly what one would label as generative, it provides a valuable framework for delivering fast and robust solutions. The same logic applies to post-rationalization*

techniques, where the process might carry on for months if a computational approach wasn't utilized.

P7: *In my design work, computation is used for many different applications such as patterning, form-finding, optimization, and fabrication. Most of these processes rely on mathematical rules and follow certain material or abstract techniques which means they could be automated using scripts. This way computation provides both efficiency in terms of generating feasible results and it opens up multiple avenues for how design could be carried because the designer gains the advantage to look at multiple variants of a system and choose an end result that satisfies a design problem.*

After all, architecture is, in a sense, the art of making. However, computational design tools do not provide continuity without any problems in the production process. This is true not only in terms of the file-to-factory concept but also in the design-to-file phase. Manufacturing of architectural elements in factories are under control and precise but applications the construction site are not as easy as in factory. In addition, the scale of the manufactured elements also presents other practical challenges in terms of precision. Design-to-file is also not seamless and continuous. For instance, changing software and transfer of design information are some factors of computational design process.

Digitization

Digitization is a phenomenon that is central to computational design thinking, both philosophically and practically. Computational design thinking originates from the belief that any physical elements, or abstract process, decision-making or thinking system can be modelled within numerical quantities. In parallel, any digitized model can be transferred very easily to the manufacturing mediums. The participants, who stated production-related reasons for preferring computational design, also mentioned the issue of digitization. P11 and P14 state as follows:

P11: *The most important aspect for designers using computation is the translation of each elements to numerical values. This then enriches the* *digital model with further embedded information and facilitates the smooth and detailed fabrication of the architects design.*

P14: *Computational tools allow for interpreting numerical data into design. This gives the possibility for more efficient designs, which fulfill specific parameters.*

Digitization can be understood as a form of representation, wherein representation itself encompasses the purposeful transformation of knowledge. In the context of digitization, information or data is translated into a digital format, allowing for its storage, manipulation, and transfer. This process involves converting real-world objects, concepts, or phenomena into a digital representation that can be accessed, analyzed, and shared through various mediums. Representation, in a broader sense, involves the intentional act of transforming knowledge from one form to another, often using symbols, signs, or languages as vehicles of expression. It serves as a means to communicate, convey meaning, and capture the essence of ideas, concepts, or entities. Through representation, complex and abstract notions can be distilled into tangible and intelligible forms, enabling comprehension, interpretation, and engagement with the represented subject matter. It provides a bridge between the conceptual realm and the perceptible world, facilitating understanding, analysis, and communication of knowledge across different contexts and stakeholders. Whether it is through digitization or other modes of representation, the transformative nature of representation allows for the transmission, preservation, and evolution of knowledge in diverse domains, including design, art, and science.

Evaluation

Five of the participants stated that they preferred computational design tools because of their advantages related to their evaluation capabilities. It has been seen that computational design tools are attractive to the participants in terms of testing, optimization, and simulation opportunities in the design process. Participants mentioned that computational design tools offer the opportunity to improve the design by focusing on certain points. Instant feedbacks and sophisticated design evaluation make them attractive in comparison to traditional design tools. Contributions of the participants P10 and P14 below might illustrate this:

P10: *it extends my creative ability to look for the sweet spot of my design ideas. I can create more options and evaluate them better for the aspects I like to see in my design work -whether energy, cost, or spatial performance.*

P14: *Computational tools allow for interpreting numerical data into design. This gives the possibility for more efficient designs, which fulfill specific parameters. If the parameters have been set correctly by the designer, the final result is proved to be optimized to the given criteria. Moreover, the parametric drawings allow continuous instant modifications, which is crucial when you have to face strict deadlines.*

It could be inferred that designers shape their ideas around fixed expectations. They search within a specific design space by checking and positioning themselves in terms of their reference points. Computational tools help them find the optimum position that keeps a balance between tolerance and flexibility. In addition to that, there is a mutual relationship between computational tool and designer. Reflexive and interactive actions direct the design process until the designers' expectations are fulfilled.

Mind Mapping Medium

Computational design tools facilitate the mapping of intangible thoughts, abstract data, or ill-structured initial design ideas, bringing them closer to visual representations of architectural space. This could be achieved through the use of digital or analog tools, both computational and non-computational. The first step towards mapping ideas in the computational design phase is to visualize them through algorithms. Visualization is not just an issue of representation but also closely related to digitization and evaluation. Computational design tools brought a new search of visualizing abstract data in graphical language. It is a kind of transition platform for intangible thoughts becoming representable and representable thoughts for becoming concrete spaces. This dimension of computational design tools was emphasized by participant P1:

P1: *I use computational tools as ways to visualize abstract data into some kind of graphic language, then to experiment with these outcomes as they are gradually being translated to space. They are mediums used along with other ones to express, process, and communicate ideas and as such they* *may help think in different directions. Moreover, they offer great computational power, which allows to perform alternatives through parametric and generative processes. In effect, they can have a great influence in design concerning creativity.*

Additionally, mapping of mind by the visualization capacities of computational tools are utilized in the design evaluation processes. Integration of visualization feedbacks creates dynamic modification loops. Participants P2 and P17 declared similar views as follows:

P2: *We use computational tools because sometimes it is more efficient and helped in the generative process, also it helps visualizing, we use it a lot for testing variations on possible solutions, also in the constructive process, not necessarily to enhance creativity.*

P17: *Having an algorithmic design process, computational design tools enable the visualization of complex geometric relationships based on simple mathematical relationships.*

It could be highlighted that mind mapping is a way of self-thinking for participants. They upload their design considerations to the design tool for visualizing them meaningfully. Digitization and evaluation via visual representations of computational tools is an intrinsic process of designers. It is precisely where the essence of the design action takes place.

Complexity

Complexity, as one of the most emphasized terms, presents confusions in terms of as to whether it is a cause or effect. It is even possible to attribute positive or negative meanings according to the place where it is used. Complexity has generally matched with geometry, form, and design by the participants. Indispensability of computational tools stems from the need of managing complexity. Participants P4, P13 and P6 indicate complexity as a common:

P4: *I use computational tools mainly because they are currently the only method in realizing my complex design goals. Or possibly - it can be argued*

- that my design goals have become more complex because of the use of computational tools. However, the underlying personal motivation towards technology and complexity (in formal and visual aspects) has veered me towards the use of algorithms and scripting. And once you get into it, there is no turning back: the level of freedom they offer is unparalleled.

P13: *It allows complex form creation and consideration. It automates certain pattern development. More simply, it introduces ease of modification of the result over paper solutions.*

P6: *The use of algorithms allows for much more geometric complexity, information-based forms, and playful explorations - sketching alone could not offer. Additionally, computer-aided-manufacturing can now handle this complexity - which a human worker before couldn't.*

The interdisciplinary paradigm transitions between architecture, math and science have always been experienced. Mathematics and computer science try to deal with complexity in well-defined patterns within their own disciplines, however in architecture which already includes ill-defined problems it become over-complicated and a faddish hybrid zone.

Impossibility

In theory and practice there are some facts which are not possible without computational tools. Envisioning particular geometries and representations are only possible through computational and mathematical expressions. In practice, the production of some complex forms is not possible without these integrated technologies. The views of the participants draw a boundary between before and after computational design tools, showing that they are perceived as two separate periods. P14, P9, P6 and P5 are examples of this view:

P14: *We do not believe that computational tools enhance creativity but rather extend the designers skills in visualizing and evaluating his/her ideas and subsequently materialize complex geometries, which is not possible without computation.*

P9: *The goal is newness. To create architectural form that has hitherto not been seen, that could not have been produced, that could not have been conceived of. Computational tools to expand the mind and imagination.*

P6: *The use of algorithms allow for much more geometric complexity, information-based forms and playful explorations - sketching alone could not offer. Additionally, computer-aided-manufacturing can now handle this complexity - which a human worker before couldn't.*

P5: *That said, we (in my office and in my teaching) continually rely on computational tools and scripts to explore new possibilities, in ways that simply might not be possible to image without these tools.*

The expressions of the participants indicate that computational design tools started a new era both intellectually and practically. It is felt by participants this era came with a sense of pleasure from benefiting from new opportunities that did not exist before. The use of computational tools, in this regard, is self-justification since they now exist and are ready to be used.

Becoming the Standard of the Practice

Last but not least, digital design tools have become one of the most common design tools of the practice whether they are used for their computational capacities or not. The determinations of the participants P8 and P16 regarding the situation are as follows:

P8: *The reasons I use and have used computational tools are for several reasons. Firstly, enhanced precision - computers allow degrees of accuracy and freedom of exploration not before achievable in formal explorations. Secondly it is becoming the industry standard in almost every facet of the design process.*

P16: *They are both. I mean it is really important to have the complete control of the geometries and to give freedom to the designer, so I can say computational design tools are now a basic standard tool when we work in complex geometries field.*

It is worth noting that the standardization concept mentioned by the participants overlap with the current global economic system. They could be named as the new customary tools of the practice.

In this section the headings related to the pragmatic reasons of computational design tools are presented. In addition to becoming the standard in the practice, production, digitization, evaluation, visualization, complexity, and feasibility concepts are mentioned by the participants. All participants who commented on the pragmatic reasons for preferring computational design tools deliver positive opinions based on the new possibilities and capacities that computational design tools offer them.

Architectural Space and Spatiality

When the answers of the first question were examined, it is determined that five participants commented on architectural space and spatiality. As a living and transforming entity, the concept of space had been exposed to radical changes through architectural history (Schwarzer & Schmarsow, 1991) and computational design tools are also expected to cause transformations on our conception of architectural space. Although the opinions are not expressed extensively, the findings related to architectural space can be seen in some of the reasons. Emerging views on architectural space while explaining the reasons for choosing computational design tools help to trace different understandings and approaches in architectural practice.

Designerly Ways of Computing

One of the findings about the understandings of space is conceiving it through its constituting elements and the relations between these. According to this approach, physical space consists of combination of parts and the structure of relations between them. This piecemeal bottom-up network of relationships composes the space. In addition, these components have behaviors where design decisions are embedded. Lastly, designerly representation of spatial design concepts (Cross, 1982) via algorithms and software needs a deliberate intention. Responsibility for the architectural space is imbued in the designer rather than the tool. The statements of the participant P12 could be emphasized in terms of describing such an approach:

P12: *Yes it is more efficient, yes it facilitates the generative process, yes it helps visualizing, and yes it enhance creativity on certain conditions. And it helps spatial thinking but only when the designer thinks from within the process, meaning that the digital designer needs to define the relations between all the components. Since the components define the physical boundaries of a space, whether open space, semi-open space or enclosed space, setting the rules of the behavior of the smallest building components includes the spatial design concept. Yet I must say that not many are aware of this and build relationships blindfolded not knowing what exactly how much they fall back into conventions without even developing their own view on what these relations could be. So, the proper answer is that it depends on the state of mind of the designer, spatial design concepts reside between the ears, and are then transferred to algorithms and project specific software.*

Participant P12 is critical of being controlled under the limitations of the design tool through the design process and the blinding of spatial values. This approach shows the search for a balanced and transitive collaboration between the use of design tools and the search for spatial values. This cooperation requires competence and awareness both in the use of computational tools and in conceiving spatial qualities. These are already within the quest of contemporary architectural education/formation planning.

Distorted Space

Another issue that arises regarding the imagination of space is that the relationship between the designer and the representation happens through the two-dimensional screen. There is a risk of misinterpretation of the space caused by the false sense of vision while working with the digital 3D model on 2D screens and the expected result is the incompatibility of conceived (imagined) and perceived (built) space. Additionally, software representations, which are isolated from physically and contextually real-world conditions, can lead to alienation considering the spatial qualities. Participant P6 made the following statements addressing similar concerns:

P6: *I think seeing in itself is not spatial - it is more two-dimensional representation of a 3d space in our brain. Therefore, I think the model* *doesn't need to be spatial. A digital 3d model allows to explore the future space - from first-person viewpoints - with the danger of seeing the building as a floating object. The use of algorithms allows for much more geometric complexity, information-based forms and playful explorations - sketching alone could not offer.*

Sensorial decomposition, which means dominance of vision and exclusion of other senses, and isolated design environment, which means scale and context free platform, create a distorted space. The designers cannot accurately represent what is on their mind within such a distorted environment. There are problems related to the calibration of the space they see on the screen and the space to be built in real life.

Pre-computer Spatial Considerations

P5 emphasized this distinction by mentioning that he was an architect in the pre-computer period. Some faculties like figure and form, space and atmosphere, material and affect regarding space he had in the pre-computer period were specifically defined by him as follows:

P5: *Like sketching and physical models, per your prompt above, computation tools are a means to an end. I became an architect before the use of desktop computing and drawing. As such, I am used to thinking about figure and form, space and atmosphere, material and affect, without feeling the need to "go to the computer" to explore what is attainable. That said, we (in my office and in my teaching) continually rely on computational tools and scripts to explore new possibilities, in ways that simply might not be possible to image without these tools. Design intent directs the tool, but the tool also has the power to challenge initial ideas, or to open up new directions.*

Considering the history of architecture, how little time is compared to the time when computers are used. However, considering its widespread use and acceptance, this favor shown by users is not only due to technological success but also dependent on economic and social dimensions. Since the exhibition of Rudofsky (1964), which brings a new perspective to modern architecture, 'Architecture without architects' has been possible but architecture without computers in contemporary architecture is questionable.

Emergence of Space via Design Process with Computational Tools

Although this approach is examined under the title of visualization, it can also be discussed as the emergence of space. From this point of view, space is understood as the outcome of the abstract data. Abstract data as an input is converted into graphical expressions with a numerical language. There are back and forth actions and experiments through the trial and error and design research. At the end, architectural space emerges as an output:

> P1: *I use computational tools as ways to visualise abstract data into some kind of graphic language, then to experiment with these outcomes as they are gradually being translated to space.*

Identification, representation, transfer, and transformation of architectural data is a complicated process. It is thought that these processes could be examined in computational design where procedures need to be explicit, transparent, and traceable. Seamless transition from completed design to production is a succeeded phenomenon, yet in the design phase the processes of exporting the space as the final product are not equally matured.

Nuances concerning the architectural space and spatiality are pointed out in this category. Designerly interpretation of computation, the distortion of architectural space, pre-computer spatial thinking and emergence of space within computational process are explained in different clusters.

In the first question of the interview the preferred reasons of using computational design tools are investigated. Prominent reasons are gathered in different clusters and grouped in two categories as imagination related and pragmatic reasons. The issue of architectural space and spatiality, which the thesis focuses on as a research topic, is also presented as another closely examined category.

4.1.2. Examination of Question 1.2

The question 1.2 is as follows:

1.2. In your personal design process, how do you benefit from or use these conventional spatial representations and the newly emerging computational tools? Do you sketch something first, or do you do modelling / coding / scripting first or do you use all of these simultaneously? If you prefer certain tools explain how they are used during your design process and if you prefer more than one tool how and why you go back and forth between different tools?

The second question aims to examine the computational design process through the use of design tools. The answers to this question helped in investigating the kind of decisions made in the design process, their rationale and when they are made to understand which design tools are preferred at what stage and the transitions between design actions. Participants were asked to explain different uses of design tools, and any instrumentalization and customization methods. The question of whether they continue with a single design tool or not, and how they switch between tools was also asked to the respondents.

The answers to the second question indicate many points regarding the participants' expectations from design tools, their order of use and different paths of solutions. The chronological order in the use of design tools revealed which tool is predominantly used in the conceptual design phase. One of the most central findings is that participants use more than one tool together. They change their design tools by switching to a different design tool and/or converting the design tool with a different purpose. The main point here is to reach solutions that are refined enough to meet the expectations in the design process.

Findings of the second question indicate the prominence of the integrated use of different design tools, switching between design tools, chronological order of design tool use, design of the design tool, and the refinement of the design.

4.1.2.1. Integrated Use of Different Design Tools

The answers given by ten participants to the second question indicate that the use of different tools simultaneously is the most prominent topic of this section. Participants explained how they understand, use, and interpret design tools in their design.

Knight and Stiny (2001) propose a matrix to categorize different strategies of computation. Following their categorization, a similar trajectory is employed in this study. When the usage patterns of the participants are examined the design tools could be classified as analog and digital or they could be classified in terms of user's manner of thinking and utilization of the design tool as computational or non-computational. Designers switch between design tools which are analog or digital and which are utilized within computational thinking or not.

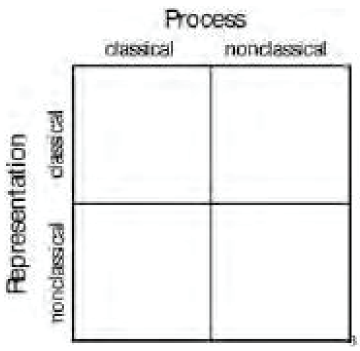

Figure 11. Knight and Stiny's categorization of computation.
(Source: Knight and Stiny, 2001)

The integrated use of tools arises from different reasons and manifests itself in several different combinations, forms, and methods. Different types of the integrated use of different design tools are related to complementary, combinatory, integrated, simultaneous, and articulated use of these tools. As shown in Table 4, the fact that design tools have different technologies is not an obstacle to their different utilizations. As noted by some participants, an analog tool could be used computationally, or a digital tool could be used non-computationally.

Table 4. Classification of design tools.

		Design Tool Technology	
		Analog	Digital
Design Tool Utilization Concept	Computational		
	Non-computational		

Complementary use of different design tools is clearly a common way of usage in the design process. Designers benefit from different tools within different combinations for example analog / computational tool with digital / computational or digital / non-computational tool with digital / computational. Different needs on the different phases of the design process prompts designers to alternate their design tools. P17 and P4 emphasize this as follows:

> P17: *I start most design processes by sketching a simple idea which represents a pseudo code of mathematical relationships. Sketching out all options based on the pseudo code becomes nearly impossible because permutations and combinations of iterations can become very complex quickly. The computational tools allow for starting with a logic, observe the outcomes and quickly revise the logic based on a design intention. In my design process sketching, scripting and 3d modeling take place simultaneously until I reach a fluid system for iteration.*

> P4: *My design process usually starts with an idea formulated through manual sketching or a mental image. Quite quickly after that I try to move to 3d, either by manual modeling or small form-creating scripts. Depending on the idea. My main tools are Rhino and Grasshopper, but often I have to use RhinoScript or Python also. I use Processing and Java also, but not in formal or structural architectural design. 'My design process uses all*

available methods simultaneously, and often is a mixture of 3D, rendering and sketching.

Different combinations of design tools reveal designers' patterns of use. Although usage patterns may be different, it can be concluded that instead of absolute and incompatible design tools, designers might prefer more integrative and articulated ones.

Another finding that can be observed relates to the types of tools articulated, matched, and combined during the design process. Analogue and digital tools are different technologies and have different capacities. Some of them are tangible and physical objects while some of them are intangible and nonphysical, software for instance. Participants utilized these tools sometimes in a computational or sometimes in a non-computational manner. The focus here is to realize there are many possible combinations independently of computational thinking.

Table 5. Analog and digital tools.

Analog Tools	**Digital Tools**
Pencil, paper, eraser, ruler, miter, level, etc.	Mouse, keyboard, screen, computer, software, etc.

Considering the tools used in the design process there are many combinations e.g., analog-analog, analog-digital, digital-analog, digital-digital. Participant P2 stated that after starting the design by sketching with pencil and paper, she continued with digital tools which describes a transition from analog to digital tools. There is also transitivity between different digital tools, especially between different software as stated by P2:

P2: *We start with sketching and as we work in an networks in different places we brainstorm on that basis, but very soon we jump into the computer, physical modelling comes far after. If needed, we start scripting to test variations on concepts.*

While creating personal specific design tools with different combinations, more complicated and hybrid tools have arisen with continued evolution. For instance, CNC

(Computer Numerical Control) machines are a fusion of analog and digital tool. New tools emerged and are shaped in line with needs and demands.

An important point to consider when examining design tools is the form of action performed with the tool. Design tools can be examined in terms of the actions they take and the way they perform these actions. There are some moments when the designer delegates the action to the tools which ensures a duality, partnership, and task sharing. In the last instance, design is an output of the articulated actions not of the tools.

Table 6. Tools and actions.

Tools Analog and Digital	**Actions** Direct-Indirect /Active-Passive
Pencil, paper, eraser, ruler, miter, level, mouse, keyboard, screen, computer, software, etc.	Sketch, draw, model, erase, measure, calculate, code, generate, etc.

Integrated use of different design tools is another type of complementary usage. Different services of design tools are combined partially to get the expected result. This innovative attitude of designer could be related to creativity and customization concepts also. Participant P3 expressed an explanation that can serve as an example for this idea as follows:

> P3: *As I'm left-handed, sometimes I sketch with left hand, while using a mouse with right hand at the same time. Sometimes I scan and digitize hand sketches to be used as inputs in Grasshopper.*

Designers create their own patterns of use in line with their expectations from design tools. They integrate different services of design tools to fulfill their needs in terms of representation or production. Using design tools with unusual and extraordinary methods could be one of the different sources of the originality or creativity in a way.

It is observed that the design tools are changeable at different stages of the design process according to different needs. Another point stated by the participants is that these usage patterns are flexible and changeable, not fixed. Ways of tool use are dependent on project types, conditions, and the designers themselves. Participants P16 and P3 express what their design tool choice decisions depend on as follows:

> P16: *the process changes for every project, it depend by a lot of things. We have to assume that digital tools are important, but we can not work without hand sketch and also physical model, in order to explore our drawing to reach the goal that we have in mind. Basically, it's a sort of loop, you can start scripting then sketch or vice versa, then modeling or detailing, it really depends by the task you are assigned or what do you need to do.*

> P3: *The tool choice and process highly depend on the required exploration research, available time, required precision and geometric quality, material parameters, fabrication methods etc... there should be a long list which changes in every design situation.*

Although the choice of design tool seems uncertain due to many different reasons independent of each other, it is not seen as a rigid and unchangeable choice. Participants' statements show that they arrange and set up the design tools in line with their expectations.

4.1.2.2. Sketching in the Computational Design Process

When viewed chronologically for the determination of the design tool at the outset, the sketch is still the most dominant tool. Indispensability of sketch as the first design tool based on many factors. It has more superior aspects like indirectness, practicality, and familiarity when it is compared to other design tools. It is seen in the following explanations of the four participants that sketching gains a new different layer while used in different dimensions within the computational design understanding:

> P14: *At the early design stage we use sketches in order to specify the arbitrary forms and intentions of the design. Right after the idea is conceived, we proceed with visualizing it with computational tools. The evolution of the computational model is accompanied by hand sketches throughout the project.*

P11: *In general, everything starts with a draft sketch about the process. If the exploration is geometrical, then more elaborate sketching and referral to precedents is needed.*

P10: *A sketch is usually the first thing to do. Actually, sketching and coding and modelling work in parallel.*

P17: *I start most design processes by sketching a simple idea which represents a pseudo code of mathematical relationships.*

Although sketching maintains its importance in the computational design process, now it is a hybrid form of computation and drawing. This assumption can be made that sketching has turned into a search for principles of computational design thinking rather than a search for an imagined architectural space. Geometry, mathematics, and coding are accompanying concepts of sketching in architectural design.

