DISTRIBUTED EXPERTISE OF COMPUTATIONAL PRACTICES IN ARCHITECTURAL DESIGN TEAMS

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
Livanur ERBİL ALTINTAŞ

July 2023
İZMİR

We approve the thesis of **Livanur ERBİL ALTINTAŞ**

Examining Committee Members:

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

Assoc. Prof. Dr. Tonguç AKIŞ
Department of Architecture, İzmir Institute of Technology

Prof. Dr. Ebru ÇUBUKÇU
Department of City and Regional Planning, Dokuz Eylül University

Prof. Dr. Leman Figen GÜL
Department of Architecture, İstanbul Technical University

Prof. Dr. H. Murat GÜNAYDIN
Department of Architecture, İstanbul Technical University

19 July 2023

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

Assoc. Prof. Dr. Ayça TUNÇ COX
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

I am sincerely grateful to all those who have contributed to the completion of this thesis, both through their guidance and unwavering support. This work would not have been possible without the collective efforts of a multitude of individuals, each of whom played a significant role in shaping my academic journey.

First and foremost, I extend my deepest appreciation to my advisor, Prof. Dr. Fehmi Doğan, whose guidance, expertise, and continuous encouragement propelled me through the various stages of this research. His insightful feedback and dedication have been invaluable, and I am truly fortunate to have had the opportunity to work under his mentorship.

I am equally indebted to my co-advisor, Assoc. Prof. Dr. Altuğ Kasalı, whose unique perspectives and insightful discussions greatly enriched the direction of my research. His unwavering commitment to excellence has left an indelible mark on this thesis.

I am deeply thankful to the members of my thesis committee, Prof. Dr. Ebru Çubukçu and Assoc. Prof. Dr. Tonguç Akış, for their time, expertise, and critical insights during the thesis process. Their thoughtful questions and suggestions have significantly enhanced the rigor and quality of this work.

Sincere thanks go to my jury members Prof. Dr. Leman Figen Gül and Prof. Dr. H. Murat Günaydın for their careful consideration, and thoughtful comments.

I would like to thank Melike Altınışık and Gökhan Avcıoğlu, who made my research possible, and the office staff for their endless patience and support throughout the research process.

I am also grateful to Prof. Dr. Erdem Erten for his unwavering support and advises. Throughout my academic life at Izmir Institute of Technology, the unique guidance he provided was invaluable.

The fellows of room 107, including Ece Ceren Engür, İlknur Uygun, Onurcan Çakır, and Yelin Demir Altıntaş, provided me with their friendship, support, and moments of rest and joviality that I needed during the intense stages of the research. Their unwavering belief in my abilities motivated me, and for that, I am truly grateful. I would also like to sincerely thank Batuhan Taneri, who has been my companion since I embarked on my academic journey and has consistently provided support and encouragement.

I extend my heartfelt gratitude to my family. Their unending love, support, and understanding have been my pillars of strength throughout this academic journey. Their sacrifices and encouragement have propelled me forward, and I am proud to have you by my side.

My eternal and profound gratitude goes to my beloved husband Burhan Altıntaş. I am sincerely thankful for his unwavering patience, his genuine eagerness to assist me, and his constant support that propelled me forward in times of utmost need. This journey would have been insurmountable without his presence by my side.

Thank you all for being a part of this milestone in my academic and personal growth.

ABSTRACT

DISTRIBUTED EXPERTISE OF COMPUTATIONAL PRACTICES IN ARCHITECTURAL DESIGN TEAMS

This study focuses on how knowledge is produced and distributed among participants with different skills and knowledge bases, how knowledge is distributed through designers, tools and representations to define the role and tasks of each team participant, and the nature of interaction within the team focused cognitive system. The thesis explores cases involving computational tools in architectural design using ethnographic methods, focusing on understanding how a distributed cognitive system facilitates multidisciplinary collaboration within design teams.

One aspect of the study delves into designers' use of computational design tools in a collaborative work environment, seeking to comprehend how these tools enable serendipitous design explorations. The team's management of design development processes and serendipity is analyzed, particularly how the system generates multiple alternatives in an explorative setting, which influences the extent of exploration.

Second, the thesis focuses on generating and evaluating alternative design options in computational design applications. Office and team leaders play an important role in making the design idea and process more legible and transparent for team participants, clients and consultants. The study also explores how design ideas are represented and externalized in the distributed cognitive system to achieve a legible schema that guides the design process. It is important to recognize the space in which they can improvise, along with the transparency of the design process, among team participants who specialize in different fields.

Third, the study explored various aspects of computational applications in architectural design and their impact on collaborative processes within distributed cognitive systems. By examining the interactions between multidisciplinary team participants and the role of both digital and non-digital tools, the research provides insights into how multidisciplinary design teams navigate in their creative activities.

Keywords: Distributed Cognition; Architectural Design Teams; Computational Design; Design Process; Design Cognition

ÖZET

MİMARİ TASARIM EKİPLERİNİN HESAPLAMALI PRATİKLERİNDE DAĞITILMIŞ UZMANLIK

Bu çalışma, bilginin farklı beceri ve bilgi tabanlarına sahip katılımcılar arasında nasıl üretildiğine ve dağıtıldığına, bilginin tasarımcılar, araçlar ve temsiller aracılığıyla her ekip üyesinin rolünü ve görevlerini tanımlamak için nasıl iletildiğine ve odaklanılan bilişsel sistem içindeki etkileşimlerin doğasına odaklanmaktadır. Tez, etnografik yöntemler kullanarak mimari tasarımda hesaplama araçlarını içeren durumları keşfeder ve dağıtılmış bir bilişsel sistemin tasarım ekipleri içinde çok disiplinli işbirliğini nasıl kolaylaştırdığını anlamaya odaklanır.

Araştırmanın bir yönü, tasarımcıların işbirliği ortamında hesaplamalı tasarım araçlarını nasıl kullandığını inceler ve bu araçların tesadüfi tasarım keşiflerini nasıl sağladığını anlamayı amaçlar. Ekibin tasarım çözümünü geliştirme süreçlerini ve tesadüf keşifleri nasıl yönettiğini, özellikle sistemde keşif yapıcı bir ortamda birden fazla alternatifin nasıl üretildiğini ve bu durumun keşfin derecesini nasıl etkilediği sunar.