Feedback loops between the forms which are digitally executed from the manual sketches, as preliminary codes, are a new way of making sense. A code may become sketchable and, eventually, an executable fact in the design process. Sketching has become an integrated process with the use of computers with the fast feedback capacities. In the customary sketching process, the feedbacks are only between the designer and the sketch, while in the new case the computer is also involved as a third agent. This situation could be seen in the contributions of the participant P7 and P5:

P7: *I do sketching and scripting simultaneously. In many cases the sketches are preliminary to how geometry will be later coded in an algorithm. Alternatively, I look for algorithms that could be modified to solve a certain design problem. I use python and grasshopper extensively in my work and it's often an iterative process where an initial idea is improved procedurally to arrive at a generative system. So, moving towards design from both algorithmic and sketching avenues is possible.*

P5: *I rely heavily on my internal visualization in my process towards developing new designs. This is not intuitive, but a mental analysis of the scope of work to be addressed. Once certain possibilities and problems* *present themselves, then computation is used to respond, as we would with a model or sketch. But computation is embedded with such a sophisticated feedback loop, and the possibility for generating alternatives, that leaves it the sketch behind in terms of real usefulness.*

Sketching with a computational point of view opens new kind of feedback channels in the design process. Thanks to the rapid feedback possibilities of computers, the early design phase sketching process become available to quickly test the future design and production phases. Issues like form, production, material, or cost become more predictable. The new way of sketching in the computational design process goes on as a triple interactive trial and error process which is carried out between the designer, sketch, and the computer.

4.1.2.3. Design of the Design Tool

Changing the design tool has a dual meaning and the first is explained in the integrated use of different design tools section. The second meaning relates to the design of the design tool itself. As participant P9 states, developing the design tool is also seen as part of the design process:

P9: *First creating a tool, then using it to create a design or to create permutations of the design. ... And in many cases, the tool is successively modified or expanded to enable a new feature in the design.*

Problem-specific tool development progresses similarly in digital media as well as in the Problem-specific tool development progresses similarly in digital media as well as in the physical world. Software packages are the main tools in the computational design process. Many prominent software packages which are used in the computational design process are platforms for developing digital tools via coding or drag-drop interfaces. If software is a toolbox, it could be argued that there are rigid, customizable, and yet undesigned tools within it. Especially in the computational design process participants develop their own components to be used for specific tasks. Participant P14 suggests:

P14: *We mainly use visual programming (Grasshopper) and for more complex projects we use programming in order to create our own components.*

Moreover, the limits, capacities, and insufficiencies of design tools could be defined as the reasons of complementary and combinatory use of design tools, but they also necessitate the design of the design tool. Designers' customization of and innovation in design tools in accordance with their needs and expectations creates hybrid design tools. The expressions supporting this argument come from P1 and P6:

P1: *I work between the analogue and the digital. Analogue doesn't have to be conventional. Models and sketches may be treated in experimental ways and with materials that manifest their performative properties. As such, they present generative aspects too, which may further be combined with similar computational processes. Anything may come up first so it doesn't matter where one starts from, but most importantly each one should inform the other. Design is an iterative process that involves recursive testing happening across different platforms, analogue and digital alike.*

P6: *I use the tools for what they are good in. (I see dogmatic using of tools in digital and non-digital architecture): I use sketching, were I want to be as fast as possible, want to combine different models of thinking in an intuitive way. I use parametric tools for linear geometric deformations and coding for non-linear form explorations (can be hard to translate into architecture and to keep control). I like to make physical models - but I think they becoming obsolete and it is more about nostalgia and people - including me love to see them.*

As can be seen in these statements, the intended use of the design tool can regain a different meaning from the user's point of view. The direct and obvious meaning of the tool could be reinterpreted in line with the expectations of users.

4.1.2.4. Refinement in the Design

Designers organize their design set ups to achieve a refined result in terms of their design idea, tool, or the end-product. They look for a sufficiently refined solution which fulfills their expectations. This quest could be for the representation of any design idea, the definition of the precise geometry of the imagined form, the customization of the design tool or for a precise production process. The refinement of the tool and the design at the same time is an integrated process. The refinement of both the design and the tool is an interconnected process that involves simultaneous progress and advancement. Participants P9 and P17 describe the process which is defined as the refinement in the design in computational design practice as follows:

> P9: *It's a cyclical iterative process. First creating a tool, then using it to create a design or to create permutations of the design. The design is iteratively refined. And in many cases, the tool is successively modified or expanded to enable a new feature in the design.*

> P17: *The computational tools allow for starting with a logic, observe the outcomes and quickly revise the logic based on a design intention. In my design process sketching, scripting and 3d modeling take place simultaneously until I reach a fluid system for iteration.*

The process of refinement, which means testing or exploring the early design ideas until their physical or digital representations generate meaningful feedback is more efficient when it is compared with customary non-digital design tools. Computational design tools enrich the interaction in the design process via iterative feedback loops which create an intensive interaction environment.

It seems that the computational capacity increase which is provided by computers offers versatile and much more feedback. Feedback, which are iterative coevolution actions occur between designers, representations, and the tools. Informative design feedback loops define a kind of design data transfer/interaction, which occurs between the designer and the representation, the designer, and the tool, or between the tool and another tool. Statements of participant P1, P5 and P12 could be examined in this context:

P1: *Anything may come up first so it doesn't matter where one starts from, but most importantly each one should inform the other. Design is an iterative process that involves recursive testing happening across different platforms, analogue and digital alike.*

P5: *I rely heavily on my internal visualization in my process towards developing new designs. This is not intuitive, but a mental analysis of the scope of work to be addressed. Once certain possibilities and problems present themselves, then computation is used to respond, as we would with a model or sketch. But computation is embedded with such a sophisticated feedback loop, and the possibility for generating alternatives, that leaves it the sketch behind in terms of real usefulness. The sketch will not generate a project for me, but can help seed directions.*

P12: *Our design to production procedure links the design intent directly to the manufacturing methods in a bi-directional way, as to inform the design process by the capacities of the CNC machines. This bilateral machine to machine communication is based on lean data exchange as to avoid that information gets lost in translation.*

The origin of any architectural process starts with converting the ideas, values, experiences, or intentions into representations via design tools whether they are digital or not. Capturing, developing, and representing the design idea until it becomes a final product requires the awareness of the designers. At the end of the multivariable feedback loops, the architectural product is a refinement of selective and conscious accumulation of tangible and intangible data in the design process.

In the second question of the interview four categories came into prominence which are integrated use of different tools, reconsidering sketching in the computational design practice, design of the design tool and refinement process in the design. Ways of utilizing computational and other design tools are explained in detail in these categories. Alterations and innovations in the design process that come with computational design tools are clarified.

In the first section of the interview dimensions of the tools and design process are inquired into. Emerging codes are clustered and categorized to figure out meaningful outputs. It is seen that computational design tools brought many innovative capacities and designers embrace these innovations enthusiastically, however old habits are combined and did not completely give up by the designers.

4.2. Introduction to Interview Section 2

The second section of the interview consists of three questions. These investigate the relationships between tools and spatial thinking. Before the questions, it is explained to the participants that every tool and every representational system simultaneously allows and inhibits different ways of thinking.) The main idea to be explained is the state of mutual interaction between two phenomena transforming each other. According to McLuhan (McLuhan & Fiore, 1967) 'the medium is the message'. He defines the medium and the technology as an extension of ourselves. From that point of view, tools contain messages and have effects on action. Despite performing the same action or reaching the same result, using different tools contain different messages. McLuhan (1964) states:

> "In a culture like ours, long accustomed to splitting and dividing all things as a means of control, it is sometimes a bit of a shock to be reminded that, in operational and practical fact, the medium is the message. This is merely to say that the personal and social consequences of any medium — that is, of any extension of ourselves — result from the new scale that is introduced into our affairs by each extension of ourselves, or by any new technology."

It is expected that thinking and designing with different tools in terms of architectural space needs to have different meanings and messages. From this point of view, imagination of architectural space is at least partially shaped by the tools used.

The history of the concept of space does not go back as far as architecture. Although its philosophical expansions have existed since ancient times, the concept of space is became evident especially since 19^{th} century in the discipline of architecture. Despite this clarification, the concept of space has not been evaluated with a homogenous and rigid understanding but manifold and converting approaches. Therefore, spatial thinking is an acquired artificial way of perceiving the world which is interrelated with the tools which focused here is on computational design tools (Forty, 2004).

Designers and tools are in interaction. There is a mutual transformation process due to the preference of computational design tools related to architectural space. Different ways of reconciliation occur between the importance of computational tools and the importance of the quality of space while imagining architectural space in the design process.

4.2.1. Examination of Question 2.1

This question looks into the kind of contributions computational design tools brought to spatial thinking in the design process. Participants were expected to explain their personal relations with the design tools considering their own design characteristics. Particularly, it was expected that computational tools' supporting contributions to imagination and intellectual capacity in the design phase would be touched upon by the participants. The question 2.1 asked to the participants:

2.1. Could you describe the importance of computational tools for you while imagining architectural space with regards to your design philosophy, formal repertoire, building technology etc.? Do computational tools help you conceive an image that you cannot think, draw, produce without them or do computational tools help you enrich your imagination process?

While participants were asked specifically to comment on computational design tools and spatial thinking only some of their comments were directly related to architectural space, and most were indirect connotations or interpretations. First, spatial concerns in computational practice are shaped around the following topics: piecemeal understanding of space, relationist design thinking, computational configurations involving spatial thinking, and different understandings of space. The next issue relates to the manipulations of the design tools in the design. The incompetency in the use of computational tools and biased design tools were the emerging themes under this category. It is also noticed that after use of computational tools, new ways of making sense of space have been added which are exemplified as intellectual alterations and different understandings. One of the other inferences of this question is about the designers' self enhancement by computational design tools, which is specifically related to expanding design search space and increasing design options. The prominent findings acquired as a result of this question are explained in detail below.

4.2.1.1. Spatial Concerns in Computational Design Practice

There are participants who directly stated their thoughts on computational transformation resulting in different understandings of architectural space and spatial thinking. Digital tools, especially algorithmic tools, support a linear thinking style, however, design activity does not follow a linear pattern. Designers share tasks and delegate linear activities to computational tools and take on lateral thinking (De Bono, 1970) activities. However, the design of the design system is also the responsibility of the designer. Configuration of design decisions as an input/output system becomes an essential element of the algorithmic design process (Terzidis, 2006). This approach brings the issue of indirectness in the design process. Designer designs the system and then the system designs or outputs the design. Clear articulation, order and execution of design decisions are essential in the algorithmic process. The following example could be given of an approach that defines at least some part of the design action in this way as an input/output system:

> P17: *Any design process of an architectural space can be considered a relationship of input and output. Input contains all contextual conditions and output is the designer's choice response to these conditions. I use computation as a means of designing an equation that analyze big data and output relative solutions. The process becomes an autonomous tool to envision possibilities, and ease design decisions.*

This example also could be related to an approach that defines the architectural space as fragments and relational systems. Language and intellectual structure of computation theory is built on bottom-up principles (Stiny, 2006). Algorithmic thinking, which is a system of thought focused on breaking problems into small parts to solve and reaching the whole from the parts, is one of the fundamentals (Terzidis, 2006). Therefore, creating architectural space via setting relationships between pieces and small elements correspondingly overlaps with computational thinking. Statement of participants P6 and P14 could be evaluated in this regard:

> P6: *I think digital tools are good for thinking in systems and overlapping models but are really bad for developing good spaces - which I think are better developed through intuitive top-down decisions (sketching, models).*

> P14: *The computational tools have definitely created a new design trend but by definition computational design is not connected with complex forms but rather with design efficiency. Architects do not design forms but relations between spatial elements.*

Participants' contributions prove that most of the mental effort of designers is on "spatial elements" and "relations." Then, it is speculated that piecemeal understanding of spatiality and relationist design approach between elements might impede the holistic comprehension required in architecture. It can be mentioned that the order of priorities has changed in the architectural design process after the use of computational design tools. Quantitative concepts became more dominant in the design process with the advantage of being suitable for computational design tools. However, designers are also aware that they are in search of architectural qualities.

Geometry is a very powerful representation tool in computational design tools, as it is in conventional tools. The history of geometry has also deeply affected the perception and understanding of architectural space. Computational tools make it easy to interpret different spatial understandings through geometry. Containing the idea of space in computational configurations are related to different interpretations of geometry. One of the comments of participants declares that geometry in computational tools deals with solid and tangible surroundings instead of space which is a void:

> P18: *Computational constructs embody the design ideas in their configuration - so the idea of space has to be incorporated in them as well. Geometry has played a dominant role and has been object centered rather than focused on modeling the void.*

Representation of design ideas and architectural space with computational design tools especially via non-spatial medium like scripting/coding is a new phenomenon for architectural practice. The spatial ideas are intertwined within computational configurations instead of sketches or drawings. Architecture dissolves into codes and transits to a different dimension. Architectural qualities which are imagined by the designer are redefined and adapted to this environment. The designer indirectly reaches the architectural values which had been imagined at the end of these processes.

Computational design tools have enabled new expansions in the understanding of architectural space. Users have established collaborations with these tools according to their own spatial understandings, which in a way expose their unique understanding of space. The participant P12 affirms the transformative and enhancing effects of computational design tools as follows:

> P12: *And yes computational tools change the way I conceive space. In principle I stated to think of space as weightless, as being able to be rotated, put upside down, stretched, bent etc. Many computer manipulations have become part of my thinking, and I think that is a clear enrichment. I think much more precise now, and much more free from physical constraints.*

When the expressions of the participant are examined, it is seen that the basic principles -constraints- of the real world can be suspended within computational design environments. Considering the production of spaces, which are designed in experimental environments however to be built in real life, a calibration issue arises. Calibrating 'physical constraint' free architectural space with real-life conditions requires the control of designers.

4.2.1.2. Augmented Capacities

Computational tools easily catalogue all options, which would take too long for humans to handle on their own. Increase of the design options is a positive phenomenon regardless of designer expectations. To expand the design search space, these tools provide users with forms and outputs that are almost impossible to discover in terms of time and processing limitations. Participants P3 and P1 made parallel comments within this perspective:

> P3: *On creating emergent results; it is again not about the tools, but the designers and the design situations. A simple code can generate endless combinations of shapes that you can not think of, this is obvious. A harmonograph machine can also "non-digitally" draw endless shapes that you can not think of. It is not the tools but the geometry, mathematics and/or physics underneath…*

> P1: *Computational tools, as any other one used to experiment with, help me conceive an image that I could not have thought, drawn, or produced without them, therefore they help me enrich my imagination process. I don't agree with the 'or' word between the two parts in the proposed utterance.*

Computational design tools help augment the capacities of the designers and the design environment. Moreover, the capacity increase can lead to unforeseen and unexpected outputs. Hard to follow up complicated algorithms conceal the imagined end-form in the computational process. As a result, computational design tools are more likely to facilitate surprises than conventional design tools. The phenomenon of surprise in the design process is welcomed using all design tools but it sounds that increased possibility of the surprise phenomenon with computational tools is in a way one of the main reasons of preferring computational tools. Participants P9 and P7 welcome surprises that involve uncertainty due to complex and algorithmic processes:

> P9: *The processes are deterministic, but the output is not entirely foreseeable due to the number of steps involved, the number of inter-relations and dependencies, etc. Thus, there is an element of surprise. And this surprise is welcome – it is even invoked.*

> P7: *In my view, I think we should focus how to device novel algorithms that can open up new approaches to rethink form and design. When working with generative systems or scripts it often becomes impossible to conceive the end result prior to seeing them in action. Once they are executed, than it becomes possible to reproduce new rules and approaches.*

The design tool becomes an extender and a multiplier. Executing the design system and evaluating the design outputs is not a one-way and static process. Re-design of the design tool or tailoring the design output with feedback from the design tool is an interactive and dynamic iterative cycle. The interactive loop of computational design occurs between the designer, tool, and the representations of design which is similar to conventional design processes. However, the designer delegates the design action or authority to the tool while running the algorithmic digital operations with accepting surprises which direct the architectural end-product in computational design process.

The fact that computational design tools present unexpected possibilities and results as a surprise is accepted as supportive of creativity. However, computational tools contribute not only to imagination but also to the analysis and understanding of what is thought about. Participant P2 defines that they benefit from computational tools not at the early design process nor that they push creativity but more to develop and multiply the existing design options:

> P2: *Of course, it helps, for us especially in the communication for the building technology, only through the use of computational tools we manage to build what we design and think- not necessarily the computer helps to design what we cannot imagine - but it helps to analyse and to understand.*

In addition to augmentation of non-tangible concepts in the early design process, computational tools also bring new capacities to architectural tectonics. Computational design tools have changed production paradigms with analytical and integrated methods in the construction and manufacturing stages. Although there is no consensus on whether computational design tools increase creativity or not, one of the issues agreed on is that these tools amplify designers' capacities.

Last but not least, the delegation of design authority to computational tools via algorithms might be interpreted as the decline of the central role of the architect, which is a legacy since Modern architecture. The fact that the design action and decisions become transparent and invisible overlaps with the contemporary management systems that feels people unguided, manipulated, and free in giving decisions. In addition to all the ethical concerns from the past, it is obvious that the responsibilities and 'cognitive liberty’ (Sententia, 2004) of the architect will be reshaped within the frame of computational tools.

4.2.1.3. Bottom-up and Top-down Strategies

There are two universal methods of human thinking: top-down or bottom-up. These ways of thinking symbolize two fundamentally different approaches in architectural history. These two approaches are re-evaluated in relation to computational design tools and spatial thinking. Some participants define that bottom-up nature of computational tools open new possibilities in a positive way. Bottom-up design strategy means creating self-forming systems with rules and units rather than trying to achieve an ideal holistic image in mind (Knight & Stiny, 2001). Rule-based bottom-up design methodologies of the computational tools reveal all concealed/unrecognized but acceptable (and in accordance with the rules defined by the designer) design options. The following quotes of participant P11 exemplifies this view:

> P11: *Last but not least, computational tools offer the possibility of emergent designs, via rule-based bottom-up design methodologies.*

However, some participants claim that architecture and design need a holistic top-down approach and computational tools are insufficient in this respect. One of the participants compares computational tools to conventional sketching and modelling while citing the lack of intuitive top-down decisions. In a sense, this is one of the views that explains the inadequacy of computational design tools in ‘developing good spaces’. While expressing this view, the participant P6 suggests that both approaches should be synthesized and used together:

> P6: *I think digital tools are good for thinking in systems and overlapping models but are really bad for developing good spaces - which I think are better developed through intuitive top-down decisions (sketching, models). On the other hand, starting with a system allows you to discover spaces and forms you couldn't (or didn't) think of before. As always, I think it is good to have an iterative loop between both methods.*

From this point of view, it might be expected that computational design tools to evolve via including or integrating intuitive and top-down design decisions of designers. However, until then, the task of verifying bottom-up and top-down design decisions will continue as one of the responsibilities of designers.

4.2.1.4. Feedbacks

While executing the design process with bottom-up and top-down design strategies, a highly interactive feedback process occurs. Executing the design system and evaluating the design outputs is not a one-way and static process. Re-design of the design tool or tailoring the design output with feedback from the design process is an interactive and dynamic cycle. Design tools become an extender and multiplier. Participant P7 and P10 explanations could serve as an example for this situation as follows:

> P7: *Once they are executed than it becomes possible to reproduce new rules and approaches. In a way coded systems create their own feedback loops.*

> P10: *Any feedback (the creation of input to the designer) will influence and inspire design decisions and create new pictures in the head, which inspire and drive the design. as the computer is able to create this feedback instantly it has for sure an enormous influence.*

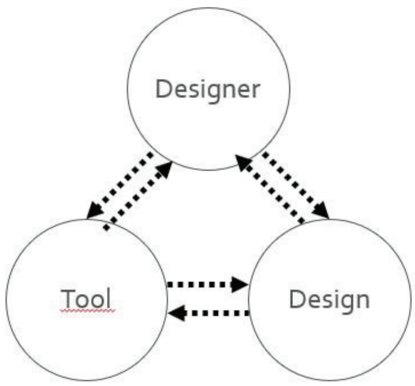

Figure 12. Feedback directions between designer, tool, and design.

The interaction loop of computational design occurs between the designer, tool, and the design which is similar to conventional design tools. However, while running the algorithmic process designer delegates his/her own control to algorithms without or partial knowledge concerning the architectural end-product in computational design process. In conventional design tools - making sketches or building a physical model- feedback responses are instantaneous and bodily but in computational design environments algorithms responses are detached, design action is intermittent, and feedback appeals more to the sense of sight at the early design phase.

4.2.1.5. Consequences of Incompetent Use of Computational Tools

There are different skill levels and mastery in using a tool. While craftsmanship is formed by the experience and knowledge gained in practice in the use of a tool, mastery requires being able to interpret and even criticize this tool. The state or feeling of being competent in using the design tool is a kind of self-proclaimed or self-proven situation. When considering computational design tools, there are warnings against some risks in terms of non-neutrality and assimilation. These are suggested as consequences of incompetent or unqualified use of computational design tools. The experiences reported by P3 is as follows:

> P3: *The key is that there are somethings in designerly thinking, which are free from all representations, I call "attitude". If you limit the extent of imagination with the available media, it becomes harder to develop an attitude which produces unique imagination. This is why a broad knowledge and experience in algorithms, scripting and geometry is very important for young architects. Not to do it quickly; but carefully and wisely.*

Non-neutrality of Design Tools: Biased or Customized

Computational systems, executed with specific programming languages, have syntax and semantics like natural and artificial languages. Computational design tools and software used in architectural design are also written with certain programming languages. Programming languages are developed with a specific purpose and point of view by their designers. Moreover, some architectural design software offers ready-made libraries which are defined as time-saving and easy-to-use. Experiences regarding the problems that arise in such situations is explained by participants P3 and P7 as follows:

> P3: *The "...our tools shape us" part is very risky and dangerous for me. I intentionally escape from Grasshopper when it starts to change my ideas and replace them with ready-made solutions radically. The most obvious example is the famous Voronoi component, maybe you know Grasshopper. In GH, Voronoi subdivision solution is very easily calculated by a few clicks. You don't need to understand why Voronoi developed this algorithm, how and why it should be used etc. This is why GH displays below warning if you use Voronoi component too much.*

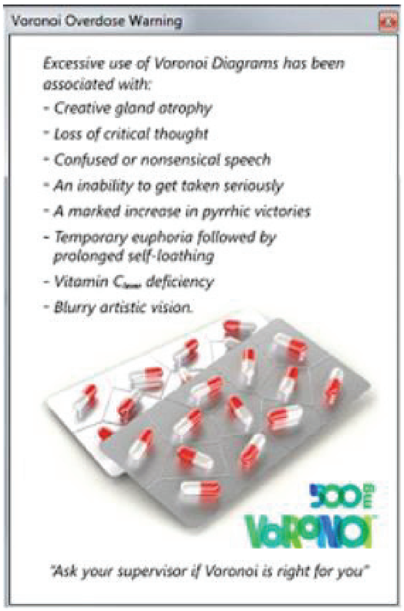

Figure 13. Voronoi overdose warning.
(Source: Image submitted by participant.)

> P7: *I agree with McLuhan that computational tools often provide certain constraints for design processes. Rhino, SketchUp, 3dsMax all have their advantages and certain ways to generate geometry. In my view I think we should focus how to device novel algorithms that can open up new approaches to rethink form and design.*

Participants find it problematic working without realizing these limited and fixed design sets. It is pointed out that designers should be aware of this realizing that digital design tools are already designed from a particular perspective, which is shaped with a specific mind set. Software-fixed design setup could cause limitations or over resemblance risk. It is suggested to develop a personal design attitude in terms of avoiding one-way thinking sets. Innovation and creativity are designers' responsibility, not in the scope of the design tool. Approaches of participants P11 and P3 about this issue are expressed as follows:

> P11: *It comes without saying that computational tools and ready-made implementation of spatial algorithms have enriched the designer's agenda. Are the designs that they produce truly innovative though? Gaudi was innovative and conceptualized highly complex, morphologically and geometrically, design without utilizing a computer. In my mind computational tools do not let you conceive a masterpiece you could not draw or perceive. They enable the designer to "draw" and fabricate his/her designs, no matter how complex these may be. In this respect, they do not provide a new imagination process, instead they free the designer from the constraints of complexity, fabrication and data management that wouldn't be possible with a traditional CAD design methodology.*

> P3: *As a geometry teacher, I aim to teach not only geometry but help students develop a personal attitude for design, which I believe is partially about shaping their own tool sets, not letting any tool to encapsulate their "imagination". The key is that there are some thing in designerly thinking, which are free from all representations, I call "attitude". If you limit the extent of imagination with the available media, it becomes harder to develop an attitude which produces unique imagination. This is why a broad knowledge and experience in algorithms, scripting and geometry is very important for young architects. Not to do it quickly; but carefully and wisely.*

According to participant P3, one of the consequences of the incompetent use of computational design tools is ignorance of users about the built-in biases of tools. Respondents' reservations show that inexperienced use of the design tool can lead to limited or self-enclosed design situations, just as excessive expertise in the design tool can lead to the same thing. For both novices and experts, extreme similarity and uniform thinking stood out as points to be considered. This situation, which can also be considered as a trend or style, can also turn into an architectural assimilation. Participant P14 explains the pressure created by the widespread use of computational design tools by using the term bombardment and the state of being mainstream using the term democratization as follows:

> P14: *Although, these new tools enable and democratize formal complexity and therefore we are bombardized, through social media and not only, with images of spectacular, complex designs. This is a derivative of the prominent computational design thinking of our generation, but it doesn't mean that we are more creative. Computational tools have definitely created a new design trend but by definition computational design is not connected with complex forms but rather with design efficiency.*

In the process of widespread use of computational design tools in architecture different types of users emerged. One of them is the group of people who are familiar with digital technologies, yet their formal education in architecture schools did not include any computational tools, then they merged them within their architectural profession. Participant P5 is one of the members of this group, who addresses how meaning can be conceptualized in architecture by digital tools:

> P5: *In this case, I suppose I am arguing for a concept of architecture, not for a concept of tool. Tools matter, but if I read between the lines of your initial abstract, there may exist a dubious reliance on digital tools to conceptualize meaning (space) in architecture today. For me, digital tools are important, necessary, super interesting, revolutionary, radical, and new: but as I've said, remain a means to an end towards a work of architecture.*

While making inferences about design tool-oriented thoughts, it is seen that what is crucial is how they are integrated with the spatial and architectural concerns and expectations. As stated by the participant P5, the main purpose here, however very important, is not a design tool, but architecture.

4.2.2. Examination of Question 2.2

In question 2.2, the initial phases of the architectural design are inquired. It is emphasized that the focus is on how the architectural space, which is formed because of many decisions, is shaped by design tools, starting from the conceptual design phase. With this question, participants were expected to explain their design process chronologically to detect in which stages and what kind of interactions are experienced between designers, design tools and the architectural space. It is expected that the different participants' characteristic design processes will enlighten the alternative understandings regarding the use of computational design tools in architectural space production. The question 2.2 asked to the participants as follows:

2.2. When do your spatial ideas and images emerge in your design process? Could you describe the process, relations, and negotiations between the computational tools and your understanding of space? How do the computational design tools foster spatial thinking? Could you illustrate it through examples of your own work?

Question 2.2 aims at the qualitative analysis of the transitions which are along with the chronological order in the emergence and development of spatial ideas. Emergence of space in the computational design process is considered in three phases. Architectural space emerges first as an intangible mental representation in designers' mind. Then it is represented in many ways, tools, and methods which have an active role in shaping the space. Finally, the built work of architecture is an ideal target where the bodily experience of/in space takes place. In addition to phases of architectural space design, participants mentioned different ways of thinking and designing which are basically comparing and referencing the bottom-up and top-down directions of architectural process. Descriptions of the participants while evaluating the architectural space draws attention in terms of their differentiated and hybrid jargon of the computational design practitioners.

Considering architectural space within data-oriented approaches is a concept often referred to by participants. In line with this interest, following the design data from the early design phase until built space is highlighted as a separate title.

4.2.2.1. Phases of Mental, Representational, and Bodily Experience of Space

Especially considering the architectural space, the design process consists of different phases. Space is experienced by designers within different modes; mental and representational and bodily. In the mental and representational experience of space all possible inputs are gathered. Feedbacks are short-term and self-examining. The relationship between the designer and the design is sterile. However, the bodily experience of space is an exposed activity. This experience indirectly participates in the design of architectural space in the long-term. The designer's mental sets are formed by his bodily experiences. From the early design phases to the divergent modes of spatial experiences (AR, VR, and MR) computational tools are integrated in architectural design. Participant P1 opens how computational design tools are used in mental and representational design processes as follows:

> P1: *The process starts from abstract collections of data and the appointment of abstract schemes such as grid, lines, surfaces to interact with the data. Ideas may emerge at each step performed to translate abstractions to more and more concrete graphics and spatial definitions. The produced space is a refined step along this process from abstract to concrete. Computational tools may be used at any step throughout the process, depending on what one may want to experiment with.*

Architecture as a discipline is based on projecting all possibilities in the process undertaken to achieve the desired result. Project is considered more successful when it progresses within the limits of predictions and reaches the envisaged idea. However, although in design a designer tries to consider as much as possible, the very nature of life may exceed all predictions. In the architectural production process, this struggle continues in a balanced way. Computational design tools serve to increase the control in architecture as much as possible, especially in manufacturing and construction processes. However, some users do not expect excessive control in the design process. They are not even in favor of reducing the gap between the final product and the represented design too much.

The participant positively demands the tension between the imagined and the realized. The views of the participant P5 on this subject are as follows:

> P5: *I try to design architecture in a way that I think will be impossible to model entirely in a computer. I want to be surprised when I see a work being built at full scale - to discover attributes, qualities, and relationships that are unexpected, unanticipated, and even problematic. In this way perhaps I am a hopeful formalist. I want the anticipation of looking forward to seeing something realized, and to have it exist (finally) on its own terms, and to teach me something. The more sophisticated our modeling software becomes, the more difficult it is to maintain this anticipation, since the built ends up being so similar to its previous representation. This gap bridging the representational to the real will only get smaller, to a point where we are forced to look for other unanticipated experiences through unforeseen unpredictability of human engagement, etc. This is not a physical (or atmospheric, environmental) space, since these can be modeled and designed with high precision, but a social space, which remain unpredictable, messy, and unknowable in time.*

In addition to the unevenness between represented and the built, there is another point mentioned which is the human experience in relation to space and time. Participant P5 brings up the concepts of physical and social space. He declares that architectural space may be precisely modelled and represented from a physical point of view, which focuses on the corporeality of the architecture with the computational tool capacities. However, he claims, the lived social space is very complicated to evaluate within the human cognitive capacity even with the support of the computational tools.

Spatial experience has different dimensions. One of them is singular, personal, corporeal, isolated, and the other one is plural, communal, representational, and exposed. When it comes to the experience of architectural space, integrating concepts that cannot be converted to computable values like experience-based issues into computational design processes is currently the responsibility of the architects.

4.2.2.2. Directions of Thinking and Designing

Space emerges in different ways in the computational design process. Many decisions related to the project are already given even before the start of design. However, two main approaches stand out in terms of design method or ways of designing, which are top-down and bottom-up. It is explained by the participant P7 that these two different methods require and refer to different ways of thinking. P7 defines that the bottom-up design strategy defines the space algorithmically and it is a generative approach while the top-down design strategy accepts space as a predefined entity. These two different processes are used for different purposes in computational design. Participant P7's views on the subject are as follows:

> P7: *In many of my public artworks* [Figure 14 and Figure 15]*, space is either defined through an algorithmic process, or a predefined space is tessellated for fabrication. In the first case the approach is more generative, in the second it's more decorative. In the example on the left, there is an idea of a canopy structure and there are some structural constraints. Using these inputs, the algorithm is allowed to generate a cluster that can connect multiple structural members in unison, so the design process is somewhat self-organizing or adapting to a predetermined field condition. In the example on the right, the final form of the pavilion is predefined which is then tessellated to fabricate final pieces of construction. In this sense it's more top-down and the notion or quality of space is more experienced in the final result.*

Figure 14. Opulent Grove.
(Source: Image submitted by participant.)

Figure 15. Skin Pavilion.
(Source: Image submitted by participant.)

Participant P7 designs and produces in both methods. In the last sentence, it is declared that '…the quality of space is more experienced…' in the top-down way of designing. From this point of view, it can be deduced that bottom-up algorithmic design processes might trivialize, devalue, and obscure spatial qualities.

4.2.2.3. Vocabulary and Language of Architectural Space Descriptions

According to Schmarsow the history of architecture could be categorized according to different understandings of space instead of periods or styles (Forty, 2004). Architectural space has many alternative definitions and each one of them is the reflection of a particular understanding of space (Moholy Nagy, 1928). The relationship between thought and language is considered by many philosophers like Wittgenstein, Heidegger, and Chomsky in different dimensions. The main admission of the philosophy of language is that the representation of ideas takes place through language. Considering the architectural space in the same way, any understanding or thoughts on space has its own vocabulary to explain their definitions. Changing vocabulary means changing attitudes and comprehensions.

When participants' responses are examined, there are some operations, actions, methods, vocabularies and interests, architecture related or not, influencing design decisions while shaping architectural space. Outsourced (spolia) terms are mostly referenced from the most interacted disciplines which are mathematics, geometry, and production technologies. Statement of participant P12 could be evaluated in this regard:

> P12: *Spatial ideas are more and more based on the experience of spatial arrangements in time and over time. For example, the A2 Cockpit building [2005] in the acoustic barrier in Utrecht in The Netherlands is based on the concept of a 60 second experience of driving along that structure. In those 60 seconds many things happen: one experiences the snakelike structure to become concave, then convex, then a featured fold line is introduced to tweak the reflections for the upper and lower part of the glazed barrier – using featured foldlines as car designers do - and then continues into the pumped-up volume of the luxury car showroom, then taken over by another foldline etc. Continuous variation and gradual transformations are key factors in our designs. This idea of spatial continuity is also represented in the interior where we made an endless loop, a spatial lemniscate as we did before in the Waterpavilion [1997].*

Figure 16. The building referenced by P12.
(Source: Participant's official website.)

The main idea of the referenced project is design of the 60 seconds driving experience on a highway. Experience and time are the eternal concerns of architecture which are also mainly considered here. The snake metaphor attributed to the building and the strong character of this approach can be seen especially at the façade. The point to be noted here is the words and concepts chosen while explaining the process of transforming design ideas into forms. Words like concave, convex, fold line, tweak, continuous variation, gradual transformation, endless loop, etc. are a small group of differentiated jargon of computational design practice. Terms and words are combined to form concepts, and these concepts crystallize and turn into different design approaches.

4.2.2.4. Data-oriented Definition of Architectural Space

From the early design idea to the final construction stage, the space is formed as a result of many interactions and decisions. There may be different anchors in different projects, for example, in some projects the form is dominant from start to finish, while in some projects the production method may have the leading role. Design tools can also be more dominant or recessive in some cases. Although they are used in the design process, they play a supporting role rather than an immersive one. Statement of participant P15 could be evaluated in this regard:

> P15: *There are several examples of computation applied to space and all of them consider different aspects. It is something strictly related to the data we want to analyze or simulate. I don't believe in a direct formal output from computation process usually, that's something that I consider more related to a 'traditional' approach if we are talking of proper shapes. I strongly believe this phase of the process needs to be informed and computational tools are a good way to achieve that.*

Computational design tools do not always help in the discovery of design's core concept but sometimes they are used for searching and considering any problem integrated with the main design concept. Some participants use them not to explore formal alternatives but for testing and training the feasibility of their set decisions.

Two points are worth mentioning here. First, the architectural space is evaluated via data (quantitative, digital, numerical, or computational) by designers. Architectural space is formed by the accumulation of data/knowledge along with materials and labor.

Second, in the process of obtaining the architectural space as a whole, partial and focused experiments are conducted. Architecture inherently imposes a transitional way of thinking between top-down and bottom-up working methods. Space also conveys the messages of a system that is both holistic and composed of parts. Designers need to evaluate the whole and the piecemeal consideration of spatial attributes simultaneously.

4.2.2.5. Design Data Exchange

The advantages of the file-to-factory are prefabrication, precision and economical optimization, efficient material use/consumption, instant feedback loops between design, engineering, and production practices. This concept is based on a seamless transformation of design data. Data exchange is investigated in architectural practice, particularly on the relationships between design and AEC teams working together through BIM programs (Abdelmohsen, 2011). Design data exchange takes place in various combinations between designer, tool, and design —for example, designer < > designer, design < > designer, tool < > designer, or tool < > tool, etc.—. Transformation of "the design data" in the process of design is common, however it needs to be re-evaluated with computational design tools.

Comparing conventional (sketching, physical modelling, drawing, etc.) and computational (scripting, digital modelling, printing, etc.) representation tools, the basic difference between is the indirectness, stratification, and discontinuity of computational processes. Although it is claimed that computational tools support a seamless design experience, the distance between the problem and the person widens. Designers must switch between different tools and environments. There is not a single tool that meets all expectations and needs. The learning time of these tools and the time and effort spent on specializing in the transition between tools should also be considered. Experiences of participant P17 can be evaluated in this context:

> P17: *It all starts with an abstract idea of a space driven by its qualities. Based on that initial idea, I establish rules and design an iterative process. Roboloom is an aggregation of digital objects. Digital refers to a finite set of rules, in which the objects are embedded with a connection logic, they*

can only attach to a neighbor by a rotational degree of 120. When I sketched out its logic, I represented the units as tetrahedron-the primitive form of the modified geometry. Sketching out by hand or in 3D-modeling software I was only able to conceive the aggregation of 5-10 units. However, sketching out this logic helped me understand the algorithmic relationship, and allowed me to put the aggregation in python to visualize the aggregation of the modified geometry in large quantities. Once I established a process, I was able to aggregate in large quantities while easily modifying the weave densities of each unit based on light qualities.

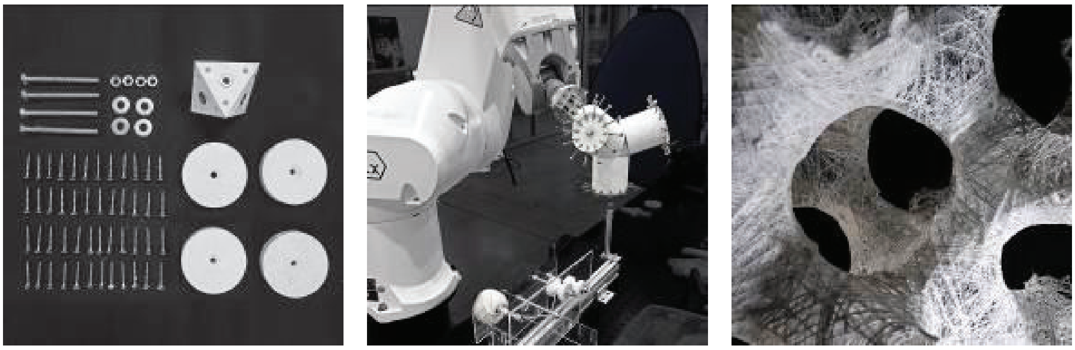

Figure 17. The project referenced by P17.
(Source: Images submitted by participant.)

Structuring and organizing algorithms to reach the imagined spatial qualities via considering material and production conditions sounds like a very architectural way of computing. Mathematical equations created during the coding phase and customizations made to production tools might be an example of the behavior of a designer who is aware of the space and expects certain spatial qualities. Prioritizing the criteria and qualities of imagined architectural space requires a certain level of experience in both scripting and fabricating. However, lack of experience is not the reason for every unsuccessful architectural work. Users who are fixated in their design ideas develop solutions by forcing these new tools to reach the result they imagine. Designers working with computational design tools face two stages in the process of putting their ideas into practice. In order to reach the spatial qualities demanded, it is needed to reach a certain level of scripting and means of production. Only in these circumstances the arguments of file-to-factory and seamless design environments of computational design tools could be useful.

4.2.2.6. Interpreting Design Tool

Although tools are produced for certain purposes, users can interpret the functions included in the tools in their own way. In fact, this interpretive behavior - using the design tool in an unaccustomed way or manner- is a sign of creativity, customization, and mastery. Innovation is another challenge in the process of architectural space design. Participant P1 gives an example regarding the re-interpretation of the design tool:

> P1: *... You use a knife to cut a hard surface or even to carve out material by interacting with the properties of knife and material, as you use computational tools to perform variations rapidly even to compute them by making use of the properties of the digital medium, which translate to easiness to compute fast and to produce alternative outputs. As such, computers are of great use during the initial phases of design, performed with abstract schemes to produce various outputs then explored in further detail, depending on the design aims.*

Similar creative tendencies in digital design tools are mostly seen as generating new tools or modifying the capacities of existing ones. Designer becomes a tool creator or developer in this case. While most designers work within the limits of existing software, in some cases, designers might develop their own custom digital tools which are utilized for specific problems. These plugins are usually developed within mainstream or open-source software packages. They include limited design parameters and service for well-defined design problems like partitioning a façade surface in certain proportions, tessellation, or fragmentation of specific geometries, iterative complex Boolean operations etc. This kind of package software offers an explorative test environment for designers. Designers could efficiently conceive and immediately handle the complex forms and geometries. The project developed by the participant P18 can be an example for this:

> P18: *In the cadenary project it is a process of discovery, the definition of design sets up an exploration space that is then explored through form finding processes from the interaction of topology and geometry constraints defined by the user.*

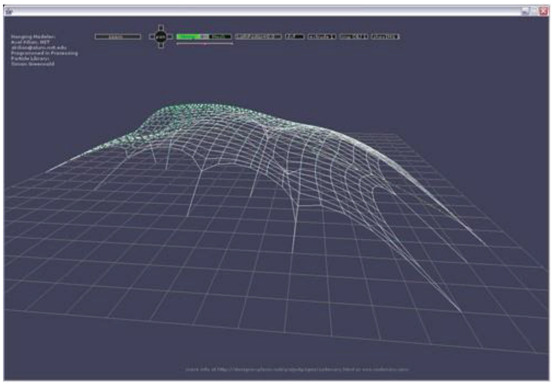

Figure 18. The project referenced by P18, CADenary tool v2.
(Source: Participant’s official website.)

Setting up an idea as an algorithmic model is one of the most important aspects of architecture with computational design tools. As seen in both examples, one of the new challenges of contemporary designers is to describe and structure the ideas as systems which are generative, integrative, and stepwise. The focus of design labor shifts from the design itself to the algorithms, which is scripted in artificial languages in the interface of the preferred software. A new design medium has emerged that requires a different evaluation and interpretation than the customary one. Architectural end-products obtained through algorithmic models or scripts are still evaluated with fundamental methods like Gestalt composition principles. These codes are developed within transparent platforms where design ideas and production methods are clearly expressed. However, the criteria to evaluate the algorithmic models and scripts, which are the products of the designers, has not been discussed yet. Efficiency, functionality, execution performance, inclusiveness, openness to integration and communication related criteria can be evaluation issues related to computer and informatics disciplines but, these evaluation criteria can be reconsidered from an architectural point of view. Just like the subjectivity of the designer's process of creating a space, the way they utilize and benefit from the design tool is also subjective.

4.2.3. Examination of Question 2.3

This question focuses on the quality of space and investigates the interactions between the designers’ approach and computational design tools in considering the quality of space. The question 2.3 is as follows:

2.3. What is the importance of the quality of space in your design approach and how do computational design tools ensure the quality of space?

Responses of the participants to question 2.3 revealed three categories which are titled as ensuring the quality of space, its integrative understanding within computational design process, and relating architectural qualities with computational quantities. Per the first category, the quality of space is evaluated from the perspective of both computational and customary design tools. Second, the views relating spatial quality to its components are summarized. In spite of the importance attributed to spatial quality, it is stated that this is ignored or remained in the background using computational tools. In the last category, the responses were discussed in relation to how qualitative values can be obtained with quantitative parameters were examined.

4.2.3.1. Ensuring the Quality of Space

The quality of the space is a depiction of the platonic understanding of Modern architects, like the allegory of the cave. It is accepted to exist but cannot be grasped exactly what it is and a kind of an ideal goal that every design tries to converge to. However, in practice architects look for a solution which fulfills the physical, climatic, economic, social, cultural, psychological, official, personal, etc. requirements while abiding by the oath of keeping quality of architectural space as the most important feature in design.

When asked how the spatial quality is ensured via computational design tools, a dominant negative view emerges in the responses. The answers indicate that the responsibility of achieving spatial quality belongs to the designer. Participants P12 and P7 concur at this point stating that computational tools by themselves don't ensure the quality of space:

> P12: *They don't, it is all in the instructions the designers give to the computational tools. Each line in the script is a design decision, based on pre-conceived spatial ideas.*

P7: *I am not sure if generative systems ensure a predetermined quality of space. In many cases it is really difficult to determine criteria that will both ensure spatial quality for the end result and provide an input parameter for how scripts would work.*

However, it also seems that this is not unique to computational design tools. According to many respondents, the design tools are neutral and have no priority over each other in terms of ensuring the quality of the space. Any tools don't ensure the quality of space but assist and support designers via realizing their intentions. The following two participants, P15 and P1, have made this point clear:

P15: *I believe the first thing is not the tool, but the quality and nature of the parameter to deal with, to reach the desired quality.*

P1: *No tool can ensure quality of space, but a tool may help translate data as qualitative aspects of design to a graphic or diagrammatic language and then to a spatial language, therefore bridging the gap between analysis and synthesis, or between the problem and its solution.*

The computational tools do not spontaneously and directly affect the quality of space. Designers orchestrate the connection between the design and the design tool. Although the relation between the design tool and the quality of space is intrinsic and dependent, this is not unintentional and automatic. Participants P4 and P13 express their opinions as follows:

P4: *I do not see that computational tools per se have a great effect on the quality of space. The computational tools are there to offer me means to express my desires - to visualize and to construct the forms and structure. The quality of space is a qualitative factor that cannot be measured computationally.*

P13: *I do not think that it is automatically ensured.*

Design validation is used instead of ensuring the quality of space by participant P3. This, however, also brings up a second question of how the quality of space can be verified:

P3: *It is hard to ensure the quality of space but I validate my designs by producing prototypes of them in 1:1. It is a personal case (I don't design buildings, just installations for Biennials and exhibitions).*

In terms of computability, finding an acceptable criterion for design validation is a reasonable method. For example, energy efficient buildings could be projected, calculated, started-up, compared with projections. However, it is open to debate which validation process is acceptable, especially for non-computable concepts. Space and design as ephemeral and multi-dimensional concepts are hard to validate computationally. To offer a solution, reference parameters of the design or the architectural space might be specified and projected literally before the design process or different projects and spaces could be listed as a design ruler to compare and detect more suitable, efficient, or fulfilling one.