İkinci olarak tez, hesaplamalı tasarım uygulamalarında alternatif tasarım seçeneklerinin oluşturulmasını ve değerlendirilmesini ele alır. Ofis ve ekip liderleri, tasarım fikrini ve sürecini ekip katılımcıları, müşteriler ve danışmanlar için daha okunabilir ve şeffaf hale getirmede önemli bir rol oynarlar. Çalışma, tasarım fikirlerinin dağıtılmış bilişsel sistem içinde nasıl temsil edildiğini ve dışa vurulduğunu inceleyerek tasarım sürecini anlamayı amaçlar. Farklı alanlarda uzmanlaşan ekip katılımcıları arasında tasarım sürecinin şeffaflığı ile birlikte doğaçlama yapabilecekleri alanı tanımak önemlidir.

Üçüncü olarak, bu çalışma mimari tasarımda hesaplamalı tasarım araçlarının çeşitli yönlerini ve dağıtılmış bilişsel sistemler içinde işbirlikçi süreçlere olan etkilerini inceler. Multidisipliner ekip katılımcıları arasındaki etkileşimleri ve hem dijital hem de dijital olmayan araçların rolünü inceleyerek, çok disiplinli tasarım ekiplerinin tasarım süreçlerini nasıl yönlendirdiğine dair bulgular sunar.

Anahtar kelimeler: Dağıtılmış Biliş; Mimarlık Tasarım Ekipleri; Hesaplama Tasarım; Tasarım Süreci; Tasarımda Biliş

To my son, Kuzey

-may you never stop exploring.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure	**Page**

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND STATEMENT OF THE PROBLEM

Computational design tools have taken on a role in shaping both products and collaborative design processes. In recent years, the computational design tools have taken an important place not only in the individual practices of designers but also in interactions within multidisciplinary teams, empowering participants to explore innovative ideas and potentially enhancing the overall quality and efficiency of design outputs. Architectural design, by its nature, is open to collaboration and inevitably draws from other disciplines (Cuff, 1992; Groat & Wang, 2013). Many new technologies that have emerged with today's advancements are rapidly being integrated into architectural design processes, shaping the nature and routes of design processes. Now, there is less emphasis on the concept of the star architect. With emerging new technologies, there can be participants in architectural teams that actively contribute as experts through their computational design knowledgeand skills, regardless of their level of experience in the field. Therefore, in architecture, the nature of design processes is changing especially in formation of teams.

The profession of architecture typically involves a level of collaboration with other professions and embraces multidisciplinarity within the design process even starting from the initial visioning phases. In fact, it is no longer possible to mention designers who have expertise in a single knowledge domain. Increasingly versatile and multidisciplinary individuals are participating in design processes (D’souza, 2020). An architect might not only engage in coding but also has mastery in other professional areas such as graphic design, illustration, and more. Even individuals who may not have a strong design background but can provide productivity, and contribute to the production process of design idea are also part of teams. Previously, inexperienced architects who had recently completed their architectural education would join teams to develop their design skills. However, in recent times, they not only aim to strengthen their design skills but also specialize in particular technological tools, even without necessarily emphasizing their design expertise, contributing as experts cpncerning the digital tools to be employed in design development. As a result, in each design process, the paths to finding solutions are constantly being reimagined, formed, and the dynamics and organization within teams are repeatedly established. In these environments where disciplinary diversity is increasing, team participants seek and create new communication channels among themselves.

Individuals specialized in the field of architecture but unfamiliar with emerging computational design tools incorporate these tools into the process, providing new ways and directions in design processes. Through computational design tools, the ways of initiating the design process and progressing through it may have differentiated from the past. In traditional design processes, architects often produce sketches to transfer their initial design ideas or develop them through physical models(Yaneva, 2005). However, in recent times, computational design tools involved in the design process offer designers a design process that starts with a series of rules through an algorithm and accompanied three-dimensional visualizations of the algorithms.

Architectural tools have great potential to understand knowledge and task distribution strategies among team participants, both human and non-human, and to facilitate multidisciplinary mechanisms and problem-solving strategies in architectural design teams. Computational design tools facilitated new communication strategies and ways among team participants by integrating new languages. In order to introduce novelty and innovation, designers are always in a state of exploration. However, it is also crucial for designers that the design concept and the intended purpose of the product are understandable. In collaborative design processes involving computational design tools, designers strive to both explore endless possibilities offered by these tools and complete the process with a comprehensible design. In this dissertation, the use of computational design tools in architectural design teams has been investigsted to understand the dynamics within multidisciplinary design teams.

1.1. Problem Definition

This study focuses on how information is produced and distributed among participants with different skill sets and knowledge bases, how knowledge is transferred through designers, tools, and representations to describe the role and tasks of each team participant, and the nature of the interaction within the focused cognitive system. It particularly investigates the following question: “Throughout different stages of the compuational design process, how different tasks and knowledge are distributed among experts and how they dynamically collaborate?”

Design is considered as a cognitive process consisting of interaction, computation, generation, communication, synthesis, and manipulation of tasks (Cross, 2006; Lyon, 2005, 2011). In architectural design processes, designing and construction stages involves various participants from different disciplines, who contribute to the solutions of design problems pursuant to their responsibilities defined by their disciplines.

Regarding this framework of interdisciplinary nature of the profession, a critical question stands out: How do architects compromise with experts from other disciplines in the design process in relation to design approach, method, representation tools, and systems? According to Cuff (1992), an architect or a designer can have a leading role in some of the design decisions but there is significant coordination and collaboration almost at every step of the practice. Architects can be experts in many subjects such as aesthetics, site planing, functional programming, structural design, mechanical systems, visual communication among others, but it is obvious that architects constantly need expert contributions from other disciplines (Cuff, 1992). Recently with the advancement in digital technologies enhancing interaction, in design practices many experts from different domains can collaborate anytime and anywhere. With the use of digital technologies and the involvement of different expertise domains, architectural design become a more sophisticated system, which in turn needs to be investigated in its own right.

This research focuses on architectural practices as a system in order to understand the complex mechanisms within computational design processes. Analytically, researching architectural project production mechanisms as a system requires a holistic view. The system that comprises humans, objects, and tools can be elaborated as a Distributed Cognitive System (Hutchins, 1995). Distributed cognition assumes that any task can be distributed to parts of the system in planning and execution (Hutchins, 2014). Through this view, collaboration is not only cognized among human participants; it also includes other elements constituting the system such as tools and representations. This research considers design process as a distributed cognitive system and discusses interactions among tools, representations, and other participants in the way they pursue a design project.

Based on Hutchins' work particularly and distributed cognition research in general, design is considerated as a cognitive system involving problem solving through interaction, communication, computation, synthesis, and production in a context where individuals, design tools, and representations undertake complementary tasks. Design process is a knowledge-production process among multiple actors in collaborative environments (Lyon, 2011). Most of the studies in distributed cognition (Hollan et al., 2000; Hutchins, 1995a; Kirsh, 2010) are focused on understanding external and internal representations that work together to construct and coordinate complicated social systems. Excellence in design emerges when close and remote knowledge domains are brought together (Cuff, 1992). In architectural design teams, collaborating designers might have changing roles in the design process. Shifts in the roles are possible within team participants in the design process. Now, with digital technologies, shifts within the team could be less possible because of specialization in different domains which could result in irreplaceable and unsubstituted roles in the design process.

Moreover, newly emerging architectural representation systems, such as algorithmic code languages and complex systems of digital representation tools, brought forward different expertise in design (Burry, 2003). These new representational systems are also used considerably in design and changing the representational systems in architectural design process (Oxman, 2006). In design teams, participants who are specialized in different areas work collaboratively and use different representation systems in the design process. Lately, some of the outstanding architectural design offices have collaborated with specialist participants from remote disciplines and have used different representation systems or have even invented new architectural programs such as CATIA (Loukissas, 2009).

In comparison to most studies focusing on collaboration, knowledge distribution in design interdisciplinary teams is less overlapped among the participants and each participant is specialized in a particular domain. As a consequence of the specialization in different domains, it is possible to mention the once hierarchical order present in many architectural teams has gradually decreased (Yaneva, 2009). Correspondingly, it can be said that in architecture the necessity of collaboration among individuals who are specialized in distant disciplines is increasing gradually (Derry, Schunn, & Gernsbacher, 2014; Paulus, 1999). In reference to the distributed cognition research, rather than hierarchy, the design work is analyzed through an overlapping system that includes tools and representation systems. Compared to other domains of distributed tasks such as piloting (Hutchins, 1995b) and navigation (Hutchins, 1995a), designing is a creative idea generation process in which diversity is increased by maximizing contributions from team participants and at times minimizing a hierarchical organization. In a distributed cognitive system, which is supposed to ensure creativity through the right amount of focus on variation and on decision making requires a sensitive balance between divergence and convergence, and between a hierarchical and horizontal organization. This study could make new contributions in distributed cognition framework highlighting these aspects of design collaboration.

The research uses qualitative methods, primarily ethnographic field techniques, highlighting authentic aspects of distributed cognitive systems of the studied design teams. The study inquires into design teams' communications, knowledge transferring approaches, and representation techniques in the design process. Ethnographic research method involves extended observations of a group, through observing day-to-day lives of people, and interviewing group participants (Brewer, 2003; Creswell, 2007; Sommer & Sommer, 1997).

1.2. Research Focus and Research Questions

The general research question of the study is as follows:

"Throughout the different stages of the architectural design process, how different tasks are distributed, how different experts dynamically collaborate in a design process, and how is the professional knowledge distributed among them?"

Specifically, the research inquires the following questions:

1. What is the nature of collaboration in architectural design teams?
2. How different tasks are distributed among experts are distributed and how these experts dynamically collaborate across design phases?
3. What are the representation systems in architectural design teams? How do design team participants employ different representation systems in executing particular tasks?

4. How do interdisciplinary teams generate and coordinate representations collectively in the context of architectural design?

1.3. Research Goals and Objectives

The main goal of the dissertation research is to explore how architectural design teams distribute knowledge among team participants through the adoption of computational design tools in practice. This study aims to explore the goals and challenges in the design process and communication within collaborative design teams. This is achieved through the following objectives that encompass investigating the structure of teams, the roles of participants, the impact and integration of computational and other design tools into the design process, the flow of information between different disciplines, and communication mechanisms. The objectives are also concerned with exploring the nature of collaboration and interaction among team participants and identifying the contribution of different disciplines in design practice.

There are many studies that have questioned the nature of design collaboration (Binder et al., 2013; N. Cross & Cross, 1995; Cuff, 1992; Fischer, 2005; Milliken, Bartel, & Kurtzberg, 2003; Paulus & Nijstad, 2003; Yaneva, 2009b). Cuff (1992), in her seminal work on the profession, investigated architectural practice within the framework of an ethnographic study using an organizational analysis of the architectural firms. Cuff (1992) concludes her ethnographic observations with “thick descriptions” of collaborative tasks. She engages with the complex nature of problem solving in architecture. Another ethnographic research is Yaneva’s work (2009b). Yaneva employs the actor-network theory and provides thick descriptions of activities in the architecture firms. Another related study conducted by Kasali and Nersessian (2015) uses distributed cognitive system theory. Kasali and Nersessian applies ethnographic approach to study a multi-disciplinary design team operating in socio-cognitive environment. The study aims at understanding the nature of complex interactions within interdisciplinary design teams. The study is focused on distribution of knowledge that emerges through interactions of experts who have different disciplinary backgrounds. The study findings suggest that designers “strategically” employ a variety of representations to solicit feedback from participants with different disciplinary backgrounds in healthcare design practice.

Within the scope of this study, the design teams with participants specialized in different areas were interviewed and observed in their authentic work environments to understand the distribution of knowledge in architectural design teams. In the conventional architectural design environment, there is a process starting from initial sketches to detailing of a project. In this distributed system, there is a particular representational system, 2D paper-based or digital drawing, to transfer ideas between team members. However, recently, there is a differentiation in representational systems in architectural design with the introduction of computational design representations, such as parametric and algorithmic. Designers have started to use algorithmic codes and digital tools. This research investigates the collaborative process between the developers of design ideas and the developers of digital tools in the design process within collaborative work. It adopts the lens provided by the theory of distributed cognition to examine collaboration in the architectural design process.