4.2.3.2. Quality of Space as an Integrative Phenomenon

Quality of space is ensured with multiple computable architectural qualities which means that it is an integrated and accumulating phenomenon. Many decisions on different issues impact the desired spatial quality. The spatial quality is achieved by many factors. In computational design, the qualities of a mathematical surface, a process, a code, geometric research, etc might all have an impact on the quality of space. P3 states as follows:

P3: *There is the quality of material product, quality of a mathematical surface, quality of a process, or quality of a code... etc.. For me, the quality of geometric research is an important one. I think maybe the quality of space might be the sum of such parts.*

While some of the factors which ensure the spatial quality are new, some already well agreed upon factors like the spatial imagination, awareness of scale, tactile perception, bodily integration, etc. are relatively less visible in participants' responses.

Spatial Quality and the Disregarded Position of Space in Computational Design

Four out of eighteen participants mentioned the spatial quality as the most important constituent of architecture. The following excerpt from P16 is an example of these views:

> P16: *It' s really important the quality of space because it is the primary object of our architecture. Computational design tools are helping the designer to achieve what they have in mind and to control all the aspects of it.*

The participant quoted above had a positive opinion on achieving spatial quality with computational design tools, yet his view was in the minority.

While emphasizing the importance of spatial quality, three participants agreed that the importance of the architectural space is underrated in the computational design practice. Interestingly, the diminishing emphasis on the importance of space in computational design practice is expressed together with references to architectural education. While participant P10 criticizes the overemphasis on the spatial and formal in architecture, participant P18 criticizes the underrepresentation of the architectural space in his teaching process. Contributions of the participants are as followed:

> P10: *Spatial quality is of great importance in our work. While we are mainly producing architectural demonstrations and not buildings in our research center, we are still very keen on creating interesting and inspiring spaces, which are actually perceivable by visitors. This thinking might indeed be fueled by our architectural education in the 90s and we see that our own teaching doesn't have the same total focus on this mantra. (which might be good, as many factors make good architecture and the architectural education of the 90s might be criticized for a too strong emphasis on spatial and formal questions.)*

> P18: *Generally, space is underrepresented in computational design, but it is one of the most important qualities of architecture. So, there is a need to push things forward to put greater emphasis on space and not just geometric description of matter. This is something I have critiqued in my*

teaching in regard to the over reliance on geometry for design representation.

P6: *Space is a (!) core principle of architecture, but - especially in the digital architecture world - not the main focus. Aspects of architecture like form, construction-methods, behavior, playfulness, local social / political impact, and re-use... are equally important. (And I think it depends on the project: the space and the light in a church is maybe more important, while it is infrastructure in a hospital...)*

There are signs that interest in architectural space has waned in computational design practice. In addition to that, the opinions of the participants brought up the issue that adequate attention to architectural space is associated with giving sufficient importance to it during the architectural education.

Relating Architectural Qualities with Computational Quantities

Architectural space is designed via parameters, which are defined in algorithms in computational design practice. The computational design process is about setting up a dynamic relational system with precisely responsive algorithmic structures. Designers describe their concerns and priorities within virtual environments as logical inputs and the system reproduces outputs which meet the design requirements. Spatial qualities, which are desired to be achieved, must be defined in accordance with certain programming languages. Programming languages are artificial media, all based on mathematical logic. To realize the architectural design process within the inclusiveness of an extrinsic and abstract language with the uncustomary tools is a paradigm shift for the profession of architecture. Designers seek to achieve spatial quality while keeping algorithms executable and interoperable. Participant P7's views can be given as an example that explains a limited part of this process:

P7: *In many cases it is really difficult to determine criteria that will both ensure spatial quality for the end result and provide an input parameter for how scripts would work. One possible solution would be to define certain performative criteria such as optimizing solar gain for a space. Then the* *generative system would be run through an optimization protocol to achieve the most viable result.*

P17: *The computational design tools present a platform to easily modify an object based on desired parameters to precisely respond to function and context. Making of objects that precisely respond to these basics, allows finding a catalog of design solutions that foster spatial and functional qualities based on space and context.*

Associating Data and Parameter Quality with Spatial Quality

Computational design is described by some as if it is an information management system and computational design tools are utilized in the process of data manipulation. It is thought that the quality of the data or the parameter and the design/space quality are correlated and associated. In other words, if the data and parameters are well qualified (well configured and structured) then the design as a product will satisfy expectations as predicted. P15 defines data and information related quality expectations as follows:

P15: *As mentioned before, computational design tools are a way to manage data and to design through information. I believe the first thing is not the tool, but the quality and nature of the parameter to deal with, to reach the desired quality.*

Spatial quality in the computational design practice is a fragmented, integrated, and accumulated entity and not only related to qualities of tangible elements but also abstracted inputs.

Qualities and Quantities of Architectural Space

Qualities and quantities which constitute space in computational design practice are matched within space builder algorithms. Computational design practice utilizes computational design tools as a mediator to get the outputs. It is similar to the process of producing proteins from DNA chains through mRNA in biogenetics. There are processes called transcription and translation, DNA to mRNA to protein (Clancy & Brown, 2008). DNA is the origin and the source of the requested output, which is a protein. The mRNA is a transcribed copy of the main DNA template codes. The mRNA mediates between the

DNA and the protein. Proteins are translated from the mRNA chains. Matching takes place between nucleotides in a similar way at the cellular level. Simple basic principles of cellular life include all complexity of the living creatures. Computational mediators and design codes might be interpreted in a cellular life-like perspective in architectural design.

Table 7. Qualities and quantities which create space.

	Computable	Non-Computable
Quantities create space	Length, depth, width, degree, weight, lux, etc.	*π, e, 1/7, ∞* ...
Qualities create space	Form, structure, scale, proportion, orientation, illumination/lighting, climatization, time?	Imagination, inhabitants, society, personal experience, uncertainty of future

Considering spatial qualities and quantities together in the scope of computational design practice the concept of computability becomes critical. Computability of quantities is a concept that mathematics has continuously studied. There are computable and non-computable quantities among the factors that make up the architectural space. In addition, the computable qualities which are basic elements of architecture are the values that also create the codes in the background. However, incorporating non-computable attributes into the process of creating architectural space with computational design tools falls outside the scope of algorithmic definitions. In fact, these concepts cannot be calculated precisely in any design process, but projections are created based on personal assumptions and experiences. Although computational design tools greatly reduce the fuzzy areas that occur in the design of architectural space, the areas related to the design process are still quite blurry.

Design Tools as Bridges to Translate Data

In computational design, the abstract data form into an architectural space in different phases. Design problems are imagined, selected, and figured out in the designer's mind. Design factors of interest is transcribed as data and parameters which are executed in algorithms to set up a system to solve, analyze, and edit the outcomes. Next, there is a reflective conversation that takes place between designer and the outcomes of the algorithmic process. It is a process editing and adjusting the outcomes until reaching the desired solution. The following views of participant P1 illustrates this point:

> P1: *No tool can ensure quality of space, but a tool may help translate data as qualitative aspects of design to a graphic or diagrammatic language and then to a spatial language, therefore bridging the gap between analysis and synthesis, or between the problem and its solution.*

There are different kinds of translations that happen in the design process. Intangible ideas, thoughts and concerns are represented in the tangible, graphical and/or textual languages. At the end, all design effort accumulates towards the architectural spatial language. Tools are the bridges of translation between different languages.

4.2.4. Examination of Question 2.4

Question 2.4 investigated the interactions between designers and their tools through the computational design process while their spatial decisions are shaped and how they are both mutually fine-tuned. Participants were asked to focus on the computational design process. Pairing of the design tool and the designer is a very personal and intrinsic process. One end of this paring process is craftsmanship and continuity, while the other end is innovation and experimentation. Different adaptation and customization methods were expected to be explored to achieve the individual spatial expectations of the designers. Question 2.4 is as follows:

2.4 Could you describe spatial concerns that are personally significant to you and could you describe how your personal spatial concerns and anticipations are fed into the computational process? Please illustrate through examples of your own work?

Design and space are two ambiguous phenomena in architecture which are also very subjective. The responses to question 2.4 revealed six categories. The first category relates to the backgrounds of the participants and the way they describe themselves. In the second category, participants' personal experiences are presented. Third category includes statements related to piecemeal consideration of architecture and space. The fourth category is about how the participants transform their design ideas into representations through computational tools. The next category is about integrating bodily relationships with computational design tools. Last category is about the changes in the architectural profession triggered by computational design tools.

4.2.4.1. Designers Considering Architectural Background

The respondents define their way of thinking in reference to issues such as styles and isms. These help designers to describe and position themselves.

Geometry is one of the practical solutions for architectural problems. Geometry has evolved throughout human history. Different geometrical principles cause different comprehensions of the world. Geometry theories keep their consistency according to their own principles. Participant P10 distinguishes their position by pointing out the difference between the simple Euclidean shapes and the topological complexity as follows:

> P10: *we have a long-standing interest in topological complexity and the perception of this - maybe in contrast to simple Euclidian shapes, which dominate the traditional architecture.*

One of the prominent common complaints mentioned by the participants is that overdose and incompetent utilization of computational tools for the sake of complex geometries caused uncontrollability. Participant P15's view illustrates this point as follows:

> P15: *Personally, I don't have 'spatial concerns' related to computational design approaches, since I believe the use of these tools just to generate 'crazy sinuous shapes' is gone, and it is not their proper application.*

Architecture proceeds with all its past references within its professional culture. It also interacts with philosophy, art, and science. However computational design is a novel practice in architecture, it keeps and takes shape in between the dynamic attributes of architectural history. Participant P4's references to the basic doctrines/isms of history could be considered at this point:

> P4: *Often the spatial concerns for me revolve around the effective use of space. Polymorphic shapes are expressive, but also often a little wasteful in the amount of effective usable space they provide. It is a compromise between expressivism and functionalism of form that one has to deal with.*

It could be stated that participants consciously and actively converge or stay away from certain thoughts, concepts, and groups. It is observed that there is a tendency in the views of the participants to match specific forms and shapes with trends and isms. From this point of view, it could be said that there are different cultures which are classified according to the meanings attributed to geometric entities. Forms are messages of architecture whose disciples are grouped around them. Although not as homogeneous as previous styles, computational design practice has also led to differentiation in the history of architecture and created its own sphere of influence.

4.2.4.2. Individual Experiences in the Formation of the Architectural Space

Here, architectural space is considered in terms of issues specific to computational design practice. Designers' experiences are examined to understand the dynamics of the spatial emergence process. Respondents' way of imagining architectural space and the utilization and customization of the tools have similarities and differences.

Participant P17 explains that she is especially focused on contrasts in architecture. Difficulties related to complexity and fabrication are faced with computational assistance. There were representation and production problems. Non-Euclidean forms, complicated modulation and tessellation problems were related to geometrical representation. In addition, high precision requirement and incapacitated hardware problems were exasperated with the conventional fabrication methods. P17 explains as follows:

P17: *The relationship of contrasting elements of architecture that make a space is very significant to, me-solid and void, occupied and unoccupied, etc.*

In consideration with the fabrication process, the challenge of Animated Tessellations was the making of solid and void spaces with using only 5 types of units.

Animated Tessellations 2.0 was a project where I studied pattern making of a façade proposal with modular concrete blocks based on a hexagonal aggregation. There was a total of 5 unit types that had to be aggregated to create the composition of the façade. The computational tools allowed generating a catalog of studies of combined units that provide varied percentage of solid-void combinations.

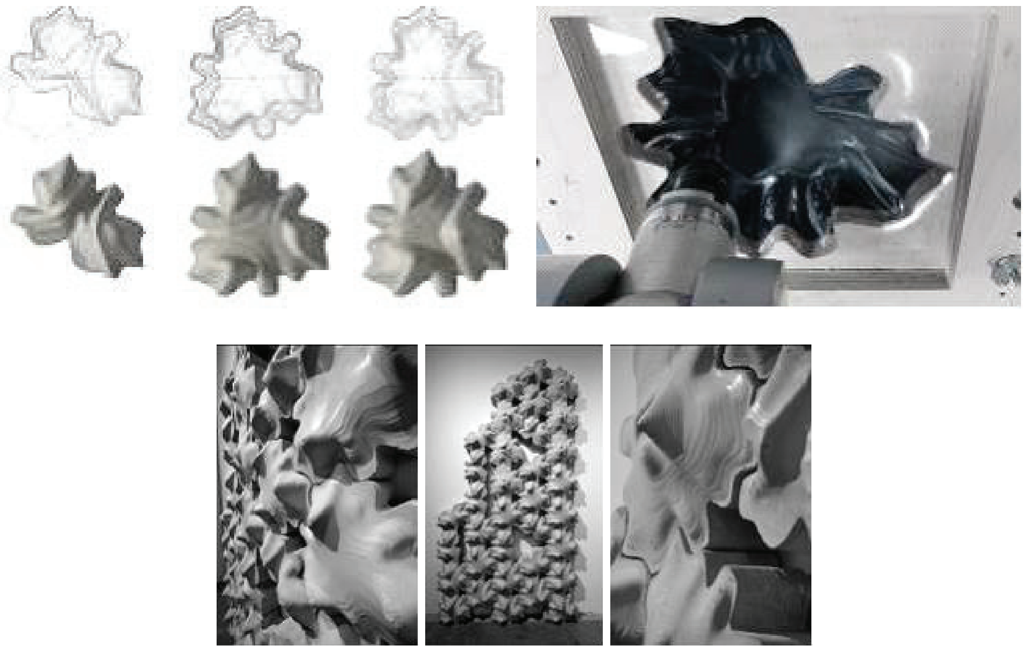

Figure 19. Animated Tessellations 2.0.
(Source: Images submitted by P17.)

Similarly, participant P6 declares that contrasting differences are a primary concern for him. However, he explains a specific interest in the bodily spatial experience. Some architectural elements, according to him, such as roof heights, openings, and patterns are parameterizable, while floor plan decisions are determined by top-down decisions and are non-parameterizable:

P6: *I think it is always important to have a multiplicity in spatial-experience: walking from small to big, low ceiling to big open roof etc. Different overlapping rhythms... I think my floor plan decisions are more top-down while roof heights and openings can be parametrically informed - additional patterns are parametric or coded.*

Importance attributed to spatial experience is highlighted by another participant, P5. According to him, there is a challenge between the fixed and static materiality of architecture and the dynamic and fluid digital representations in his Wynwood Garage project. This project was realized by two separate offices in charge of the building design and façade design. P5's office designed the façade of the building. The spatialization of the two-dimensional façade skin was identified as the main goal of the project. There were design decisions about the context, pattern, perforation, and tectonics of the skin façade, which are specifically interpreted in the project. Main concentration of his contribution is on the visual effects of the façade skin instead of tangible and contextual qualities. It is described as follows:

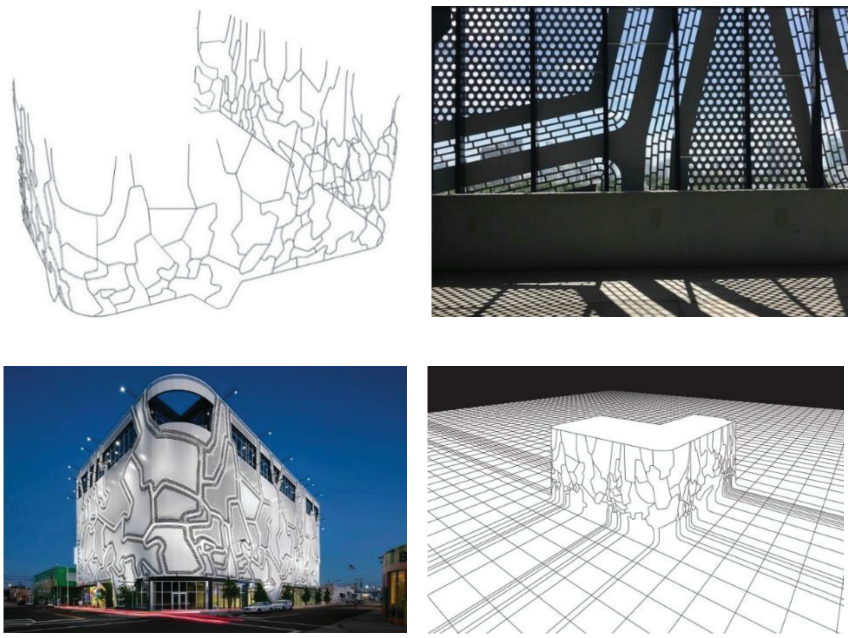

Figure 20. The building referenced by P5.
(Source: Participant’s official website.)

P5: *If digital tools are embedded with a kind of fluidity in that they allow for representational dynamics, might architecture itself once realized similarly capture some essence of untapped behavior, even if materially* *fixed and inert? Can built form still retain its formlessness? For each one of my projects, I remain engaged with questions pertaining to point of view in space as experience. My current façade design for the Wynwood Garage in Miami (finished in 2018) is a relatively flat skin wrapping nearly an entire 8 story block. Though materially fixed, its intended effects when viewed from afar, in between, and up-close play a critical role in its final design. Flat can be spatial.*

Participant P2 also emphasizes the context while explaining her project. Although it is a project that offers solutions in a large exhibition area , its interior experience is mentioned specifically. This project stands out with its dominant fractured and fluid formal language. P2 explains as follows:

P2: *Spatial relation to a context and inner spatial experience are of the same importance to a project, all our project try to be permeable and communicating towards the context as well as to offer a manifold interior spatial experience. I think the perfect project to see this will the Xi'an Horticultural Expo.*

Figure 21. The building referenced by P2.
(Source: Participant's official website.)

Participants customize and differentiate themselves in terms of utilization of computational tools and their architectural characteristics. In addition, computational design tools serve at various range of scales from one brick of a wall to an extensive site plan. In the selected examples, it is seen that computational design practice uses computational tools intensively in representation and production of architectural entities. There is a well-integrated and interconnected design and production continuum from representation, fabrication, to construction. Computational design tools gave designers the confidence that any feature – form, geometry, size, scale, detail, material, etc. – could be considered together, no matter how complicated they are.

The presented project examples above covers the late phases of design but, designers' confidence in computational tools in finalizing the architectural project is of utmost importance for the use of computational tools. Last but not least, architectural offices are more and more specialized to provide services like façade architecture, sustainability consultancy, computational design consultancy, etc. The growing and complex architectural needs have led to such specialization.

4.2.4.3. Fragmented and Piecemeal Consideration of Architecture and Space

One of the outstanding issues regarding the design of the architectural space in computational design practice is its fragmented and piecemeal consideration, which get revealed in two ways. First in the way design issues are separated and isolated into their essential elements and focusing on them individually. Second, once decomposed parts are individually handled, they should be brought back together. Decomposition of architectural entities into its elements is considered almost natural in computational practice.

In the design process, there are two aspects of thinking: top-down and bottom-up. Design is a combination of decisions consisting of top-down ones related to general form and character while the bottom-up is for detailing and customizing the pieces. Architectural design and computational thinking are similar in that sense that the whole is first decomposed and then recomposed. Designing a whole from the pieces is a fundamental principle for both.

Architectural design includes inductive and deductive ways of reasoning by nature. In conventional architectural practice the changing minor or major design decisions are filtered and monitored in the natural speed of thought. However, in computational practice, there is an instant interaction and transformation in design without the close coordination between the whole and the part. Part is a whole and whole is a part, fully integrated. There is not any scale difference while thinking, which requires a different mental set up and comprehension. Calyx project of participant P7 could be an example for this type of framing. Participant reinterprets fractal geometries as an ornament and designs a canopy consisting of individual pieces that are different from each other. The participant P7 explains his views as follows:

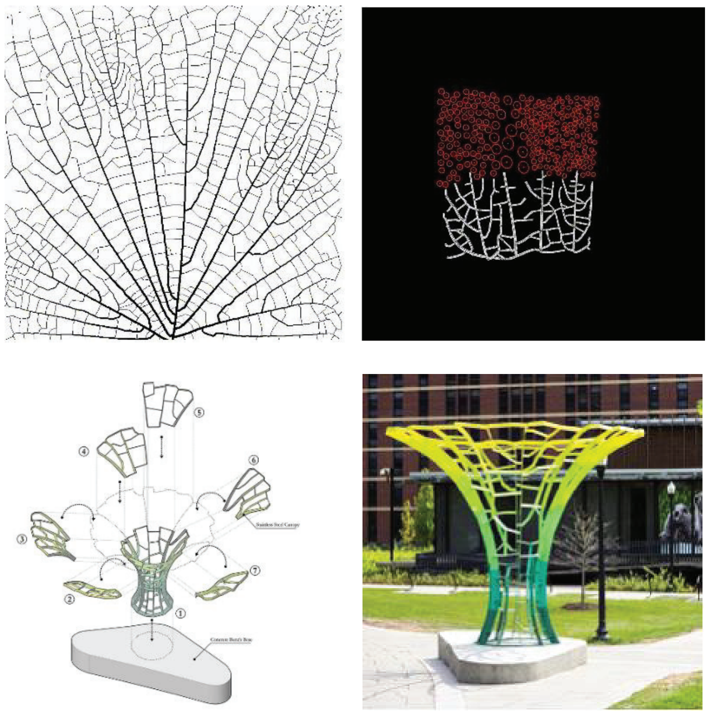

Figure 22. Calyx Project.
(Source: Images submitted by participant.)

P7: *I believe that computational tools offer a new way of fragmenting the spaces that we design. In a sense we can get surrounded with fractal geometries where ornament and space can intertwine and behave in a cohesive manner. Implemented in this way it becomes possible to rethink how we perceive and construct space in a highly decorative fashion. The example below shows how a simple form could be made out of hundreds of smaller pieces that still defines the covered space and provides an aesthetic idea for tessellation.*

Extensive particularity of architectural design components is the issue that participant P12 raises up. Thanks to the increased capacity of computational design tools, all parts of the architectural product can be individually designed. Piecemeal and componential consideration of architecture is one of the prominent benefits of computational practice. The creation of holistic complex geometric forms from small individual parts is adopted enthusiastically by computational design practice. Participant P12 explains as follows:

> P12: *Taking advantage of the speed of calculations in the computer we can now embark on an architecture where all constituting components are unique in shape and dimensions, even in material, structural and aesthetic performance. This is a disruptive radical revolution that will change the architectural profession completely.*

The method of thinking by dividing the issue into its fragments is again a subject where architecture and computational thinking overlap, like the top-down / bottom-up approach. Algorithmic thinking, which is the core of the computational thinking, is based on step-by-step progression to the solution. Designers must describe the design problems, albeit through their fragments, in well-defined, measurable (computable), and procedural (algorithmic) explanations to reach a solution. Even though ambiguity might be interpreted as a vocational and artistic achievement within architecture, it is a very fundamental problem in the algorithmic applications of computational design practice. Configured algorithm becomes a clear and linear handling of design problem. Algorithm is a setup of articulated, instructive, and rule-based logical and mathematical operations. Participant P11 explains this idea as follows:

> P11: *In specific, I always try to generate complex and dynamic forms that adhere to at least a few basic rules of rationalization. My spatial interpretation is not completely free, but it's constrained by the necessity of realization. Last but not least, I always try to maintain a parallel system that describes the design in a few steps that can be either measurable or simply procedural.*

A segmented and articulated representation of the world seems to be natural for algorithmic thinking, which can be considered equivalent to the concept of scale in architecture. Scale is one of the fundamental principles of architecture. Scale coordinates bottom-up and top-down thinking systems. Considering anything in detail or in general view is related to scale. Architects are familiar with considering pieces and the whole together instantaneously in their mind. However, scale is a dynamic concept in computational practice. While it takes a lot of effort and time to process changing decisions to all scales, they can be visualized instantly thanks to the representation advantages offered by computational design tools. Interconnectedness between components of whole create instant feedbacks and update all scales. Cross-checks needed between top-down or bottom-up design decisions become automatized and obsolete in case of revisions. Mutual recognition of scale in algorithmic thinking and architectural design is a key concept both in practice and in thought.