1.4. Structure of Dissertation

The dissertation is organized into 7 chapters including the introduction chapter. Chapter 2 presents a review of supporting literature. Chapter 3 presents the primary methods utilized in the dissertation, providing an overview of ethnographic research and its significance. It also includes an explanation of grounded theory coding and analysis, emphasizing their importance in the research. The chapter provides descriptions of the offices and the projects observed through the research in field. The chapter further presents codes and categories which emerged in coding phases. The chapter also presents reliability and verification, highlighting the steps taken to ensure the validity of the findings. Following Chapters 4, 5, and 6, this study presents core descriptive episodes and engages in discussions each chapter. Chapter 4 presents design exploration mechanisms in architectural design teams, with a particular focus on computational design tools. This chapter presents the various methods and approaches used in design exploration, revealing their importance and impact in architectural practice. It explores the role of computational design tools in the design discovery process. Additionally, the chapter discusses the challenges and potential implications of using these tools in architectural design teams. Chapter 5 presents and discusses how architectural teams utilize computational design tools to conduct their design processes and develop design products with the goal of achieving legibility. Through an in-depth analysis, this chapter provides thick descriptions and discussions on how architectural teams navigate the challenges and opportunities associated with using computational design tools to achieve legibility in their design processes and products. Chapter 6 presents and discusses the mechanisms of knowledge sharing in multidisciplinary team environments through descriptive episodes. It explores how knowledge is shared and distributed among team participants from different disciplines within the context of the architectural design process. In a collaborative work environment, the chapter examines different collaboration mechanisms that arise from the overlap or non-overlap of knowledge domains. Chapter 7 concludes by summarizing the findings of this dissertation and provides recommendations for future research routes.

CHAPTER 2

BACKGROUND

This chapter provides a literature review of topics related to the basic inquiry of the research. Section 2.1 presents an overview of existing research related to professional design practice and design methods. Section 2.2 presents the distributed cognition theory, which constitutes the theoretical framework of the thesis, and other relevant theories related to team works. Section 2.3 presents a discussion on collaboration and Section 2.4 provides a review of the significance of communication and representational systems in design teams.

2.1. Professional Design Practice

Dana Cuff (1992) conducted one of the seminal ethnographic studies on the architectural profession. In her study, she focused on the collaborative environment in architectural offices, embracing architectural practice as a holistic phenomenon. Cuff proposes that architectural practice emerges from a shared environment with multiple participants who carry out various tasks necessary for designing and constructing buildings. According to Cuff (1992), an individual designer may determine the appearance of a building, but issues related to practice, clients, and collective action influence how the design will be implemented. Traditionally, collaboration between architects and other disciplines is often considered to involve difficulties (Cuff, 1992). Similarly, the collaboration among architectural designers themselves is also a compelling situation. Each designer brings a peculiar understanding of design and his/her own method in the design process, which requires them to effectively communicate their ideas to other participants.

Recent developments in digital and computational technologies require an in-depth investigation of their significance and implications for architectural design as well in the way they may facilitate or impact collaboration between architecture and other fields. In this context, an experimental design research conducted by Sabin (2013) explored the intersection of architecture, computation, and science. The study primarily examined the relationship between code and pattern, material and geometry, as well as fabrication and assembly. The aim of the study was to establish a connection between computation, biology, and design, highlighting the interplay between these disciplines. In the study, Sabin (2013) explored the connections between weaving and computation, specifically within the context of architecture and woven forms. Sabin (2013) identified the potential relationships between architecture and the intricate patterns and structures found in woven textiles. While the ultimate objectives of architecture and science may differ, Sabin highlighted that disciplines, such as computation and biology, can offer valuable systems-based models for architecture to study and develop form, function, and structure. By drawing inspiration and knowledge from these disciplines, architecture can benefit from new insights and approaches to enhance its design and construction processes.

According to Sabin (2013), collaboration between architects and scientists holds great potential for productive exchanges in the field of design. Within this multidisciplinary context, Sabin's research led to the establishment of a hybrid architectural-biological research and design network known as LabStudio. LabStudio focuses on exploring architectural models and design tools that allow the study of micro-environmental architectures found in biological forms and their corresponding functions. By bringing together expertise from both architecture and biology, LabStudio aims to investigate and understand the intricate relationship between design and biological systems, paving the way for innovative approaches in architecture. The research undertaken in LabStudio pursues collaborative work that exposes designers and researchers to new modes of thinking and facilitates a deeper understanding of form and function within specific contexts. By engaging in collaborative endeavors, designers and researchers are able to break free from traditional disciplinary boundaries and explore innovative approaches that integrate multiple perspectives. This collaborative approach encourages cross-pollination of ideas, knowledge, and methodologies, fostering a richer understanding of how form and function intersect and influence one another.

At a prominent architectural firm, i.e., Skidmore, Owings & Merrill (SOM), architect Neil Katz has worked for many years, utilizing computational tools in design. SOM is an interdisciplinary firm that operates within a highly collaborative environment, and it has embraced the Architecture & Engineering Series (AES) software, which is specifically designed to facilitate collaboration (Katz et al., 2013). Katz participated in the Smartgeometry workshop (SG2006), which brought together designers interested in methods such as scripting and parametric modeling. The workshop specifically focused on the utilization of Generative Components (GC), a parametric CAD software. During the workshop, design and structure teams collaborated closely on the design abstractions of the model and shared information. Working together, they set out to create a collaborative environment using GC, pushing the boundaries of their abilities, and exploring new possibilities for collective design. The team was divided into two parts: the massing team, which focused on exploring form using algorithmic and optimization techniques to create a form that met specific criteria determined by the team, and the articulation team, which also employed algorithmic and optimization techniques to develop a skin and massing model. However, the articulation team had to adapt their work based on the alterations made in the design process. To facilitate collaboration and data exchange, the team established a virtual network. Within this collaborative environment, the team utilized tools and formats designed to facilitate data exchange, analysis, and visualizations during the design process (Katz et al., 2013). This collaborative and comprehensive work environment provided opportunities for generating design ideas among multiple participants. The digital tools employed by the team allowed for the visualization and understanding of the non-visible characteristics of the design idea, enhancing the representation of the model (Katz et al., 2013).

Some architectural offices, such as Herzog & de Meuron, develop computational tools in alignment with their architectural concepts. These offices collaborate with artists and experts from various fields to enhance their knowledge and skills in design (Peters, 2013). Within Herzog & de Meuron, the Digital Technology Group (DT) works in close synchronization with the design team throughout the entire project, from the initial design idea to the completion of the building construction (Peters, 2013). The DT group consists of a small team of 12 individuals who specialize in computer-aided design (CAD) management, building information modeling (BIM), parametric design and scripting, visualization and video, as well as digital fabrication (Peters, 2013). But DT group is not assigned with any architectural tasks, the group serves to facilitate architectural design (Strehlke, 2009). On the other hand, the design team is not to build digital design tools. The development of digital tools, as Strehlke (2009), the team leader of DT group, points out is focused on only creating architecture, and he states: “it is not a technology that we try to do something with it; the focus is more on design intent and the right tool, and develop the tool to make concept work" (Peters, 2013, p.58).

This study recognizes the changes in the architectural profession in the way digital technologies are more and more incorporated in various aspects of the design process and proposes to investigate the recent changes by way of looking at collaborative environment in the architectural profession as a distributed cognitive system.

2.2. Distributed Cognition Theory

People interact with other people, artifacts, technologies, tools, surfaces, and the things that are represented to others. People also interact with their environments as being 'embedded' to coordinate their internal cognitive tasks with external tools (Kirsh, 2008). A system, constituted by humans and their environment as a whole, is a distributed cognitive system. Edwin Hutchins and his colleagues introduced the *Distributed Cognition Theory* to describe and explain cognitive processes ongoing within such systems. The theory considers cognition as a process going beyond the limits of only human cognition, instead it proposes that cognition is distributed across internal individuals' minds, external cognitive tools, and groups of people, and across space and time (Hutchins, 1991, 1995a, 2004, 2006; Norman, 1991). People's cognitive activities results from interactions with external artifacts and with other people's activities in a task that are determined by socio-cultural contexts and physical environment that they are situated in (Hutchins, 1995a; Suchman, 1987). Distributed cognition discovers and explains the principles of coordination, externalization, representation, and interaction (Hutchins, 1995a), and frames a socio-technical system which consists of people working together, with certain tools and representational systems through the process (Hutchins, 1995a). Interactions between internal and external representations result in communications in a task (Hutchins, 1995a). Distributed cognition theory highlights the context-distributed nature of cognition between individuals and context (Hutchins, 1995a).

Distributed cognition theory frames the cognitive process of human and non-human mechanisms that are participate in a task (Hutchins, 2004). In the book Cognition in the Wild, Edwin Hutchins (Hutchins, 1995a) attempts to apply the principal metaphor of cognitive science, cognition as computation, to the operation of the navigational system. Hutchins (Hutchins, 1995a) believes that large computational system consists individuals' computational thinking which are part of this large system. He describes computation observed in the activity of the larger system as "computation realized through the creation, transformation, and propagation of representational states" (Hutchins, 1995a). According to Hutchins (Hutchins, 1995a), to understand navigation system, we need to understand information processing system within the organization. Hutchins (Hutchins, 1995a) refers to David Marr's view for information processing system. According to Marr (2010), there are three levels of description for information processing system: (1) computational theory of the task that the system performs (what system does, why it does it); (2) choice of representation for the input and output and the algorithm to be used to transform one into the other; (3) the details of how the algorithm and representation are realized physically. Hutchins (Hutchins, 1995a) gives examples from Western tradition of piloting to make us understand this abstract computational account. He points the importance of the representation of the system and the implementation of the computational system. Hutchins (Hutchins, 1995a) exemplifies navigation tools and explains their computational systems. He mentions 'mental track'(keeping position to unseen rotation), 'mind's eye' (knowing the position without enough information), and 'representational artifice' (projection of external and internal structure onto a single spatial image) of the navigators (Hutchins, 1995a). Hutchins (Hutchins, 1995a) also attempts to understand the navigation from a cognitive perspective by considering the whole suite of tools that are used in executing tasks.

Collaborative work can also be interpreted as a series of tasks undertaken by multiple actors, which are human and non-human actors, participating in a task. The theory of *Actor Network Theory*, is an approach developed by Bruno Latour, Michel Callon, and John Lawin in 1980s, which frames the multiple actors and mechanisms in a task (Ritzer, 2004), and is considered another way of looking at collaborative systems. Actor-network-theory is defined as "a conceptual frame for exploring collective sociotechnical processes, whose spokespersons have paid particular attention to science and technologic activity" (Ritzer, 2004: p. 1). Actor–network theory is an approach to social theory and research. The approach Actor Network Theory originated in science studies, which deals with objects as part of social networks. Latour, Callon and Law's analysis is a set of dialogues that describes the developing structure of a network which consists of both human and non-human actors and their interactions. Callon (1987) states that the structure of the network is "reducible neither to an actor alone nor to a network. An actor-network is simultaneously an actor whose activity is networking heterogeneous elements and a network that is able to redefine and transform what it is made of" (Callon, 1987: p. 94). Moreover, Actor–network theory differs itself from other sociotechnical approaches that emphasize the role of human and non-human elements. Latour (1996) mentions there should be the same analytical and descriptive framework about a human or a text or a machine: "an actor in Actor–network theory is a semiotic definition – an actant – that is something that acts or to which activity is granted by another…an actant can literally be anything provided it is granted to be the source of action" (Latour, 1996: p. 373). "Actor" can be human or nonhuman or an institution. Actors' characters and qualities are described as dialogues between representatives of human and non-human actants (Ritzer, 2004). Actant means human and non-human actors and they take shape in a network through their relationship with one another. Actants take part in networked connections and they describe, name and provide them with a content or an action or an aim or subjectivity (Ritzer, 2004). However, "actors" are conscious beings, and "actants" includes all kinds of autonomous figures, which are creating our world (Latour, 2005). Both terms can be used interchangeably. Actors can be anything that has ability to act both including people and material objects such as speeches, inscriptions (anything written), technical products, a human, things being studied, ideas, groups, professions, designs, skills, etc. The term "network" is defined very meticulously. It has two meanings. One is the technical meaning of network, which is used in electricity, trains, sewages, internet, and so on. The second one is used in "sociology of organization, to introduce a difference between organizations, markets, and states. In this case, network represents one informal way of associating together human agents" (Latour, 2005: p. 129). Moreover, the term "network" comes with its own problems. According to Latour (2005) the term "network" has unintended meanings. Firstly, it refers to the shape of network. Secondly, it implies 'transportation without deformations' in actor network theory, which is not possible because, actor- network requires numerous translations which results with deformations and changes. Moreover, networks are related to a process of building of activities which are acted by actors or actants (Ritzer, 2004). The networks that are created by actors have nodes and links. Each of them are being obtained semiotically and also they make the networks local, variable, and unsuspected (Ritzer, 2004).