4.2.4.4. Externalization from Idea to Representation

Representation is an outward expression of thought and a form of self-disclosure. There are many ways to represent architectural ideas, such as speaking, writing, sketching, drawing, modeling, painting, and building. Starting from the very first steps, representations of ideas turn into different artifacts. Design ideas lead to decisions and architectural concerns that are translated and integrated back into the representations. The resulting representations are artifacts in which many ideas, decisions, and information are embedded. Computational design practice brought innovations and differences in terms of architectural representation.

Computational thinking is based on algorithms which consist of parameters. The issue of parameterization arises when architectural concerns are considered from the perspective of computational thinking. Some of the architectural concepts are coherent and relevant parameters for translation and integration in terms of their measurability such as physical dimensions of entities, climate, and comfort zone conditions, building structure and infrastructure systems, performance-based preferences. These are measurable quantities that form part of the architecture. Computational design tools offer environments where multiple quantitative systems can be simulated in an integrated manner. When the explanation of how personal qualities are interpreted and they are responded to in computational design, the issues mentioned by some of the participants appear relevant. Considering the points highlighted by one of the participants, P14, similar views regarding spatial values came to the fore as follows:

> P14: *I am personally very concerned about fluctuations and transformations of spatial forms in real time according to various inputs deriving from the surroundings. Thus, the space becomes 4-dimensional since the notion of time becomes very crucial. The idea of adaptive, shape-shifting structures can lead to more energy efficient and practical design solutions.*
>
> *In this framework, I input to my algorithms physics behaviors which enables me to simulate the movement of the design. Furthermore, I base these fluctuations on reactive material properties, therefore material properties parameters are embedded into my definitions. For example, at a reactive wearable project that I worked with a photomechanical material, the inputs for my simulations were the length, the contraction percentage of the material and the time that the actuator was on.*

According to some respondents, spatial concerns change according to projects at hand. Spatial concerns depend on how the design problem is defined which basically follows the designer's own point of view. Spatial concerns are depended on the problem definition and defining the problem is a way of understanding the world.

In addition to the quantitative concerns, qualitative values are also essential in architecture. Participant P1 mentions of them as 'socially driven aspects' of architecture. In his point of view socially driven aspects of architecture might be integrated into and represented within the algorithmic process. He explains that all tools are different in the way they represent the world and externalize the thoughts by which designers customize and interpret their preferred values. He describes as follows:

> P1: *Each project may have different spatial concerns, but they are all related to how a design problem is described, what is considered as being important such as proximities, influences, flows, points of interest, all sorts of activities, other influences and so on. This list generally includes socially driven aspects, and it is important to bring them in the design scene right*

from the start and represent them. With analogue sketch tools, this would happen with the use of thick pencils and ink, or with other techniques the designer experiments with; with digital tools, this happens with the introduction of fields, particle emissions, collisions among soft elements, animated elements, combination of them and so on.

Representation of anything within any method and with any tool in principle could be considered as possible. However, there are some un-representable concepts which are procedurally unconvertible, un-reciprocal, and out of rules. Architectural design action, in which uncertainties are managed, cannot be considered only with computable aspects. There are many un-representable phenomena in architecture. Then computational assistance might be helpful in terms of 'determining indeterminacy' (Fazi, 2018) through design activity. The way that designers express their architectural ideas in the process of managing uncertainties reveals the individual thought patterns of designers.

4.2.4.5. Bodily Senses as the Interaction Tool of Perceiving and Making

One of the participants referred to their own research about perceiving and making sense of architectural space by bodily senses. Body and the senses could be considered as a mediator and an interface of individual spatial experience. Considering body and the senses as connector between tangible and intangible values is worth underlining. This point of view is explained in detail by participant P18 as follows:

P18: *My current focus is on embodied computation, on how computational and physical constructs can be linked more strongly. Here the spatial dimension of the built artifact is also a sensory dimension when for instance the place someone stands within an architecture can act as a form of interface. I have been teaching studios on architectural robotics, where space is a dimension that can be sensed through human presence and manifested itself in student projects such as the "space modulator" by MArch student ….*

Collaboration of computational design tools with the body as a source of individual spatial experience could also add new dimensions to augmented reality studies. It could be concluded that spatial experience is a rich and complicated experience not only supporting sight but also other tactile senses.

4.2.4.6. Changes in the Profession of Architecture

In participants' reactions to the questions of this research, many opinions regarding changes and innovations in the architectural profession triggered by computational design tools were presented. One relates to how offices are more and more specialized in a particular subject, which leads to subcontracting different design tasks to different specialized firms together with an urgent need of more efficient collaborative practices in the profession. Computational design offers many advantages in this respect like collaboration friendly environments, remote participation ability, shared design platforms. This situation is also consistent with the decomposition of design tasks into fragmented units. Design action in a way is fragmented according to specialization and expertise. Participant P3 describes this as follows:

> P3: *Most of my spatial concerns are about the geometry of space. My design process is generally research and reading on geometry. In order to produce something out of this research, I have to learn geometry and mathematics, and experiment new coding skills along with material and fabrication technologies. All my works show different kinds of geometric research with physical outcomes. I don't design alone and prefer to be in design groups taking the role of the geometer-fabricator.*

The ability of specialized teams to work together on the same project is closely related to the issue of design data transfer. The procedural conversion and compliance problems of digital file formats and the which data concerns which discipline and in what detail problems are considered within contemporary BIM software and potential solutions are sought within the scope of them.

In the second section of the interview four questions are analyzed in order to in-depth understanding and different dimensions about tools and spatial thinking are inquired. In question 2.1, it is expected from participants to evaluate how their design philosophy and the computational design tools are combined together. From the birth of the design idea, the steps of the progressing process were investigated with computational design tools in question 2.2. Participants were asked to explain their thoughts on spatial quality by associating them with computational design tools in question 2.3. In the last question, the participants were asked to express their unique spatial values and understandings through their sample projects.

4.3. Introduction to Interview Section 3

In the last section of the interview, opinions on architectural space in the context of computational design approaches is explored. The concept of architectural space has multiple meanings and definitions throughout the history of architecture. Similarly, computational design approaches, like the concept of architectural space, do not have a single and rigid understanding of architectural space. Computational design cultures (Burry, 2011, p.8), which differ from each other, are expected to develop different approaches to the concept of architectural space.

4.3.1. Examination of Question 3.1

In the last question of the interview the compatibility of architectural space and computational tools is opened to discussion to investigate whether there are new and unprecedented, changed and transformed, or preserved and constant concepts in the way space is though about. Among the computational design approaches, differences and similarities in our understanding of architectural space are discussed in detail. Question 3.1 asked the participants:

3.1 Which spatial qualities, concepts, values and norms are best represented and examined through computational tools and which ones are not? What are some emerging aspects of architectural space and some alterations in our understanding of space which might have been triggered by the computational design approach? Please illustrate with examples if possible?

The following four categories were composed by examining the contributions from the participants. In the first category, the transformation, and the new state of the design process after computational design approaches and tools became popular in architectural practice is explained in detail. In the next category, reflections on computability and noncomputability are discussed within participants' selected quotations. Emerging aspects and alterations in the conception of space in computational design approaches are presented in the third category. Last, improvable points of computational design tools are pointed out in the fourth category.

4.3.1.1. Transformation and New State of the Design Process

The new tools triggered the transformation of many aspects of the design process. As a result of these transformations, designing with computational tools created its own architectural universe. Language, way of thinking and networks of the architectural profession are restructured, and the designer still maintains its central role in this transformation. Thinking and designing the architecture over exchangeable data and sharing the intellectual effort that has emerged and accumulated with other professions via digital tools over artificial languages defines many differences in the way of doing architecture. Lean data exchange through different stages of design and between different media and human, while keeping necessary information, is one of the fundamental concepts of computational design process. Participant P12 explains this point broadly as follows:

> P12: *The designer has to be able to describe his concepts in such a way that they can be verified, quantified, and transferred via lean data exchange. If the designer is not able to do this him/herself, he/she will lose control and be left to the limited spatial skills of others in the building chain. Our aim is to explode the building chain, to unleash the building chain and operate as a distributed network of expert designers and producers, communicating exclusively via databases. That is the new spatial language designers will have to make their own. The new language is HTML, Java, C++, Grasshopper, XML, Python, or any other programming vehicle as to exchange data in their most essential form.*

At this point a kind of universal definition of architecture emerges as the management of the ongoing process until the mental effort turns into a tangible product.

Exploring Covert and Imperceptible Search Spaces

Design exploration is another topic that changes format with digital design tools. Designers manage their design exploration by manipulating design tools. It is a process to keep the balance between controlling some aspects of design while exploring through hidden, unanticipated, unexpected, maybe accidental but surprising territories. Designer makes some decisions before starting to design, but the rest of the ungiven decisions are a gray area which is finalized by trial and error, and testing ideas. Participant P15 expresses this as follows:

> P15: *...I think the real power of computation in architecture is the possibility of showing what is not directly visible or predictable, or also the chance of controlling multiple aspects at the same time. It could be the production of different layout options for a building depending on the optimization, or urban data about customer users and most used routes, ..in my opinion sky is the limit, really.*

Interaction of Architectural Representation

One of the mentioned points about the transformation of the conventional design process is related to architectural representation. Representation of architectural thoughts within two-dimensional or three-dimensional representations result in different kinds of comprehension. Computational tools help to design directly a three-dimensional object without any need of mediating 2D drawings. Explanations of the participant P9 indicates that the design activity starts directly within three-dimensional model and two-dimensional drawings are just exported as construction drawings for machinery production procedures as follows:

> P9: *Most computational tools start with 3D surfaces or volumes, they don't start with plans, sections, elevations. Rather, these plans and sections can then be displayed from the 3D information. For an increasing number of architectural artefacts, plans and sections are as superfluous as*

> *construction drawings. The 3D information can be directly translated into instructions for CNC machines and additive manufacturing. The lack of plans and sections could lead to a more sculptural approach in architecture, one that favors more complex curvatures. It also frees the design – for better or worse – from a sense of scale. Computational tools allow jumps in scale, or entirely continuous scales, that would be more difficult to achieve using traditional means.*

The opinion that architecture and/or space is thought directly in three dimensions, bypassing the 2D drawings, was considered positively, but the loss of the sense of scale was also mentioned as a point to be considered. Thinking or representing architectural space directly as an instantaneous and interactive 3D representation is a new phenomenon of architectural practice allowed by the increasing speed of digital design tools in recent years. The fundamental two-dimensional drawings of architectural discipline - plan, section - become secondary outputs of the architectural model which is able to be handled in three dimensions easily. Consequently, almost all three-dimensional modeling software packages offer basic two-dimensional drawing windows to their users as the starting interface format to accommodate the switch between 3D and 2D.

Integrated and Interdependent Architectural Design Process

Another transformation in the design process is the increasing integration and interconnectedness. From early design to production and operation, the design process is knitted with a network of feedbacks. This situation is explained by participant P18 as follows:

> P18: *I do not believe in the separation between design phase and the phase of usage, but rather view design as a form of embodied computation, where the construction of a physical artifact through computational means is only the midpoint of the design process, which continues in the artifact being inserted into the world shaping it and computing through its matter and in connection with the humans and environment that surrounds* it.

Computational design process offers instant multidirectional feedback between performance, cost, manufacture, material, sustainability analysis to designers. Interactive process includes trial-error actions and customization of the design tool. Explanation of participant P17's design process could be evaluated in this perspective:

> P17: *For the project en Pointe, I fabricated a mandrel that attached to the Staubli robot to extrude plastic over it. The toolpaths for the robot were generated using grasshopper and python based on the concept of structural lace. After months of trial and error, I was able to get results in physical that represented the digital toolpaths. The robot sure did follow all the toolpaths, but the under-accounted problem was the material properties of the plastic; such as, the temperature and the time it takes to cool down, the distance and time before hardening before it can be stretched,.., that didn't allow achieving one-to-one results to that of the digital model. The months of experimentation were a result of testing tool paths and all the physical factors above to then fine-tune speed, temperature, and amount of plastic extruded. Eventually, I had to write a python code that took in any tool path and broke it down into custom extrusion speed and temperature for every translational move in between target points.*

Although there is always a flow of feedback in the architectural design process, with the transformation experienced with computational design tools, the channels in which this flow takes place have become more diverse and its speed has increased.

4.3.1.2. Reflections on Computability and Noncomputability

Computability is one of the prominent concepts and is discussed by the participants while considering the last question which is focused on the spatial qualities and understanding of space. Although the way in which the participants deal with this issue sometimes shows similarities, there are also those who make comments that draw attention to different points.

All six participants who touched upon the computability issue agree that computational design tools are competent and convenient if architectural problems at hand are measurable, quantifiable, and numerical. For example, figuring out the geometrical relations about forms, providing comfort zone values, evaluations regarding the material and its use, maximum minimum optimization of different issues, construction, operation, and consumption analysis, considering fabrication solutions. All quantitative aspects are computationally representable for them. Participants P17 and P15 declare their views as follows:

P17: *Any problem based on data is fit for a computational solution. ...*

P15: *In general, I would say that there's no quantitative aspect that is not representable through computation if properly modelled and abstracted. ...*

However, there are nuances worth mentioning after this consensus. One of the participants attributed this to the condition of being a competent designer and the selection of proper tool for the action as follows:

P1: *I would think that everything could be represented and examined somehow through advanced computational tools, but this is up to the designer's intuition, skills, talent, and experience, performed along with the tools one chooses to work with. I would use a rough pencil to perform fast sketches, but not a detailed drawing, for which I would use one with thin sharp lead. In a similar manner, there is not a single tool that can do everything in the best possible way and designers should develop agility to become familiar easily with any tool, method, or process.*

Afterwards, it is stated that not all architectural values and concepts can be covered within the framework of computational design tools because of representability problem. Lack of representability of intuitive qualities of space like aesthetics relates to basically the problem of integrating non-computable values into computation process. Participant P11 opened this issue up for discussion as follows:

P11: *Everything that can be quantified, can also be assessed by a computational analysis. In that sense, everything that has to do with BIM, or sustainability-oriented values is best represented by computational design. That way, possible discrepancies in the way buildings perform or cost are addressed in advance, hence tackled, and resolved with alterations of the design. At the same time, ambiguous and intangible qualities of space* *like aesthetics, ratios etc. which are intuitive cannot be represented at all. Moreover, most of the time contradict the results of the analysis of the quantifiable elements. New emerging elements of space mainly have to do with the new technological input into architecture, (whatever that is, residential, work-space, cultural etc.). More and more automation is integrated within contemporary designs, an that's something tightly connected with the physical computing aspects of design computation or the ability of mapping and collecting data for analysis and pattern recognition in spatial behavior, in the same online data collection operates.*

Another participant, who relates representability and computability, approaches the subject through the experiential factors. From this point of view, experience is not a quantifiable value and for this reason it is computationally unrepresentable. Experience, representability, and quantifiability concepts are interrelated by the participant P4 as follows:

P4: *Quantifiable spatial qualities are definitely the ones that are the best represented and explored through computational tools. Amount of light/daylight, shadows, air flows, heating, but also paneling, structural optimization, etc. are very meaningful and usable design information. The experiential factors are not easily quantifiable and thus not easily representable.*

Computational thinking is independent of tools. Computers or digital design tools are not prerequisite for computational design. Comprehension of the computational concepts, which are matched with composition principles, defines a different way of designing. Participant P3 explains how computational thinking and compositional principles overlap in architectural education as follows:

P3: *In our "computation-based basic design" and "architectural geometry" classes we see that the concepts of "emergence", "iteration", "logic", "system", "relationship", "balance", "formation", "deformation", and "material computation" are very suitable for the first-year architectural education. However, our tools are conventional drawing and model making techniques.*

In this educational experience it is revealed that there are different computational approaches possible which can be classified along computerized vs. non-computerized processes. In manual or non-computerized computation, the focus is on thinking and principles using customary non-digital design tools. However, in digital or computerized computation, which is supported by computers, the ways and principles of computing are automated.

It is seen that computability is discussed in the discipline of architecture especially in architectural space design by describing two different worlds and considering the transitivity, transformability, or pairing up between qualitative and quantitative values. Considering this duality, the architectural space emerges from the unity of quantitative and qualitative values. Continuing to advance computational design tools on computable values will lead to a sterile and reductive architectural culture. Here, users should seek values for architectural spatial qualities, considering the risks of oversimplification and reductionism of computable architectural quantities. The reconciliation of architectural qualities and quantities requires a kind of mastery, which some participants also mentioned in the process of computational design.

4.3.1.3. Emerging Issues and Changes in the Understanding of Space

By introducing new possibilities, emerging computational design tools might emphasize new or particular dimensions of architectural space. Unprecedented forms and geometries enabled spatial experiences that have not been experienced before. Participant P14 explains this by basing it on two iconic figures of contemporary architecture as follows:

> P14: … *For example, Zaha Hadid and Frank Gehry deconstructivism give a spatial quality that we were not able to experience before the presence of computational tools. Architects are now able to realize more imaginary designs…*

Traditional understanding of space consists of static perspectives and fragments of moments. Thinking of space as stable and constant indicates the over-emphasis on the sense of vision in traditional understanding. Visual senses dominated the perception of space in traditional spatial understanding of architecture. Some have suggested that computational tools brought inclusion and integration of missing senses of spatial experience like auditory and tactile qualities. The apparent transformation regarding this was explained by P10 as follows:

> P10: *the traditional understanding of space is somewhat limited to visual aspects - and to static moments, that are defined by perspectives. computational tools allow, through simulation, to supply feedback of spaces on levels, which make our spatial experience, such as sound and material qualities. The implication of light and the interaction of light sources from artificial and natural sources is as well only possible with computational tools. It is for sure possible to imagine and create related spatial concepts without the help of computation (as 3000+ years of architecture show), but computational tools allow us to master and receive further inspiration. Computational tools grant furthermore access to an understanding of temporal aspects of time. The rise of the sun, the movement through space and the creation of events, which alter and influence the spatial understanding of the inhabitants and visitors of users are possible with computation. Any feedback (the creation of input to the designer) will influence and inspire design decisions and create new pictures in the head, which inspire and drive the design. As the computer is able to create this feedback instantly it has for sure an enormous influence.*

> P14: *In my opinion computation allowed the creation and construction of complex curved geometries. Biological forms can emerge from algorithms and create fluid, continuous spatial qualities.*

When the contributions of the designers like participant P14, who prefer to use computational design tools, are examined, concepts such as continuity, fluidity, transiency are frequently mentioned in relation to spatial quality.

Some ways of handling computational design tools have also been revived by associating them with concepts from architectural history. Ornament, one of the cursed concepts of architecture during the modern times, is emphasized as a phenomenon that can be revisited. The precise transition between form and production, easily offered by computational design tools, reintroduces ornament as a new spatial quality. Participants P6 and P7 present examples of this approach as follows:

> P6: *Maybe the use of computer-aided-manufacturing is making things more ornamental and baroque again - which makes architects, maybe, think about light and shadow in a smaller scale again. Additionally, digital tools are very good for intensive site-analysis (CFD) - although I see people using them more as a post-design advertising tool. I think the use of digital tools reduces the quality of space (for now), but it allows to think more systematic and formal - and maybe towards an information-based bottom-up architecture.*

> P7: *I think computational tools offer an advanced way to look at geometry, form and pattern and could help us define various spatial qualities through ornamentation and decoration. This means that we don't have to look for spatial qualities in the form of space, but rather in the form of structural elements that define space. The example below illustrates this approach where computational tools could be used to define ornamental structural members that can both define and aestheticize architecture where various spatial qualities could emerge.*

Figure 23. Ornament as a structure.
(Source: Images submitted by P17.)

Ornament is not interpreted in these two examples in a classical way but in terms of scale differentiations and integrated with the structural system which are not possible without the capacities of digital tools. From this point of view, it can be inferred that computational design tools can also reinvigorate the timeless concepts of architectural history again.

Last but not least, one of the participants mentioned the topic of augmented reality. Augmented reality has two associated concepts, virtual reality, and mixed reality, all of which deal with different levels and degrees of manipulation of reality. These three related concepts differ from each other according to the spatial qualities they cover. In addition to computational design tools, developing additional technologies also open new areas of experience by removing the bodily existence, which is the most basic element of architectural spatial experience. Spatial values and concepts related to experience in these newly opened dimensions are future discussion topics.

4.3.1.4. Ways to Improve Computational Design Tools

Two participants made an intersecting critique on a common issue, which basically could be defined as integrating visual and analytical aspects of design simultaneously. The main expectation here is the simultaneous foreseeing of design and production outcomes through feedback loops between form and material. The expectation of instant feedback on material performances in parallel and integrated with computer visualization is explained by P17 as follows:

> P17: *Visualization of material manifestation isn't greatly represented and conceived through computational design platforms; however, material qualities-properties can be examined through numeric values to help build a computational platform that serve solving a spatial problem. These values become an abstract idea throughout the computational design process, for which I believe a simultaneous exchange between physical and digital design is necessary.*

Final effect is interpreted as a photo-realistic expectation of designer from computational design tools in addition to geometrical success. Insufficiency of computational tools in rendering visual reality and poor representation of architectural tectonics with computational tools are issues which are pointed out by participant P16 as follows:

P16: *Details, joints and material effect cannot be understood very well only with computational tools, you can cover the geometrical setting out but not the final effect. So, I'm assuming that the computational design tool is the bridge between the idea and the fabrication of it.*

Checking form-geometry, strength-physics, and production-fabrication in different digital platforms is integrated but separated activities. These activities are within the expertise domain of different professions as architecture, engineering, and construction. Expectation of the visual rendering reality and the precise numerical control together in the same platform forces another level of reconsideration of architectural tectonics within computational and digital capacities. Architecture looks for a new way of seamless transition from fiction to reality with the help of computational design tools. It seems that after the possibilities offered by the computational design tools, the expectation for a supreme tool, with which Vitruvius' three principles - firmitas-structure, utilitas-function, venustas-aesthetics - can be controlled instantly, has emerged.

The responses to the last section of the interview are discussed under four categories. When evaluated together with other sections, the results obtained during this research on architectural space within the scope of computational design approaches could help designers develop their own positions.