Employing the actor-network theory in architectural design teams, Yaneva (2009) describes collaborative environment of architectural practice using “mundane trajectories” attributing to participants, objects, tools, and all that exist in a task. With thick descriptions of the observations, Yaneva (2009) tells mundane stories of design. Her purpose was generating an “infra-reflexive descriptions” of design practice. Yaneva (2009) makes analysis of interactions of human and non-human actors’ networks in a design office. Her research is carried out by meticulously observing the daily activities of designers in a design process which covers social, material, and cultural networks between multiple actors. Referring to actor-network theory of Latour (2005), Yaneva (2009) interprets the “social” as a connecting element, not a separate domain.

To make a comparison between Distributed Cognition and Activity Theory, first, both theories deal with humans, non-humans, and tasks (Nardi, 1995). Distributed Cognition begins by defining the system's goal, which is an abstract systematic concept. This concept does not involve the sensations of the system's participants. In Activity Theory, activity is shaped by an object held by a subject. Objects partially define the activity in Activity Theory. Distributed Cognition aims to understand how intelligence is sustained in a system; on the other hand, Activity Theory aims to describe social relations and processes. Both theories engage in a collaborative environment, but Distributed Cognition mostly focuses on “how” questions, while Activity Theory asks “why” questions in a collaborative task. In the Distributed Cognition theory, information processing in a system is directly observable and focuses on how information moves through the system. On the other hand, Activity Theory mostly focuses on how social relations are shaped in a system and how information is open to interpretation.

For both theories, the goal is a central focus. In Activity Theory, the focus is on the historical development of activity and the role of artifacts, where a tool mediates an activity. Distributed Cognition is concerned with similar notions. Hutchins (Hutchins, 1995a) highlights the roles of artifacts in a task, using the example of a cartographer who performs computations to create a chart for navigators to use. The navigators do not need to know how the chart was made, but the device becomes more powerful when its users do not know how and why it works. Thus, a tool becomes a task holder partially (Hutchins, 1995a). In the Distributed Cognition theory, human beings and artifacts are assumed to be conceptually equal, both acting as "agents" in a system (Hutchins, 1995a). In a system, artifacts and people collaborate in a task. On the other hand, Activity Theory assumes that people and artifacts are not equivalent. Artifacts belong to humans and serve as instruments in activities (B. Latour, 2005).

Another theory proposed by Clark & Chalmers (1998) is the Extended Mind theory, which explains systems from a human-centered viewpoint. It states that "human organism is linked with an external entity in a two-way interaction, creating coupled system that can be seen as a cognitive system in its own right" (Clark & Chalmers, 1998, p. 2). The Extended Mind theory explores a wide range of possible relations, both internal and external (Clark & Chalmers, 1998). According to Hutchins (2014), the main difference between Distributed Cognition and the Extended Mind theory is that the Extended Mind theory is considered a type of cognition, whereas Distributed Cognition is a perspective on cognition rather than a specific type. The Extended Mind theory assumes that there is a center in the cognitive system, which is the organism itself, while Distributed Cognition does not assume a center for any cognitive system (Hutchins, 2014). Distributed Cognition investigates how a cognitive process emerges from the interactions among elements in a system (Hutchins, 2014).

2.3. Collaboration in Design

Design, as a cognitive process, is a problem solving activity enacted between individuals in specific design contexts through interaction, computation, generation, communication, synthesis, and manipulation of tasks (Cross, 2006; Lyon, 2005, 2011). The design and construction process of any design project involves numerous participants who perform various tasks that are required to create a design idea (Cuff, 1992).

Architectural practice has been considered as a collective action activated by diverse social, environmental, formal and technical, and professional outcomes (Cuff, 1992). Architects collaborate in the design process with other designers who have different expertise in terms of design approach, methods, and use of representational systems. The collective action and collaboration with other professionals have been considered as a challenge because of disciplinary boundaries between architects and non-architects (Doctors, 2015). Architects also collaborate with other architects and non-human agents which are tools and representations. Collaboration among architects has been described as a social network, and the network includes not only architects but also engineers, machines, animals, arts, objects, humans and non-humans (Law, 2003). Cuff (1992) claims that architectural practice is a holistic process among multiple participants and states: “architectural practice is the everyday world of work where architecture takes shape” (p. 1). Cuff puts forward that architectural practice emerges in a distributed environment by multiple participants that carry out various tasks necessary to design and build the buildings. According to Cuff (1992) an individual designer could be determining what the building should be but all other issues of practice, clients, and collective action concern how the design will be applied. She highlights designers’ need of collaboration especially in the construction process and mentions that an architect could be an expert in different areas such as aesthetics, siting, function, structure, mechanical system, graphic conversations, and etc., but, to build a design idea, architect needs the assistance of other experts (Cuff, 1992). Even though many great buildings are known to be designed by an architect, Cuff (1992) points out that there are collaborators who undertake many tasks in the design and construction process. However, collaborative practice in traditional methods resulted in limited participations (Hight & Perry, 2013).

Algorithmic design tools have been part of the design process recently but, designers have used algorithms before digital tools were developed. Algorithm has a role in everyday activities even if it is not learned, because when we are faced with a problem; our priorities, values, dilemmas and experiences are structuring the solutions (Lave, 1988). According to Hutchins (Hutchins, 1995a) many problems in a distributed environment can be calculated, solved and stored, and built in tools and technologies. Because cognition is computational, and any task can be computable (Hutchins, 1995a). Moreover, the tools and technologies that have been developed are being used in ongoing activities with less calculations.

Recently, in the architectural design process, there is a growing number of designers who use algorithmic codes in digital tools. However, it is impossible to visualize what is behind the codes and what algorithmic codes define in all possibilities by mind. While different design tools are used in a design team, team participants should be communicating with each other. To communicate, designers use various representational systems. In the design process, designers might be switching between representation systems because of many reasons: to communicate with each other, to understand and represent the design idea in better ways, and to solve a design problem.

According to Shaw (2010) shared representations play important role in collaborative work because of constructing social interactions. His study shows how design emergence is facilitated through visual cognition, as suggested by Oxman (2002), and through collaborative emergence in conversation and performance, as advanced by Sawyer & DeZutter (2009). It also explores the concurrent nature of these two processes in design practice. The study suggests that shared representations and social interaction as conversation both strengthen the collaborative design (Shaw, 2010).

Architectural design involves an ongoing process among multiple actors, such as the client, designer, and consultant. Architectural design is a process that is distributed among these actors. According to Hutchins (Hutchins, 1995a) when knowledge is distributed in a social system among individuals, the task could be more guaranteed. Hutchins (Hutchins, 1995a) defines two kinds of distribution of knowledge in a system, one is overlapping knowledge distribution (Figure 2. 1), and the other is non-overlapping knowledge distribution (Figure 2. 2). The overlapping knowledge distribution is characteristic of cooperative works, and it avoids possible errors and interruptions. The overlapped knowledge distribution is a hierarchical system in which knowledge of the experts in a task decreases redundancy, and knowledge of the novice is more redundant (Hutchins, 1995a). Commonly, non-overlapping distribution of knowledge has been considered more effective but, it is a less robust system than overlapped one because, such systems lacks self-monitoring (Hutchins, 1995a). In terms of robustness, Hutchins refers to a system that succeeds in the work process (Hutchins, 1995a).

	Job A	**Job B**	**Job C**	
Person 1	○	○	○ ●	Knowledgeable ○
Person 2	○	○ ●		Performer ●
Person 3	○ ●			

Figure 2. 1. The interpretation of Hutchins's overlapping distributions of knowledge idea. Expertise level increases from Person 3 to Person 1.

(Source: The illustrations are drawn by the author.)

	Job A	Job B	Job C	
Person 1			○ ●	Knowledgeable ○
Person 2		○ ●		Performer ●
Person 3	○ ●			

Figure 2. 2. The interpretation of Hutchins's non-overlapping distributions of knowledge idea.
(Source: The illustrations are drawn by the author.)

Hutchins (Hutchins, 1995a) introduces this schemes of distribution of knowledge for well-defined tasks which are clearly separated from each other. The interesting point about design tasks is that they are ill-defined (Eastman, 1969). Some parts of the design task may be well-defined with some tasks clearly divided according to scale, or according to their content such as modelling, detailing. Design tasks, therefore, are decomposable into sub-tasks only partially. A second important point about design tasks relates to its aim to increase creativity by maximizing divergence (Guilford, 1973) where anybody could contribute to at any levels of design. Moreover, recently, domain expertise in design has changed and expertise in coding and algorithms is more and more in demand. This last change made redundancy in the system that Hutchins (Hutchins, 1995a) suggests as crucial for the functioning of the system, more difficult to attain (Figure 2. 3 and 2. 4). In the figures (Figure 2. 3 and 2. 4), P3 is the most knowledgeable participant, and it decreases to P1 as less knowledgeable about a task. In figure 2.4, Px is an outsider to the team but, Px contributes to the team its' expertise.

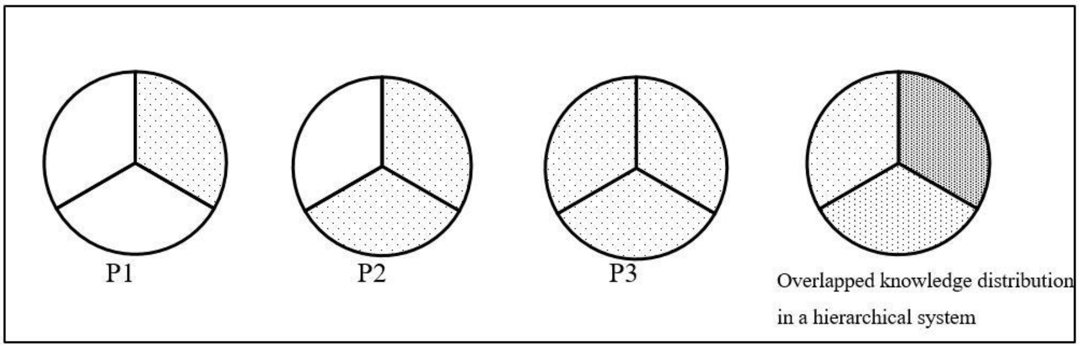


Figure 2. 3. Schematic representation of Hutchins's overlapped distribution of knowledge. (P: Person, filled area shows expertise domain of a person).
(Source: The illustrations are drawn by the author.)

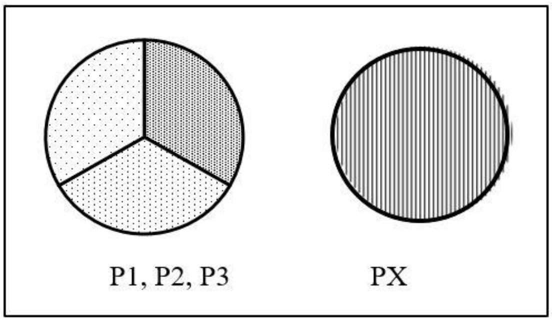


Figure 2. 4. Schematic representation of the distributed expertize in the design process. (P: Person; hatched area shows expertise domain of a person).
(Source: The illustrations are drawn by the author.)

Lately, how different expertise domains are integrated in design have been more and more important in architecture. It is observable that architectural design teams' knowledge distribution is less and less overlapped, because each designer could be specialized in a particular area in design, digital computational tools, especially coding. Traditionally, architects used to work with an expert heading a group of designers in a team (Cuff, 1992). Cuff (1992) mentions her observations in architecture firms and describes the design offices' team leaders as the lead architect in the design phase where the team leaders have less control in production phases of design project. However in the design process, Cuff (1992) puts forward that novice designers could not express their talents and thoughts because of the hierarchical team composition.