CHAPTER 5

DISCUSSION

This chapter will discuss the emergent categories and subcategories in the analysis in the context of the conception of architectural space in computational design in architecture. This discussion is structured under three headings: tools, space, and computational design. Tools are examined in two dimensions 'New Tools, New Worlds' and 'Permutations of Design Tools'. The findings on space of the thesis are discussed under the headings, 'Effects of Tools on Spatial Quality' and 'Sense of Space in Computational Practice'. Insights and implications of computational design are discussed under the following headings: 'Exchange and Translation of Design Data' and 'Acquiring Computational Way of Thinking'.

5.1. Tools

5.1.1. New Tools, New Worlds

Tools have been intrinsic partners of humankind throughout the evolutionary process, which, according to some, is the main determinant of differentiations between species (see Tomasello, 1999). Over time, tools have undergone refinements and innovative customizations for specific purposes. The world, shaped by humanity, accommodates the utilization of various specific tools. The field of architecture, too, is intricately intertwined with the evolution of tools. Nelson Goodman (Goodman, 1978) explains that different worlds become possible under different circumstances and within the realm of architectural design, the act of toolmaking represents different approaches to world-making (p.7). The use of different tools during the design process signifies the creation of distinct architectural worlds. The introduction of new tool sets opens novel design search spaces, and in the case of computational tools, it has provided new avenues for exploration, expanding the formal, intellectual, and constructional research areas in design.

In the realm of architectural design, where the evolution of tools shapes different worlds, the role of the designer as a toolmaker takes center stage, emphasizing the significance of control, mastery, and the delicate balance between the designer and their tools. According to Frascari, “tools do not translate ideas but produce ideas with their interaction with the surface” (2011, p.32) and in this interaction tacit knowledge needed which “comprises a range of conceptual and sensory information and images that can be brought together to make sense of something.” (p.18). Furthermore, the role of the designer as a toolmaker is also preserved in the computational design process. Customization and innovation through toolmaking represent advanced levels of sophistication, wherein designers become experts in utilizing their tools. Maintaining control and competence over these tools enable a delicate balance between being captivated by them and utilizing them effectively. As Picon states that using a tool brings hesitation between hubris and restrain together and architects “need to embrace the contradictory longing for power and for restraint, for standardization and for invention.” (2011, p.35). Mastery entails a harmonious cooperation and an escalating coordination established with the tools.

5.1.2. Permutations of Design Tools

Considering the participants’ responses, new computational tools brought into the design process and are used in integration with other design tools. Customary and computational tools are used together complementarily through the design process. Design tools are selected and ordered for a defined purpose through the design process. In this regard, computational representations and tools are one of the sets of tools in the architectural toolbox. Design tools are articulated to deliver the desired action or service more easily, quickly, or effectively.

Similarly, computational tools are associated with other computational tools. Participants use different software packages to accomplish specific tasks. The set of design tools articulated in digital continues to be articulated in the fabrication phase as well. Integrability of design tool is the parameter, which increases its effectiveness and potentials. Perhaps the reason that sketching and drawing is the most widely used design tool is because of their simplicity with which they can be easily integrated into any process.

Designers switch between design tools at certain times. Appropriate tools are consciously and intentionally. selected, combined, and matched to accomplish the expected design tasks. There is not any specific chronological order detected in the responses of participants in this study, however the stages known as early design, representation, and production are clearly distinguished in the computational design process as well. Regardless of computational or conventional design tools, tool switching, combining, and transition between tools have been seen as an important aspect of designing in itself in the design process.

5.2. Space

5.2.1. Effects of Tools on Spatial Quality

Spatial quality in architecture after the widespread use of computational design tools is the core focus of this dissertation. There are many factors that affect spatial quality including designers' environment, culture, education, and the design tools that involve generation of representations. The effects of computational design tools on architectural space could be considered within different perspectives. In this study, despite the difficulty in defining the concept of architectural space, its recent ramifications are inquired about in the context of computational design.

Computational tools are superior and advantageous in countable and measurable quantities as mentioned many times by participants, however intuitive and intangible qualities cannot be included in the algorithmic representations of computational design. Spatial qualities are associated with designers and tools. In an ideal design process, designers imagine the expected spatial quality considering the limitations of their design tools.

When it comes to achieving incomputable qualities with computable quantities through computational design tools, the concept of intermediation comes to the fore. Just as it is said that 'the tools are means to an end', so quantities are means to qualities. Measurable and countable tangible data mediate architectural spatial qualities within computational or conventional design practice. Different design tools impose different methods on users, but any tool by itself do not ensure the quality of space. Appropriate, competent, and creative use of tools might be the necessary and essential requirement of spatial quality.

5.2.2. Sense of Space in Computational Practice

Architectural language, thinking sets, representation methods, and the design tools have undergone important changes within the computational paradigm. According to the presented views of the participants some inferences could be derived regarding the sense of space shaped in this new paradigm. One of the findings is that perceiving space with computational design tools does not match the bodily sensations and is inconsistent in terms of internalization or calibration of the sense of scale. Since it cannot support physical calibration through bodily experiences, it is defined that working with digital models on the screen causes loss of scale. Despite all the precise measuring capabilities, digital modeling software cannot offer a feeling of phenomenological ratio and proportion yet. Distorted scale has ramifications in every dimension from the detail scale to the urban scale. It demands the user to adapt to a new way of mental perception. This distress is also compounded by the inconsistency between 2D screen viewing and three-dimensional spatial experience. Interactive and dynamic zoom environments offered by software packages are incompatible with real life visual experience. Designers' body and mind need a new calibration and adaptation in terms of ordering sensorial information.

The sense of space is also reconsidered due to the bottom-up and top-down way of thinking that get merged in the computational practice. This point combines the scale-free and piecemeal understanding of architectural space. Elements and details, which make up the spatial unity, are evaluated separately through the bottom-up thinking. Instantaneously, the digital model refreshes itself as a whole when any revisions or changes are updated in the top-down decisions. Design feedback is checked multidimensionally, considering the inserted parameters. In the deductive and inductive processes provided by computational design tools, it is possible to double check the spatial qualities at different scales. Although some of the participants stated its shortcomings in terms of instant realistic visualization, computational design tools maintain absolute and full control while merging the early design phase and the fabrication concerns together.

Considering the space as the end product of architectural process, it could be said that the computational design process is more indirect when the mediators between the designer and the space to be designed are considered. Requirements to take advantage of computational design tools increases the distance between the designer and the designed space. Designers must define abstract processes to translate design problems into artificial languages that computational design tools can process. During the computational design process, secondary or tertiary agents and languages become imperative. In addition, the semantic equivalents of the design action in the language have also expanded and changed. Previously, the design action was associated with actions such as sketching, drawing, collage, and model making however, now, coding, scripting, and designing algorithms are added to them. When the act of drawing is considered as an example, the number of tools such as pencil and paper, in which this action is conventionally performed, have also increased, and changed. Tools and actions transform while the goals remain constant. The transformation of the act of pruning the lawn might be an example to explain this better. This process could be completed via plucking by hand, cutting with scissors, and driving a lawnmower. Ultimately, the grass is trimmed, but the practitioner and the process are different in all of them. While one way of pruning the lawn is more bodily integrated and sensorial, the other way is more effective and precise. As a result, indirectness brings with it operational, instrumental, mental, and sensorial transformations.

Considering the preliminary sketches of Frank Gehry, a conventional method which helps to represent the core concepts of the entire project determines its originality. It is intended to speculate on whether computational design tools offer as much depth and inclusiveness as sketches with the generated algorithms. Design tools used to convey ideas while imagining architectural space need different evaluation criteria because they define different worlds.

5.3. Computational Design

5.3.1. Exchange and Translation of Design Data

Architectural artifact emerges out of and in result of accumulation of given decisions. The data produced by the decisions made from the moment the design start accumulates and forms the artifact. Designer processes and configures his/her own solution from the emergent data. Information emerges when the accumulated data turns into a meaningful formation. Correlatively, Lorenzo-Eiroa (Lorenzo-Eiroa, 2013), in his book "Architecture in Formation," explores the revolution that has occurred in the field of architecture, marked by a notable shift in focus towards recognizing and engaging with the underlying structures of architecture. Information technologies, especially the ability to directly work with computer codes, have played a crucial role in driving this transformation and opening up new possibilities for architectural representation. Lorenzo-Eiroa critically examines earlier approaches that heavily relied on visual judgment alone, emphasizing the need for a deeper understanding of relational structures. Design data is transferred between three types of pairs: human and human (colleagues, professionals, students, and instructors, etc.), non-human and human (pencil, rulers, computers, machines, software, etc. and artists, designers, operators, etc.), and non-human and non-human (computers, machines, software, etc.).

It is known that there are strict protocols between different software packages to export and import data in file transfer. The main purpose of these protocols is not to lose data. Problems with digital format conversion remain significant for different reasons. Conversion of similar files produced with two different software packages or opening a file produced with one software package with a different one are still problematic. Although very important, this topic will not be discussed further because it needs computer science expertise.

Sharing design data between two people through representations is much different than the relationship between two non-humans. Here, exchange and translation of data via tool reminds the conduit analogy that linguists often use as a metaphor. In fact, a double translation occurs here. It is seen that the transfer of design data between human and human cannot be considered independently of tools, since the translation that occurs with the human-tool sequence is passed across as a tool-human in the communication process. Moreover, when it is considered from Ortony's (1993) point of view, it is debatable that whether the any data can be moved and transferred literally without changing.

The focus of the dissertation is examining the transfer and formation of spatial data in the minds of designers through and with design tools. Designers first filter and export the images formed in their minds. Early representations of the mental images are the first phase of the translation of the design data. It is seen in the participants' contributions that conventional sketching with pencil and paper is still the most preferred way of expressing early design ideas. Design is an interactive and non-linear process, so there are different feedbacks and refinements along this process. Many decisions and information might be excluded, or new data could be joined along the design process.

While the design process is ongoing, the data is transferred from one stage to another by specific design tools. Designers interact with design tools via representations of them. Designers produce representations with tools. Representations are the interfaces or mediators of the communication. Whether artificial programming languages, technical application drawings, 3D mockups, digital models, or abstract sketches, the continuity of the design data to be transferred is essential. It is one of the other architect's responsibilities to maintain and protect the change and translation process of the characteristic traces that make up the design.

Furthermore, designers are opportunistic, and they benefit from serendipity enhanced through any act of translation. Serendipitous moments often arise during the translation process, where unexpected connections, insights, or discoveries can occur. These instances of serendipity contribute to the creative and innovative nature of the design process, allowing designers to explore new possibilities and push the boundaries of their initial concepts. By embracing and leveraging the opportunities that arise through translation, designers can uncover novel ideas, alternative approaches, and unforeseen solutions, enriching their design outcomes and fostering a dynamic and evolving design practice.

5.3.2. Acquiring Computational Way of Thinking

One of the problems mentioned by the participants is that incompetent users are not aware of the design data processing within computational design tools. They use these tools unconsciously or without being aware of what happens computationally in the background. They are not interested in algorithmic logic and computational thinking. Some of the computational design tools aid and abet this situation by offering ready-made patterns which are not transparent and obscure the computational logic behind. Fixed library sets, common open source ready-to-use algorithms, configurations, which all seem to be a blessing, could also turn into a trap, as some participants stated. The risks of affinity due to their unconscious use might homogenize and assimilate incompetent users. However, there are some people who use digital tools or conventional tools while being very aware of computational meaning of the design problem and considers the logic of the problem first. As some instructor participants reported learning computation without computers is a useful method of gaining computational thinking faculties for beginning students.

Previously, one of the conditions of using computational design tools was to learn the syntactic languages of programming. Most of the programming languages also required familiarity with mathematical formulations. The effort spent in learning these languages actually helped to grasp the logic of computational thinking. Later, for the convenience of the users, interfaces were developed in which the necessary options are already presented and ready to use, and not needed to be structured from the bottom-up by coding. While criticisms were expressed concerning how the facilitation offered by user interfaces overshadows computational thinking, new developments related to artificial intelligence were announced. Adobe Photoshop promoted that users could generate variations with the help of artificial intelligence. It is declared that in the new possibilities offered, users can directly explain their demands within a dialog box by standard colloquial language without translating them to express their demands in programming languages. Moreover, artificial intelligence, which is trained with images of a selected artist with the ‘multi-concept customization of text-to-image diffusion’ technology (Kumari et al., 2023), can then reproduce another image given to it with the artist style it has learned. ChatGPT (Generative Pre-trained Transformer) is another artificial intelligence development that makes learning programming languages not a necessity but makes it more difficult to see the computational thinking system in the background. It is defined as a trained model that interacts and communicates with the user in writing and accepts visual inputs also.

OpenAI, the company that produced ChatGPT, continues its work focused on artificial intelligence in a wide range. DALL·E 2 is another system that generates original, realistic, and artistic images and variations from a given text description. It could complete the original artistic painting named as outpainting or add a new object into an existing image with its own style, its inpainting. These applications will be added to the architectural toolbox in the future. It seems that the importance of language in new design processes will increase gradually. When humans were reluctant to learn artificial languages, artificial intelligence learned the natural language of humans. When the effort to become an expert in the use of a specific design tool is coupled with the ease of natural language processing of AI, it might no longer be necessary to learn a programming language, nor acquiring computational thinking. This hybrid method, which is realized with natural language processing without comprehending computational thinking, might be categorized as non-classical computation in Stiny's terms. In short, considering these developments, it is doubtful how long the insistence on acquiring the true logic of computational thinking could be sustained.

CHAPTER 6

CONCLUSION

The primary objective of this dissertation was to gain a comprehensive understanding of the notion of architectural space within the realm of computational design practice. To start the inquiry, a review of the use of the concept of architectural space and its interaction with computational design thinking is first presented. While the practices of computational design have become increasingly prevalent in the production of architecture, its repercussions in the way we think about and imagine space need to be fully examined. The result of the analysis of the data obtained from the participants were presented in Table 3.

This research is shaped around two main questions: (1) How is architectural space imagined in the computational design practice? And (2) How architectural space is considered as a multi-layered, loosely defined, and ineffable concept in computational design approaches in which procedures are precise, descriptive, and prescriptive? Investigating these questions, the study inquired into the changes experienced in the way we conceptualize space within computational design practices, and discussed what is preserved, adapted, or neglected from our modern understanding of space.

Investigating the first question, although it may not come to mind first, architectural space is imagined within the limits set by the tools. Even though the infinity of the imagination is generally accepted, practices based on making / construction such as architecture eventually have to face the capacities of the tools. But the power and limitlessness of imagination in architecture could be thought of in terms of how to use and interpret the design tools. The design process is a struggle and a compromise in which many parameters articulate and interact. The way tools accompany the designer's actions, and the way designers use their tools is a unique and creative interaction. Similarly, the design tools used in computational design practice and the ways in which they are used have led to unique and creative interactions. Tools have internal dynamics, but so do the ways in which their users evaluate them. Design process and spatial imagination takes shapes between these mutual actions. Each set of design tools and their mode of use establishes its own spatial relations.

Considering the second question of the research, architectural space is as abstract and ineffable in computational design tools as it is in conventional design tools. While computational design tools enhance designers' capacities in many directions, they can only operate when the inputs are numeric values or can be converted into computable data. Designers reflect their spatial individuality in their use of computational design tools as much as they do in their use of any other design tool. However, the importance of the individual spatial thinking that directs how the tools are used should be emphasized here. Moreover, the intellectual environment from which the tools derive is reflected in architecture with an interdisciplinary transitivity in the natural flow of life. New tools integrate and match with new ideas. Space is affected not only by the tools but also by the intellectual processes that give rise to them. The way of thinking and dealing with space interacts with the intellectual and instrumental environment in which it exists. Taken together with the dynamism and excitement generated by the developments in artificial intelligence, when it comes to computational design tools and computational thinking, their reflection in architectural space is still far from over.

The dissertation adopted a qualitative thematic analysis method based on the data obtained from the online written interviews with participants who are experts in their fields. Following a particular research agenda, the study aims at understanding space-related conceptualizations of designers through interviewing expert practitioners in the computational design. It is argued that computational tools have an impact on designers' way of thinking while imagining space, alongside the parameters that they consider. The assumption here is that space as a multi-layered and multi-meaningful concept has both computable and non-computable aspects and that it is worth exploring how computational practice treats both aspects. While computation may provide an appropriate medium for thinking about and representing the quantifiable aspects of space, the ineffable and ambiguous aspects of space highlight the limits of what is computable. The participants' contributions to the questions of this inquiry indicate an indirect, if any, interest in the concept of space and how it could be rethought with the new computational technologies.

One of the key findings of the research pertains to the reasons for using computational tools in architectural design. The analysis indicates two categories related to experts' preferences of using computational tools. The first, referred to as imagination related reasons, indicate a preference for computational tools due to the creative possibilities, freedom, and novelty they offer. These designers valued the ability of computational tools to generate new, unique, and unconventional designs that were previously unimaginable. The use of computational tools allowed them to explore uncharted territories and push the boundaries of architectural expression. The second group, known as pragmatic reasons category, cited practical considerations as the primary drivers for utilizing computational tools. These designers emphasized the advantages of computational tools in the production, fabrication, and construction processes of architecture. They highlighted benefits such as improved material and time efficiency, and reduced labor demands. Computational tools enabled them to optimize designs for efficient construction, therefore, achieve cost-effective outcomes. This suggests that there are diverse motivations behind the adoption of computational tools in architectural design.

Findings forming another superordinate category in this research relate to the integration and application of different design tools. In this respect, there were four main categories of responses. The first category, 'Modes and Combinations of Design Tool Use' explores the various ways designers utilize different tools in combination. This includes the alternation between digital and analog tools, considering both computational and non-computational modes. Examples include the combination of analog tools such as sketching, drawing, and physical models, as well as the integration of digital tools like computers and digital models. Design tools are used in four different combinations as analog-computational, analog-noncomputational, digital-computational, and digital-noncomputational. The second category, 'Chronology and Order of Use of Design Tools' examines the specific tools employed during different phases of the design process for particular purposes or services. It investigates the sequence and timing of tool usage throughout the design workflow. The third category, 'Capacities of Design Tools' focuses on the limitations and constraints associated with various design tools, whether digital, analog, or computational. It defines instances where designers may need to customize or combine tools to overcome these limitations and achieve desired outcomes. The fourth category, ‘Design of the Design Tool’ acknowledges the active role of the design tool in the design process. It recognizes that designers re-shape the tools through customization and hybridization, while the tools, in turn, shape the designers' approach, either with fixed and predetermined characteristics or with more flexible adaptability. Briefly, these findings highlight the diverse ways in which designers utilize and combine different design tools. These categories reveal the modes, sequences, limitations, and design considerations surrounding the integration of tools within the design process. Understanding these aspects contributes to a deeper understanding of the complexities and potentials of employing multiple design tools in architectural practice.

Under the superordinate category of representation with design tools, three distinct categories emerge from the findings. The first category, ‘Mind Mapping Medium’ delves into the process of mapping intangible thoughts, abstract data, or initially ill-structured design ideas to attain visual representations of architectural space. This category encompasses both digital and manual approaches, as well as computational and non-computational tools. It refers to the use of various tools to transform conceptual ideas into tangible visualizations. The second category, ‘Externalization of Emergent Design Ideas’ focuses on capturing and following up on design ideas that emerge during the design process while preserving the designer's initial intuitions and preferences. This category emphasizes the importance of maintaining and refining selected ideas that surface throughout the design process, ensuring their continuous development and integration into the design. The third category, ‘Transformation and Transfer of Design Data’ considers design as a collection of transferable data. It explores different modes of exchanging design data, such as between individuals (colleagues, teams, seniors/juniors, instructors/students), between machines and humans, or between machines and software/objects. This category highlights the significance of effective communication and data exchange in the design process, encompassing both human-human and human-machine interactions. These findings shed light on the diverse aspects of representing architectural design using various tools. They emphasize the process of mapping abstract ideas to visual representations, capturing and nurturing emergent design ideas, and facilitating the transfer and transformation of design data. Understanding these categories provides valuable insights into the ways designers leverage design tools to express, refine, and communicate their architectural concepts throughout the design process.

Within the superordinate category of ’Refinement of Design and Tools’ two categories emerge, each highlighting different aspects of the design process. The first category, ‘Evaluation Capabilities’ centers around the testing, simulation, and optimization features offered by computational tools throughout the design process. It explores how these tools enable designers to assess and refine their designs by providing advanced computational capabilities. Through the utilization of these tools, designers could simulate various scenarios, analyze performances, and optimize their designs based on specific criteria. This category underscores the advantages of computational tools in enhancing the evaluative aspects of the design process. The second category, ‘Feedbacks from Design Tool’ emphasizes the continuous evaluation, clarification, and improvement of design decisions as the design process unfolds. It recognizes the iterative nature of design and the importance of feedback loops in nurturing design outputs. This feedback can occur between different design tools, whether computational or conventional, and serves to inform and guide designers in refining their design choices. By actively incorporating feedback at various stages of the process, designers can iteratively improve their designs and ensure a more refined outcome. In brief, these findings highlight the significance of evaluation capabilities provided by computational tools, enabling designers to instantly test, simulate, and optimize their designs. Additionally, the findings underscore the importance of feedbacks from design tools, promoting continuous evaluation, clarification, and improvement of design decisions. In essence, designers refine their designs and their tools to enhance the overall quality of the final architectural outcome.

Under the superordinate category of 'Increasing design capacity' two subcategories emerged. The first subcategory, termed 'Offering catalog of options' pertains to the ability of computational design tools to present a comprehensive range of design alternatives. These tools empower designers by providing an extensive repertoire of diverse options, enabling them to explore a multitude of possibilities. The second subcategory, titled 'Expanding and exploring design search space' highlights how computational design tools broaden the scope of design exploration. By pushing the boundaries of traditional approaches, these tools open new horizons for designers to explore uncharted territories and unlock innovative design solutions. Through their capacity to expand the design search space, computational design tools facilitate creative exploration and contribute to pushing the boundaries of conventional design practices. Collectively, these findings highlight the capacity of computational design tools to enhance design processes by offering a vast array of options and transforming the way designers approach exploration and decision-making.

The next superordinate category is titled 'States of the design process' and includes two categories. The first category, 'Compatibility and consistency of tools with idea, design, and production' focuses on the alignment and coherence between design tools and the various stages of the design process. It encompasses the compatibility and consistency of tools in capturing and refining emergent design ideas, facilitating the maturation of thoughts, and enabling seamless transitions from conceptualization to production. This category emphasizes the importance of tools that seamlessly integrate with the designer's vision, design intentions, and the practical aspects of production. The second category, named 'The transformation of the design process' explores the transformative effects of computational design tools on the overall design process. It examines how these tools bring about changes and shifts in the way designers' approach and engage with the design process. By harnessing the capabilities of computational design tools, designers experience a paradigm shift in their methodologies, workflows, and problem-solving approaches. This category sheds light on the transformative potential of computational design tools to reshape conventional design processes and foster new modes of creativity and innovation. To sum up, these findings highlight the significance of tool compatibility and consistency throughout the design process, ensuring a smooth and coherent transition from ideation to production. Furthermore, they underscore the transformative nature of computational design tools, displaying their ability to revolutionize and redefine the way designers conceive, develop, and realize their design visions. Considering the findings, which illustrate the transformative potential of computational design tools and their ability to reshape conventional design processes, it becomes evident that architectural space is not a prominent and thoughtful focus.

The final superordinate category, titled 'Changes in the profession of architecture', emphasizes significant transformations within the architectural profession. It examines changes in organizational structures, working practices, and the role of architects in terms of expectations and services provided. Additionally, it explores the evolving relationships between design, tools, and designers when utilizing computational design tools, identifying both shifting dynamics and enduring aspects. Furthermore, the category explores the incorporation of outsourced concepts in architecture, particularly those derived from mathematics, which have influenced and enriched architectural discourse and practice. These findings provide valuable insights into the dynamic nature of the architectural profession, emphasizing the need for architects to adapt to new practices, embrace innovative concepts and tools, and shape the future of the field.

When we examine the emerging categories and subcategories, together with the integration of computational tools, the responses indicate that the design process has undergone a transformation and the dynamics in the architectural profession have changed.

While this study is has always focused on representation through design tools, with the advances in artificial intelligence, computational thinking itself has begun to change. New developments, such as artificial intelligence and evolving design tools, continually bring the issue of space back into focus, keeping it on the agenda. Although processes that reshape the way of thinking, representing, and defining architectural space have emerged, the existential relationship that architecture establishes with space continues.

Computational or not, no single tool alone can support our understanding of space in design in a holistic way. The important thing is for the designer to be aware of the advantages and disadvantages of the tools they use. It is essential to work not by obeying the tools but by compromising with them. The focus should be on achieving the designer's desires, needs, and demands by mastering their strengths and weaknesses.

However, some points specific to the computational design process should also be noted that the design process carried out with these tools is more indirect than conventional tools. Computational design process includes mediated actions. In addition, the distance between the designer and the final product which is considered here the architectural space is more distant and indirect.

Another worth to be mentioned topic is the delegation of the design process which is also could be re-evaluated with artificial intelligence developments. In some periods during the design process with computational design tools, which could be predicted that will increase even more with artificial intelligence, the design action is delegated to the tools. It seems that in the future the boundaries between natural and artificial distinctions between humans and tools in terms of authorship and expertise will become increasingly blurred. It is important also to remain critical and aware of these boundaries in architecture.

The transitivity and connectivity of the tools that facilitate and mediate the interaction between our thoughts and the physical world are among the key findings of this study. The design of architectural space represents a prominent example of the transition from the abstract to concrete. Within this context, it is essential to underline the significance of recognizing the relationship that these tools establish with our minds, bodies, and physical spaces.

5.4. Research Limitations

There are few practitioners and experts who carry out a project from beginning to end within computational design. These experts employed the computational tools either for parts of the design or for well-defined design problems. Although they reflect their individual architectural and spatial concerns in their practices, they are often in the supporting role rather than the leading designers.

Although it was desired to reach as many participants as possible, participation was limited due to the busy work schedules and time constraints of many of the invitees. The list of experts who use computational design tools in the architectural design process is already not a large set, those who accepted the invitation, therefore, are not as many as it was hoped for.

In addition, the written online interview method was employed. It is a method that provides effective use of time for both participants and researchers. While this method provides an advantage, it was observed that some participants gave short answers to some questions and did not support any follow-up questions and interactions.

The accessibility of the computational design tools that the study focuses on could be considered as another limitation. Access to tools is also a financial issue, but the financial dimension was not raised by the participants. Two topics that can be related to this issue which were mentioned by the participants are that computational design tools have become the standard of the architectural practice and that they democratize it. It is obvious that these tools are not accessible to everyone in practice, but this has not hindered the proliferation of computational design tools. Moreover, despite their limited accessibility, they have been perceived to democratize the practice of architecture, in the opposite way. This is an interesting topic, but emphasis of the analysis was not placed on this issue.

Lastly, at the time when the study was conducted and the data were collected, artificial intelligence studies were not as effective and widespread as they have become in the latter stages of the thesis. Opinions about artificial intelligence were not prominently featured in the data collected from the participants. Nevertheless, artificial intelligence technologies rapidly integrated into various domains, including design tools. Therefore, there is a need for further research to draw comprehensive and in-depth conclusions about the impact of artificial intelligence on computational design tools and architectural space.

5.5. Future Studies

As future studies, follow-up interviews can be conducted through focus groups and in-depth interviews highlighting selected issues and concepts. Another option is that a single interviewee could be selected, and the computational design process could be followed closely using ethnomethodological research tactics.

Architectural education, which is day by day more integrated with computational design thinking and design tools, could benefit from the implications of the dissertation. The evaluations presented on how different user profiles evaluate the space in architectural design with computational design tools could also be considered within the educational setting. The architecture curriculum could support the acquisition of practices for the interrelation and interoperability of various means of representation.

In addition, emergent artificial intelligence capacities should also be discussed and integrated with architectural design. Comparative and hybrid methods can be tried on the reliability and consistency of the written answers, not only in quantitative analysis, but also in customized trained models that can be developed.

Finally, findings of the research could be evaluated within the computational design tool providers. The individual, unique user experiences of the participants could also guide the development of new products and future of the sector.

5.6. Implications

This research aims to bring attention to architects who utilize computational design tools, fostering their interest in the concept of architectural space. By doing so, the vast and latent potentials of this unity could be explored without negating the essence of architectural space. Each identified category in the study should be considered as an independent avenue for progress and further research, contributing to the advancement of computational design. Furthermore, computational design tools enable the exploration of architectural space in new and diverse dimensions, facilitating the discovery of uncharted territories while advancing existing theoretical and practical research. The research opens the door to inferring different definitions and meanings of space, considering virtual, digital, and metaverse environments. Considering that even a post human condition is being discussed beyond the developments in AI and virtual space(s), the question of whether space is only for humans is a question that needs to be further explored with the advancement of tools in everyday practice.

REFERENCES

Abdelmohsen, S. M. A. (2011). *An ethnographically informed analysis of design intent communication in BIM-enabled architectural practice*. Georgia Institute of Technology.

Aish, R. (2005). From intuition to precision. *AA FILES*, *52*, 62.

Aysever, R. L. (2003). Dil felsefesinin geleceğine bir bakış. *Hacettepe Üniversitesi Edebiyat Fakültesi Dergisi*, *20*(2), 127–140.

Braun, V., Clarke, V., Boulton, E., Davey, L., & McEvoy, C. (2021). The online survey as a qualitative research tool. *International Journal of Social Research Methodology*, *24*(6), 641–654.

Brey, P., & Søraker, J. H. (2009). Philosophy of Computing and Information Technology. *Philosophy of Technology and Engineering Sciences*, 1341–1407. https://doi.org/10.1016/B978-0-444-51667-1.50051-3

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, *1*(2), 67–69.

Burry, M. (2011). *Scripting cultures: Architectural design and programming*. John Wiley & Sons.

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. sage.

Chaszar, A., & Joyce, S. C. (2016). Generating freedom: Questions of flexibility in digital design and architectural computation. *International Journal of Architectural Computing*, *14*(2), 167–181.

Clancy, S., & Brown, W. (2008). Translation: DNA to mRNA to. *Nature Education*.

Coates, P. (2010). *Programming. architecture*. Routledge.

Corbin, J. M., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, *13*(1), 3–21.

Corbusier, L. (1948). *New world of space*. Reynal & Hitchcock.

Corbusier, L. (2004). *The Modulor and Modulor 2*. Birkhäuser Basel. https://books.google.com.tr/books?id=exnMTOE_t-EC

Creswell, J. W. (2008). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications. https://books.google.com.tr/books?id=rg2nygAACAAJ

Cross, N. (1982). Designerly ways of knowing. *Design Studies*, *3*(4), 221–227.

Dahan-Dalmedico, A. (2011). Mathematics and the sensible world: representing, constructing, simulating. *Architectural Design*, *81*(4), 18–27.

De Bono, E. (1970). Lateral thinking. *New York*, 70.

Derix, C. (2014a). *Empathic space: the computation of human-centric architecture*. John Wiley & Sons.

Derix, C. (2014b). The space of people in computation. *Architectural Design*, *5*(84), 14–23.

Derix, C., & Izaki, Å. (2014). New curricula: syntonic systems. *Architectural Design*, *84*(5), 122–129.

Derix, C., & Jagannath, P. (2014). Near futures: associative archetypes. *Architectural Design*, *84*(5), 130–135.

Forty, A. (2004). *Words and Buildings: A Vocabulary of Modern Architecture*. Thames & Hudson. https://books.google.com.tr/books?id=J7VvQgAACAAJ

Frascari, M. (2011). *Eleven Exercises in the Art of Architectural Drawing: slow food for the architect's imagination*. Taylor & Francis.

Frazer, J. (1995). *An evolutionary architecture*.

Frazer, J. (1997). The Groningen Experiment, *CAADRIA '97: Proceedings of The Second Conference on Computer Aided Architectural Design Research in Asia*. April 17-19. pp. 345-353.

Geertz, C. (1973). Chapter 1/Thick Description: Toward an interpretive theory of culture. *The Interpretation of Cultures: Selected Essays*, 3–30.

Goodman, N. (1978). *Ways of worldmaking* (Vol. 51). Hackett Publishing.

Guest, G., MacQueen, K. M., & Namey, E. E. (2011). *Applied thematic analysis*. sage publications.

Güvenç, M. (2013, June 10). *Söylenebilir olanın bir sınırı vardır*. Mimarizm.Com.

Hillier, B. (2007). *Space is the machine: a configurational theory of architecture*. Space Syntax.

Izaki, Å., & Helme, L. (2014). Encoding user experiences. *Architectural Design*, *84*(5), 114–121.

Keller, S. (2012). Beauty, Genius, Epigenesis: The Kantian Origins of Contemporary Architecture. *Journal of Architectural Education*, *65*(2), 42–51.

Kiyak, A. (2008). *Describing the Ineffable: Le Corbusier, Le Poème Electronique and Montage*.

Knight, T., & Stiny, G. (2001). Classical and non-classical computation. *Arq: Architectural Research Quarterly*, *5*(4), 355–372.

Kotnik, T. (2010). Digital architectural design as exploration of computable functions. *International Journal of Architectural Computing*, *8*(1), 1–16.

Lefebvre, Henri. (1991). *The production of space*. Blackwell.

Legendre, G. (2011). *Mathematics of space*. John Wiley & Sons.

Lorenzo-Eiroa, P. (2013). Form in Form: On the Relationship between Digital Signifiers and Formal Autonomy. *Architecture in Formation: On the Nature of Information in Digital Architecture*, 11–22.

M Rocker, I. (2006). Calculus-based form: an interview with Greg Lynn. *Architectural Design*, *76*(4), 88–95.

Mathews, S. (2006). The Fun Palace as virtual architecture: Cedric Price and the practices of indeterminacy. *Journal of Architectural Education*, *59*(3), 39–48.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, *22*(3), 276–282.

McLuhan, M., & Fiore, Q. (1967). The medium is the message. *New York*, *123*(1), 126–128.

Mitchell, W. J. (1990). *The logic of architecture: Design, computation, and cognition*. JSTOR.

Moholy Nagy, L. (1928). 1947 (4th edit.). *The New Vision, George Wittenborn, New York*.

Moholy-Nagy, L. (1947). *The New Vision 1928: 1947*. Wittenborn.

Norberg-Schulz, C., & Loci, G. (1980). Towards a phenomenology of architecture. *New York: Rizzoli*.

Ortony, A. (1993). *Metaphor and thought*. Cambridge University Press.

Oxman, R. (2006). Theory and design in the first digital age. *Design Studies*, *27*(3), 229–265.

Parisi, L. (2013). *Contagious architecture: Computation, aesthetics, and space*. mit Press.

Pask, G. (1969). *The Architectural Relevance of Cybernetics, AD, 7 (6)*. MIT Press, Cambridge.

Perez-Gomez, A., & Pelletier, L. (1992). Architectural representation beyond perspectivism. *Perspecta*, *27*, 21–39.

Picon, A. (2011). Architecture and mathematics: Between hubris and restraint. *Architectural Design*, *81*(4), 28–35.

Schumacher, P. (2008). The concept of space in architecture: emergence, hegemony and transcendence. *Bologna: Compositori*.

Schumacher, P. (2009). Parametricism: A new global style for architecture and urban design. *Architectural Design*, *79*(4), 14–23.

Schwarzer, M. W., & Schmarsow, A. (1991). The Emergence of Architectural Space: August Schmarsow's Theory of" Raumgestaltung". *Assemblage*, *15*, 49–61.

Semper, G. (2004). *Style in the technical and tectonic arts, or, Practical aesthetics*. Getty Publications.

Sententia, W. (2004). Neuroethical considerations: cognitive liberty and converging technologies for improving human cognition. *Annals of the New York Academy of Sciences*, *1013*(1), 221–228.

Stiny, G. (2006). *Shape: talking about seeing and doing*. MIt Press.

Stiny, G., & Gün, O. Y. (2012). An Open Conversation with George Stiny About Calculating and Design. *Dosya: Computational Design*, *29*, 6–11.

Stöckli, T. (1992). *The Measurable and the Unmeasurable or From Form to Design to Existence*. 55–62.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research techniques*.

Terry, G., Hayfield, N., Clarke, V., & Braun, V. (2017). Thematic analysis. *The SAGE Handbook of Qualitative Research in Psychology*, *2*, 17–37.

Terzidis, K. (2006). *Algorithmic architecture*. Routledge.

Tomasello, M. (1999). The human adaptation for culture. *Annual Review of Anthropology*, *28*(1), 509–529.

Tuttle, R. H. (2023, March 28). *Human Evolution*. Encyclopedia Britannica. https://www.britannica.com/science/human-evolution

van Schaik, L. (2014). How can code be used to address spatiality in architecture? *Architectural Design*, *84*(5), 136–141.

Wıttgenstein, L., & Aruoba, O. (2006). *Tractatus logıco phılosophıcus*. Metis.

Yazar, T. (2009). *Mimari tasarım stüdyolarında sayısal egzersizler* [Doktora]. Yıldız Teknik Üniversitesi.

APPENDICES

APPENDIX A

CODING GUIDE

Coding guide for emergent categories and super ordinate categories.

Reasons for preference of computational tools		
1 Imagination related reasons	**Definition**	**The reasons for using computational tools which are related to the creativity, freedom, and novelty that computational tools offer in terms of new, unique and unconceived designs.**
	Example	**P9:** *To create architectural form that has hitherto not been seen, that could not have been produced, that could not have been conceived of.*
2 Pragmatic reasons	**Definition**	**The reasons for using computational tools which are related to the production, fabrication, and construction processes of architecture in terms of material, time and labor advantages.**
	Example	**P11:** *The primary reason for implementing computational methods on everyday practice is time efficiency. ... The same logic applies to post-rationalization techniques, where the process might carry on for months if a computational approach wasn't*

		utilized. The most important aspect for designers using computation is the translation of each elements to numerical values. This then enriches the digital model with further embedded information and facilitates the smooth and detailed fabrication of the architects design.

Using different design tools together		
3 Modes and combinations of design tool use	**Definition**	**Alternately using digital or analog design tools, computationally, or not, with different modes and purposes. The combinations of tools are either complementary, combinatory, integrated, or simultaneous.** **Analog + Analog Tools (sketching + drawing + physical model)** **Digital + Analog Tools (computers+sketching)(sketching +digital model)** **Digital + Digital Tools (computers+production machines) (software+software)** **Tools-Use** **Analog-Computational, Analog-Noncomputational,** **Digital-Computational, Digital-Noncomputational**
	Example	**P16:** *We have to assume that digital tools are important but we can not work without hand sketch and also physical model, in order to explore our drawing to reach the*

		goal that we have in mind. Basically it's a sort of loop, you can start scripting then sketch or viceversa, then modeling or detailing, it really depends by the task you are assigned or what do you need to do.
4 Chronology and order of use of design tools	**Definition**	**Determining which design tools are used in which phases of the design process for which purposes or services.**
	Example	**P2:** *We start with sketching and as we work in an networks in different places we brainstorm on that basis, but very soon we jump into the computer, physical modelling comes far after. If needed we start scripting to test variations on concepts.*
5 Capacities of design tools	**Definition**	**Instances of the limits and constraints of different design tools whether digital or analog, computational, or not which could be related with why designers customize and combine tools.**
	Example	**P4:** *Well, computational tools offer me the method to conceive my ideas in 3d - more accurately than sketching. They are more often used as a method of modeling, rather than for inspirational reasons.* **P17:** *Sketching out all options based on the pseudo code becomes nearly impossible because permutations and combinations of iterations can become very complex quickly.*

6 Design of the design tool	**Definition**	**Accepting the design tool as one of the active actors of the design process which is shaped by designer (customization / hybridization) and shapes designer (fixed / predetermined).**
	Example	**P9:** *It's a cyclical iterative process. First creating a tool, then using it to create a design or to create permutations of the design. The design is iteratively refined. And in many cases, the tool is successively modified or expanded to enable a new feature in the design.*

Representation with design tools		
7 Mind mapping medium	**Definition**	**Mapping of intangible thoughts, abstract data or ill-structured initial design ideas to get closer to visual representations of architectural space via digital or manual, computational or non-computational tools.**
	Example	**P1:** *I use computational tools as ways to visualise abstract data into some kind of graphic language, then to experiment with these outcomes as they are gradually being translated to space.*
8 Externalization of emergent design ideas	**Definition**	**Capturing and following-up of an emergent design idea while keeping the designer intuitions through the design process.**

	Example	**P4:** *My design process usually starts with an idea formulated through manual sketching or a mental image. Quite quickly after that I try to move to 3d, either by manual modeling or small form-creating scripts. Depending on the idea.*
9 Transformation and transfer of design data	**Definition**	**Considering design as an accumulation of transferable data. Different modes of design data exchange exist like between humans (colleagues, teams, seniors/juniors, instructors/students, etc.), between machines and humans or between machines and software/things (non-human).**
	Example	**P14:** *Computational tools allow for interpreting numerical data into design. This gives the possibility for more efficient designs, which fulfill specific parameters. If the parameters have been set correctly by the designer the final result is proved to be optimized to the given criteria.* **P12:** *The designer has to be able to describe his concepts in such a way that it can be verified, quantified and transferred via lean data exchange.*

Refinement of design and tool		
10 Evaluation capabilities	**Definition**	**Testing, simulation, and optimization facilities of**

		computational tools through the design process.
	Example	**P10:** *it extends my creative ability to look for the sweetspot of my design ideas. I can create more otpions and evaulat them better for the aspects I like to see in my design work -whetehr energy, cost or spatial performance.*
11 Design tool feedbacks	**Definition**	**Continuous evaluation, clarification, and improvement of the design decisions through the design process until it is refined. Nurturing design outputs with the iterative and informative feedback loops (between different design tools computational or conventional).**
	Example	**P1:** *Anything may come up first so it doesn't matter where one starts from, but most importantly each one should inform the other. Design is an iterative process that involves recursive testing happening across different platforms, analogue and digital alike.*

Increasing design capacity		
12 Offering catalog of options	**Definition**	**Availability of listing all various possible design options with computational design tools.**
	Example	**P17:** *Having an algorithmic design process, computational design tools enable the visualization of complex geometric relationships based on*

		simple mathematical relationships. The result is never a master-piece but a catalog of options that often surprise the initial logic.
13 Expanding and exploring design search space	**Definition**	**Computational design tools expand the design search space and change the direction of exploration.**
	Example	**P5:** *At the same time on different types of projects, we used digital tools to establish design outcomes that were beyond my initial design projections, and opened up entirely new possibilities.*

States of the design process		
14 Compatibility and consistency of tools with idea, design, and production	**Definition**	**Compatibility and consistency of tools with idea generation, design development and production which includes phases such as capturing emergent intuitive design idea, conceptual development, representation, and complete production.**
	Example	**P1:** *Ideas may emerge in each step performed to translate abstractions to more and more concrete graphics and spatial definitions. The produced space is a refined step along this process from abstract to concrete.*
15 The transformation of the design process	**Definition**	**Changes and transformations in the design process with the use of computational design tools.**

	Example	**P12:** *Each line in the script is a design decision, based on pre-conceived spatial ideas. Computational tools help to evaluate and optimize those pre-set decisions. The number-crunching computational tools are extensions to our brains, which are much slower in specific algorithmic operations. With the computation we design as if on speed, we can design on the fly, we can enjoy instant information feedback in the design to production process.*

Changes in the profession of architecture		
16 Changes in the profession of architecture	**Definition**	**Changes in the profession of architecture in terms of issues such as organization, working practice, way of working, and in the role of the architect in terms of expectations, services, naming themselves, etc.**
	Example	**P5:** *I became an architect before the use of desktop computing and drawing. As such, I am used to thinking about figure and form, space and atmosphere, material and affect, without feeling the need to "go to the computer" to explore what is attainable. That said, we (in my office and in my teaching) continually rely on computational tools and scripts to explore new possibilities, in ways that simply might not be possible to image without these tools.*

17 Changing modes of relations between design, tool, and designer	**Definition**	**Identifications of the changing and conserved relationships between design, tool and designer while using computational design tools.**
	Example	**P5:** *Design intent directs the tool, but the tool also has the power to challenge initial ideas, or to open up new directions.*
18 Outsourced concepts in architecture	**Definition**	**The outsourced (spolia) concepts of architecture which especially originated in and derived from mathematics.**
	Example	**P10:** *we have a long standing interest in topological complexity and the perception of this - maybe in contrast to simple euclidian shapes, which dominate the traditional architetcure. Works like the Dermoid are exampel of this. Furterh spatial experiences include the work of space, color and accoustics - as in the Distortion projects, which were tested in street festivals with massive amount of people.*

APPENDIX B

SAMPLE INTERVIEW

INTRODUCTION

The use of computer technologies and computational thinking in architecture has been more pervasive since the 1960s and its transformative implications have yet to be fully discussed in architectural theory. Design as a way of thinking is exposed to a considerable transformation. Architectural jargon has become associated with parameters, algorithms, relational input/output structures, data oriented models, digital tools and production systems. However, some constitutive elements of architecture, especially the concept of space, have not been emphasized as they were earlier.

Space is one of the main tenets of architecture since Modernism. Architectural education has been formulated primarily around the concept of space, which as an uncontested concept in architectural theory, is also one of the most discussed ones. Space is a polysemic term which includes many meanings and interpretations in one word.

Although the computational process might sometimes cause unexpected surprises, it is actually based on precision and prediction. Computational procedures need to be explicit, transparent and traceable in programming language. Space is embedded in computational design practice through algorithmic procedures and computable parameters. However, instead of being a constitutive tenet of the design process, space might have become an after-the-fact product in computational design approaches of contemporary architecture. Architectural object is usually composed of physical concrete elements such as surface, shell, mass, module, piece, system details etc... in computational design practice. Yet, the descriptive and prescriptive approach in computational design practice mainly deals with the "How" questions about the production and realization of architectural form. In some instances consideration of architecture as tangible objects led to a lack of concern for space, which is more intangible than tangible. It is worth to question how computational design today reckons with space as a multi-meaningful, undefinable, perpetual and ineffable concept.

Computational design proved itself by feasibility and constructability in architectural practice but the concept of space as an ephemeral concept has not yet been scrutinized in the computational paradigm. This inquiry will explore the concept of space as it is re-constituted in the paradigm of computational design practice. The assumption is that space has both computable and uncomputable aspects and that it is worth exploring how computational practice treats both aspects.

INTERVIEW QUESTIONS

1. TOOLS AND DESIGN PROCESS: Architects have been using sketching and physical models during the design process since at least Renaissance. Both of these representational tools, which are themselves spatial, allow designers to think about space through spatial devices. With the prevalent use of computational tools we see the use of new representational tools such as scripting, algorithms, and parametric formulations which are essentially non-spatial representations.

1.1. *What are the reasons why you use computational tools in your design process? Is it more efficient, does it facilitate the generative process, does it help in visualizing and spatial thinking, does it enhance creativity etc…*

Comments:

Yes it is more efficient, yes it facilitates the generative process, yes it helps visualizing, and yes it enhance creativity on certain conditions. And it helps spatial thinking but only when the designer thinks from within the process, meaning that the digital designer needs to define the relations between all the components. Since the components define the physical boundaries of a space, whether open space, semi-open space or enclosed space, setting the rules of the behavior of the smallest building components includes the spatial design concept. Yet I must say that not many are aware of this and build relationships blindfolded not knowing what exactly how much they fall back into conventions without even developing their own view on what these relations could be. So the proper answer is that it depends on the state of mind of the designer, spatial design concepts reside between the ears, and are then transferred to algorithms and project specific software.

1.2. *In your personal design process, how do you benefit from or use these conventional spatial representations and the newly emerging computational tools? Do you sketch something first, or do you do modelling / coding / scripting first or do you use all of these simultaneously? If you prefer certain tools explain how they are used during your design process and if you prefer more than one tool how and why you go back and forth between different tools?*

Comments:

We do all of these simultaneously and with a small team that is well prepared to respect each other's expertise on an equal basis. Nothing comes first, we work in a horizontal structure. I will never give my nerds a sketch and then say, now translate this into a parametric model. It does not work like that. As a designer I need to think like a programmer in order to find the initial design concept. I need to define the relationship between the points of the point cloud, becoming the vertexes that can be tweaked in the later parametric model. And likewise we start scripting from scratch as to understand and to manipulate the logic of the interactions between the components. Our design to production procedure links the design intent directly to the manufacturing methods in a bi-directional way, as to inform the design process by the capacities of the CNC machines. This bilateral machine to machine communication is based on lean data exchange as to avoid that information gets lost in translation.

2. TOOLS AND SPATIAL THINKING: Every tool, every representational system we create simultaneously opens and closes different ways of thinking, which in turn define who we are. As often attributed to McLuhan, "We shape our tools and then our tools shape us." Following McLuhan's idea, our imaginations of space are at least partially shaped by the tools we use.

2.1. *Could you describe the importance of computational tools for you while imagining architectural space with regards to your design philosophy, formal repertoire, building technology etc.? Do computational tools help you conceive an image that you cannot think, draw, produce without them or do computational tools help you enrich your imagination process?*

Comments:

Clearly yes, I agree fully with McLuhans' statement on the tools. I would even go further and state that language as a tool to communicate is directly connected to the necessity to label things. We are in our evolution strongly interlaced with things, tools, devices, instruments and machines. And yes computational tools change the way I conceive space. In principle I stated to think of space as weightless, as being able to be rotated, put upside down, stretched, bent etc. Many computer manipulations have become part of my thinking, and I think that is a clear enrichment. I think much more precise now, and much more free from physical constraints.

2.2. *When do your spatial ideas and images emerge in your design process? Could you describe the process, relations, and negotiations between the computational tools and your understanding of space? How do the computational design tools foster spatial thinking? Could you illustrate it through examples of your own work?*

Comments:

Spatial ideas are more and more based on the experience of spatial arrangements in time and over time. For example, the A2 Cockpit building [2005] in the acoustic barrier in Utrecht in The Netherlands is based on the concept of a 60 seconds experience of driving along that structure. In those 60 seconds many things happen: one experiences the snakelike structure to become concave, then convex, then a featured fold line is introduced to tweak the reflections for the upper and lower part of the glazed barrier – using featured foldlines as car designers do - and then continues into the pumped up volume of the luxury car showroom, then taken over by another foldline etc. Continuous variation and gradual transformations are key factors in our designs. This idea of spatial continuity is also represented in the interior where we made an endless loop, a spatial lemniscate as we did before in the Waterpavilion [1997].

2.3. *What is the importance of the quality of space in your design approach and how do computational design tools ensure the quality of space?*

Comments:

They don’t, it is all in the instructions the designers give to the computational tools. Each line in the script is a design decision, based on pre-conceived spatial ideas. Computational tools help to evaluate and optimize those pre-set decisions. The number-crunching computational tools are extensions to our brains, which are much slower in specific algorithmic operations. With the computation we design as if on speed, we can design on the fly, we can enjoy instant information feedback in the design to production process.

2.4. *Could you describe spatial concerns that are personally significant to you and could you describe how your personal spatial concerns and anticipations are fed into the computational process? Please illustrate through examples of your own work?*

Comments:

Taking advantage of the speed of calculations in the computer we can now embark on an architecture where all constituting components are unique in shape and dimensions, even in material, structural and aesthetic performance. This a disruptive radical revolution that will change the architectural profession completely. Not only the appearance of the built structures, but also the way we work together in distributed teams of experts, and the way everyone wherever in the world can take part in the design and production process, which can open up to the preferences of individual people.

3. UNDERSTANDING OF ARCHITECTURAL SPACE IN THE COMPUTATIONAL DESIGN APPROACHES

3.1. *Which spatial qualities, concepts, values and norms are best represented and examined through computational tools and which ones are not? What are some emerging aspects of architectural space and some alterations in our understanding of space which might have been triggered by the computational design approach? Please illustrate with examples if possible?*

Comments:

The designer has to be able to describe his concepts in such a way that it can be verified, quantified and transferred via lean data exchange. If the designer is not able to do this him/herself, he/she will lose control and be left to the limited spatial skills of others in the building chain. Our aim is to explode the building chain, to unleash the building chain and operate as a distributed network of expert designers and producers, communicating exclusively via databases. That is the new spatial language designers will have to make their own. The new language is HTML, Java, C++, Grasshopper, XML, Python or any other programming vehicle as to exchange data in their most essential form.

APPENDIX C

PRELIMINARY INTERVIEWS

__Questionnaire 1__

Informative Speech about Interview: Space and Computation

Space is a blurry and ambiguous concept but is defined and constituted many times. It has not one completed satisfactory explanation but has many convincing consistent conceptions. Space is physical (perceived; nature; spatial practice; daily routines), social (lived; representational space) and mental (conceived; representation of space; abstractions; scientists, planners etc.). Computational design methods altered the designers' thought dynamics and system. How computational design practice constituted its own space concept? Interviews of this inquiry will trace and explore how space is constituted in computational practitioner's mind. Are there any different understandings of space and what are its constituents? Do you think is there anything beyond computable? Is space computable for you?

1. Personal Background Interviewee Information

1.1. Education and profession background (schools, offices etc.)

1.2. How long has it been they work in computational architectural design practice?

1.3. Hardware and software experiences and preferences (companies, materials, methods, tool etc.)

1.4. How do you take job offers, where and how do you work?

1.5. What are the steps of their workflow? (Work routines)

1.6. What kind of projects you have worked whether computational or not? (Work diversities or product range)

1.7. How you describe yourself as an architect, coder, designer, all or none of them, or something else?

2. Architectural Space Questions: without any reference to computation or computational architectural design concepts, architectural space in their mind

2.1. How they begin their projects and do they have any specific point of their own to start?

2.2. What the priorities are of their own at start concerns like light, context, performance, form, functionality, materials, tools etc. and is it same for all projects?

2.3. What are the priorities of their profession?

2.4. What is your definition of architectural space and what architectural space means for you?

2.5. What 'the quality of space' and 'the sense of space' means for you. Not only as a designer but also as an experiencer of architectural space?

3. Computational Architectural Design Questions: any alteration in architectural design and how computational architectural design represents itself.

3.1. Could you define your design approach and methods that you use?

3.2. Is there any difference or alteration after computers integrated to the design process? Difference between computational thinking and computerization is considered.

3.3. Is computational architectural design instituted its own faculty?

3.4. What are the advantages and disadvantages of computational architectural design throughout the design process?

3.5. What are the positive or negative qualification criteria of any computational architectural design product or process?

3.6. How do you start when you take design problem at first especially how computational approach raise in yourself? How could you describe it rational, intuitional or something else?

3.7. How is it changed your way of thinking? Descriptions of their personal experiences in computational architectural design

3.8. Do you feel any conflict between architectural and computational concerns?

4. Some Topics to Discuss

<u>Questionnaire 2</u>

1. Personal Design Process / Approach

1.1. What are the steps of your design process?

1.2. How do you start your project / work? Do you sketch something first, or do you do modelling / coding / scripting? Do you try to reach an image, or do you follow abstract ideas and formula without thinking the end product?

1.3. How do you describe / define / formulate your design problem at first?

1.4. What are the parameters of your design approach?

1.5. Why do you prefer / adopt computational design approaches in your profession?

2. Personal Understanding of Space

2.1. Could you list your main concerns / topics / parameters that shape your understanding of space?

2.2. What are your priorities while creating spaces?

2.3. What defines / creates / formulates space in your mind?

2.4. What are the parameters of space you consider while designing?

2.5. Are there any models / formula that define your space conception?

2.6. What does ‘the quality of space’ mean for you? As a designer or as a user.

2.7. Could you describe your characteristic and personal spatial concerns and your understanding of space in relation to computational design?

3. Concept of Space in Computational Design

3.1. Which concerns are considered in computational design approach and how do you relate these to architectural space?

3.2. What are the essentials and nonessentials of space in computational design approach? Is there any difference conceiving architectural space in computational paradigm? What are the differences of conceiving space in computational paradigm?

3.3. Is there any difference conceiving architectural space with or without computational thinking / tools?

3.4. Which one affects conceiving of architectural space more: computational tools or computational thinking or something else?

3.5. What are the emergent criteria of space in computational paradigm? If there are any could you describe criteria of space emerged and just possible / feasible in computational design?

3.6. If space has always computationally constituted then what alterations are revealed in computational paradigm? What has changed in case of conceiving of space in designer's mind?

Questionnaire 3

Introductory: The use of computer technologies and computational thinking in architecture has been more pervasive since 1960s and some have even claim it had transformed the architectural practice. Its transformative implications have yet to be fully discussed in architectural theory. Design as a way of thinking is exposed to a considerable transformation. Architectural jargon became associated with parameters, algorithms, relational input/output structures, data-oriented models, digital tools and production systems. However, some constitutive elements of architecture, such as space, has not been emphasized as it was earlier.

Space is one of the main tenets of architecture since Modernism. Architectural education has been formulated primarily around the concept of space, which as an uncontested concept in architectural theory, is also one of the most discussed ones. Space is a polysemic term which includes many meanings and interpretations in one word.

Although computational process might cause unexpected surprises sometimes, actually it is based on precision and prediction. Computational procedures need to be explicit, transparent, and traceable in programming language. Space is embedded in computational design practice through algorithmic procedures and computable parameters. It is worth to question how computational design reckons with space as a multi-meaningful, indefinable, perpetual, and ineffable concept.

Computational design proved itself by feasibility and constructability but the concept of space as an ephemeral concept has not been scrutinized yet in the computational paradigm. There is a lack of research on understanding space in the computational architecture. This inquiry will explore the concept of space which is being re-constituted in the paradigm of computational design practice while the practice has promised new ways of thinking about space. The assumption is that space has both computable and incomputable aspects and that is worth exploring how computational practice treats both of these.

1. Design Process: Here, the design process of the interviewee is inquired. Questions refers to bodily actions and mental activities that are performed or methods, tools, materials, etc. that are used through the design process. It is expected that the questions will prompt the interviewee to discuss their early design process.

1.1. How do you start your work and what are the key steps of your design process?

Çalışmaya nasıl başlarsınız ve tasarım sürecinizin anahtar adımları nelerdir?

1.2. How would you describe your design process?

Tasarım sürecinizi nasıl tanımlarsınız?

1.3. How do you develop and shape your thoughts and ideas through your design process?

Tasarım süreciniz boyunca düşüncelerinizi ve fikirlerinizi nasıl geliştirir ve şekillendirirsiniz?

1.4. How do you define and formulate your problem definitions, design schemes and solutions?

Problem tanımınızı, tasarım şemalarınızı ve çözümlerinizi nasıl tanımlar ve formüle edersiniz?

2. Design Approach: These questions are focused on personal design approach of the interviewee. Their design criteria, design understanding, preferences and design priorities are asked to inquire about their design philosophies. These answers could help trace the emergence of architectural space in designers' mind till its embodiment in built/drawn architecture.

2.1. How do you describe your own design approach?

Kendi tasarım yaklaşımınızı nasıl tanımlarsınız?

2.2. Which concepts, values and norms are primarily considered in your design approach?

Tasarım yaklaşımınızda hangi kavramlar, değerler ve ölçütler öncelikli olarak dikkate alınır?

2.3. How and when does architectural space come into play within your design approach?

Tasarım yaklaşımınızda mimari mekan nasıl ve ne zaman devreye girer / ortaya çıkar?

3. Understanding of Space (spatial imagination and qualities of space): In this part it is expected to find / explore personal spatial understandings of the interviewee. It is also expected to detect similarities and differences in the computational architectural practice between various interpretations of architectural space.

3.1. How do you describe / define architectural space and what does it mean to you?

Mimari mekanı nasıl tanımlarsınız ve mimari mekan sizin için ne ifade eder?

3.2. How do you characterize your personal understanding of space?

Kişisel mekan anlayışınızı nasıl nitelendirirsiniz?

3.3. Which parameters do you handle while imagining architectural space through your projects? Which spatial qualities, concepts, values and norms are prioritized in your design approach?

Projelerinizde mimari mekanı hayal ederken hangi değişkenleri / parametreleri ele alırsınız? Sizin tasarım yaklaşımınızda hangi mekansal kaliteler, kavramlar, değerler ve ölçütler öncelik kazanmaktadır?

3.4. What comes first to your mind when do you think of architectural space, and could you illustrate it through examples of your own work?

Mimari mekanı dendiğinde aklınıza ilk gelen şeyler nelerdir ve kendi projelerinizden biri üzerinden örnekleyerek açıklar mısınız?

4. Concept of Space in Computational Design: In this part it is expected to inquire how computational paradigm affected architectural space and imagining architectural space. It is also expected to detect alternative understandings of architectural space in computational design.

4.1. Why do you prefer to work with computational design approaches? What does attract and motivate you in computational design in case of architectural space? Could you illustrate it through examples of your own work?

Niçin mesleğinizde hesaplamalı tasarımı tercih ediyorsunuz? Hesaplamalı tasarımda sizi mimari mekan konusunda neler cezbeder ve motive eder? Kendi işlerinizden örnekler vererek açıklayabilir misiniz?

4.2. Could you describe how your personal spatial concerns and anticipations are fed into the computational process? Could you illustrate it through examples of your own work?

Mekansal ilgilerinizin ve sezilerinizin hesaplamalı tasarım sürecine nasıl dahil olduğunu tarif eder misiniz? Kendi işlerinizden örnekler vererek açıklayabilir misiniz?

4.3. Which spatial concerns are primarily considered in computational design paradigm / approach / design?

Hesaplamalı tasarımda hangi mekansal ilgiler / kaygılar öncelikle dikkate alınmaktadır?

4.4. What are the new aspects of architectural space which might be highlighted in computational design paradigm / approach / design? Could you illustrate with examples if possible?

Hesaplamalı tasarımda mimari mekanla ilgili vurgulanan gereken yeni düşünceler / görüşler nelerdir? Örnekler vererek açıklayabilir misiniz?

4.5. If space has always computationally constituted, then what alterations in our understanding of space are revealed in computational paradigm? What has changed, if any, in imagining space?

Eğer mekan her zaman hesaplamalı olarak oluştuysa dijital / sayısal çağda mekan anlayışımızda ne gibi değişimler ortaya çıktı? Mekanı düşlerken, eğer varsa, neler değişti?

APPENDIX D

INTERRATER RELIABILITY PROCESS INSTRUCTIONS

INSTRUCTIONS TO REVIEWERS

1. Introduction

2. Research Questions

3. Coding Instructions

4. Coding Guide

5. Sample Interview

1. Introduction

This study is focused on the conceptualization and imagination of the architectural space within computational design practice. For this purpose, the experiences and opinions of people who are authority in using computational design tools are consulted. The research employs qualitative research methods and follows the principles of the thematic analysis. The qualitative data evaluated within the scope of the research is obtained from the written online remote interviews. The interrater reliability process involves the analysis of a selected interview according to the coding guide provided below. In order to validate coding and analysis processes, the rater is invited to engage in a sequence of sessions that encompass individual coding, followed by collective discussions and evaluations.

2. Research Questions

This research is shaped around two main questions:

1. How architectural space is imagined in computational design practice?

2. How architectural space is considered as a multi-layered, loosely defined, and ineffable concept in computational design approaches in which procedures are precise, descriptive, and prescriptive?

The first question is oriented to reveal the processes, factors and actions that related to the conceptualization and imagination of architectural space in computational design practice. The second question investigates the consistent and contradictory points while evaluating the concepts of space and computation from an architectural point of view.

3. Coding Instructions

The attached document provides a guide including the set of categories -emerged from codes used by the primary researchers- of this study in analyzing responses to the interview. The guide includes the definitions of categories and their superordinate categories. The rater is expected to become familiar with the categories by reading the guide. One of the competent participants interviews is selected as a sample which includes comprehensive answers to interview questions. The reviewer is asked to read the transcript carefully and use the provided categories and any other additional categories that he/she sees appropriate. The reviewer is asked to carefully mark the segments of transcriptions and indicate the associated categories. The markings can be made on a hard copy, on the digital MS Word file, or by using a coding software.

Following this individual coding exercise, a meeting will then be held between the researchers and the reviewers to look at and validate this coding scheme according to their interpretation.

Thank you for your participation.

APPENDIX E

INTERRATER RELIABILITY PROCESS FIRST ROUND CODING STUDY

these simultaneously? If you prefer certain tools explain how they are used during your design process and if you prefer more than one tool how and why you go back and forth between different tools?

20 **Comments:**

(4) Chronology and order of
(3) Modes and combinati

21 My design process usually starts with an idea formulated through manual sketching or a mental image. Quite quickly after that I try to move to 3d, either by manual modeling or small form-creating scripts. Depending on the idea. My main tools are Rhino and Grasshopper, but often I have to use RhinoScript or Python also. I use Processing and Java also, but not in formal or structural architectural design. 'My design process uses all available methods simultaneously, and often is a mixture of 3D, rendering and sketching.

22 **2. TOOLS AND SPATIAL THINKING:** Every tool, every representational system we create simultaneously opens and closes different ways of thinking, which in turn define who we are. As often attributed to McLuhan, "We shape our tools and then our tools shape us." Following McLuhan's idea, our imaginations of space are at least partially shaped by the tools we use.

23 **2.1.** Could you describe the importance of computational tools for you while imagining architectural space with regards to your design philosophy, formal repertoire, building technology etc.? Do computational tools help you conceive an image that you cannot think, draw, produce without them or do computational tools help you enrich your imagination process?

24 **Comments:**

(5) Capacities of design too

25 Well, computational tools offer me the method to conceive my ideas in 3d - more accurately than sketching. They are more often used as a method of modeling, rather than for inspirational reasons.

26 **2.2.** When do your spatial ideas and images emerge in your design process? Could you describe the process, relations, and negotiations between the computational tools and your understanding of space? How do the computational design tools foster spatial thinking? Could you illustrate it through examples of your own work?

27 **Comments:**

(3) Modes and combinati
(14) Compatibility and co

28 Spatial ideas emerge as a natural part of the design process, and for me it is hard to separate the use or effect of tools in that process. Spatial development is a mental process, which is fed by 2D and 3D representations of the space.

APPENDIX F

INVITATION TO PARTICIPATION

Dear ……………,

I am a PhD candidate at the Architecture Program of Izmir Institute of Technology, Turkey. I am interested in the conceptualizations and representations of space in computational architecture.

As part of my doctoral research, I am investigating how leading architects who use computational means in their design process benefit from these means in imagining architectural spaces. I am working with Assoc. Prof. Dr. Fehmi Dogan from the same university. I am in the process of collecting data related to my research topic through online questionnaires, interviews, and focus group discussions.

Your answers to questions below will be of utmost value for the progress of my research, which hopefully will make a modest contribution to ever growing field of computational architecture.

If these questions proved to be of interest to you and if you could spare some time, please either answer the questions online (LINK) or send me back the attached word file with your answers. I will also appreciate if you drop me a quick note in case you agree to participate in the research.

In case you have further questions please feel free to contact me at any time.

Yours sincerely,

Kadir Ozturk, PhD candidate
Department of Architecture
Izmir Institute of Technology

CURRICULUM VITAE

Name Surname: Kadir ÖZTÜRK

EDUCATION

2010 – 2023 PhD in Architecture
Izmir Institute of Technology, Faculty of Architecture
PhD Program in Architecture

2008 – 2010 Master of Architecture (MArch)
Istanbul Bilgi University,
Institute of Science and Technology
Master's Program in Architectural Design

2003 – 2008 Bachelor of Architecture (BArch)
Izmir Institute of Technology, Faculty of Architecture
Department of Architecture

1999 – 2003 High School, Konak Anatolian High School

ACADEMIC PUBLICATIONS

"Investigation of the Concept of Space Through Designer Approaches in Digital Design Applications"
12th National Symposium on Digital Design in Architecture, June 21-22, 2018 (pp. 103-113).
Kadir Öztürk, Fehmi Doğan (2018).

"Space Beyond the Computable"
8th National Symposium on Digital Design in Architecture, June 26-27, 2014 (pp. 33-40).
Kadir Öztürk, Fehmi Doğan (2014).

PROFESSIONAL EXPERIENCE

Izmir Metropolitan Municipality
Historical Environment and Cultural Assets Department
Izmir, Turkey, July 2011 – present

Buca Municipality
Urban Planning Directorate
Izmir, Turkey, February 2011 – June 2011

Duran-İltir Architecture Office
Izmir, Turkey, June 2010 – December 2010

G – G Technologies
Efficient Use of Renewable and Natural Energy Resources
Izmir, Turkey, April 2008 – June 2008

Asmira Architecture and Restoration Office
Izmir, Turkey, July 2007 – October 2007

İvme Architecture Office
Izmir, Turkey, October 2006 - February 2007

COMPETITION EXPERIENCE

Çankaya Municipality Presidency Service Building, Art Center, and Ulvi Cemal Erkin Concert Hall Architectural Design Competition
November 2010
Fourth Mention Award
with Kıvılcım Duruk, Tolga İltir, and Kaan Duran.

Düzce Chamber of Commerce and Industry Service Building Project Competition
July 2010
First Mention Award
with Kaan Duran.

Denizli Government Building Architectural Project and Urban Design Project for Surroundings Competition
July 2009
First Round Elimination
with Tolga İltir.

Concrete Design Competition 3
May 2008
Achievement Award
"Responsive Concrete"

Manisa Municipality Service Building, Trade Center, and Urban Space Arrangement Competition
February 2007
Second Mention
with Tolga İltir.