Recently, collaboration in architectural design has undergone a shift from a hierarchical system to a non-hierarchical system, primarily due to the integration of remote expertise domains, such as coding, into the design process. Within architectural design teams, collaboration is taking place among remote expertise domains, leading to an increasing need for a common language. To understand the distribution of knowledge between collaborating designers and between designers and tools in a task or process, it is crucial to define the roles and responsibilities of each agent and establish effective communication channels between them. Architectural designers may have to collaborate with experts from remote domains with whom they have no shared knowledge.

Design offices might have a clear distinction between design teams and digital design tool developers. The responsibility of the tool developers is to create, enhance or adapt digital tools specifically for the design process, while they are not involved in the actual design of the project. An example illustrating the office composition shown in above Figure 2. 4 could involve the collaboration of experts from different disciplines, including architecture, engineering, and mathematics. For instance, during the collaborative work process of the Great Court at the British Museum, the experts worked in close connection to develop a tool that helped them to execute the design task. Notably, the tool was ultimately created by a mathematician as a direct outcome of their collaborative efforts in the design process (Foster et al., 2001). The form of the Great Court roof developed from the consequences of design requirements that providing a transition from the circular form of a central reading room to the square form of the surrounding museum buildings (Szalapaj, 2005). To resolve the irregular geometry of the roof, a form generating computer program was developed in close collaboration between designers and programmers and other disciplines (Foster et al., 2001). The complex system of the form required the coordination and resolution of nodes between the inner circular part and the outer rectangular boundaries. By parameterizing the design problems and utilizing advanced engineering modeling techniques, it became possible to effectively design, analyze, and construct high-level complex roof forms. These approach allowed for a systematic and efficient process of addressing the intricacies and challenges associated with achieving a cohesive and structurally sound design for the Great Court roof at the British Museum (Szalapaj, 2005). This example shows that collaboration between remote domains could improve and actualize extraordinary design ideas. The distribution of knowledge between architects and the other domains resulted in a new tool development.

In architectural design, creative ideas often emerge through an iterative process. One of the most renowned examples of collaboration in architectural design is the partnership between architect Louis I. Kahn and structural engineer August E. Komendant during the design and construction of the Kimbell Art Museum. Their collaboration is notable for the clear distribution of roles and responsibilities between them (Donchin, 2013). Kahn and Komendant worked in synchrony, and their design process was iterative due to the intersection of their knowledge domains rather than being remote or completely overlapping (as shown in Figure 2. 5). The design project evolved through a continuous back-and-forth interaction between the architect and the engineer. As an architect, Kahn was driven to create and develop the design concept. On the other hand, Komendant, as an engineer, focused on resolving the structural challenges of the design. While Komendant preferred minimal alterations, Kahn, known for his perfectionism, often repeated many steps in the design process. Kahn's approach involved constantly updating his ideas and seeking input from various individuals around him. He valued new perspectives and sought to incorporate fresh approaches into his design. This open-mindedness and collaboration enriched the design process of the Kimbell Art Museum project (Donchin, 2013). Therefore, Kahn’s design was based on the act of reshaping again and again.

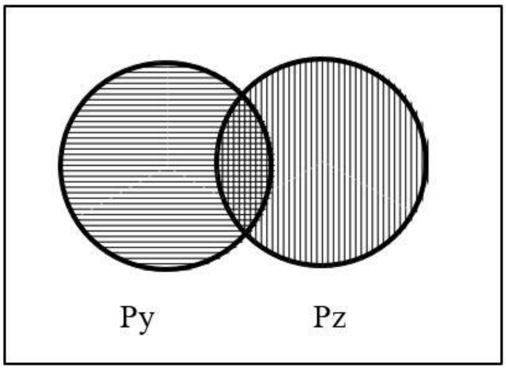


Figure 2. 5. Schematic representation of intersected knowledge domains.
(Source: The illustrations are drawn by the author.)

Design problems are ill-defined (Eastman, 1969) and needs reinterpration many times in the design process. On the other hand, the engineer Komendant was always ready for solutions in his mind, he was meticulous about the calculations (Donchin, 2013). When presented with a design problem, Komendant, as the structural engineer, would focus on solving the structural challenges. He would meticulously revise, recalculate, and redraw to ensure that the design met the necessary structural requirements. While Kahn approached the design process in a sophisticated manner, Komendant's approach was grounded in precise calculations and minimizing deviations. Komendant preferred to avoid significant changes to the design concept, as his expertise lay in finding multiple solutions for any structural problem. In the collaborative context between the engineer and the architect, both experts needed to work synchronously to achieve the design goals of the Kimbell Art Museum and other projects. While their roles were not clearly delineated or strictly hierarchical, they were able to successfully collaborate and create a cohesive design. In Figure 2. 5, the domains represented by Py and Pz (such as architecture and engineering) are assumed to be different but not remote from each other. They intersect due to a shared goal, but they are not completely overlapped like Hutchins' concept of overlapped knowledge distribution. This collaboration example demonstrates that redundancy is not applicable to the entire task, as it is not a hierarchical system. Instead, the collaboration relies on the integration and coordination of expertise from both domains to achieve a successful project outcome.

Computational algorithms have proven to be highly valuable in facilitating collaboration among multiple participants in the design process, particularly when dealing with complex design problems (Besserud et al., 2013). Algorithmic tools are widely recognized for their ability to expedite the process of solving design problems, particularly by enabling the search for a wide range of solutions and facilitating data visualization to foster collaboration among participants (Olsen & Namara, 2014).

Expertise in different domains resulted in the use of different representational systems in digital technologies. Nowadays, many architectural design firms built their digital design tools in the design process. Design firms, they even have a group of people that produce unique digital design tools for every single project. For instance, in Herzog & de Meuron architectural offices, the group of Digital Technology (DT) works with the design team from the initial design idea of a project to the end of the building construction (Strehlke, 2009). But DT group is not in charge of architectural design, the group serves to facilitate architecture (Strehlke, 2009), while the design team is not to build digital design tools. Another example is UNStudio, they have an open-source system that is called Knowledge Platforms grouped under four specific titles (Sustainability, Organization, Materials, Parametric). By organizing the platforms, UNStudio supports an interactive, nonhierarchical relationship within the team. In the UNStudio, the group called Smart Parametric Platform (SPP) manages, maintains, and develops the computational tools and processes involved in the design and construction processes.

To understand creativity within collaborative teams, another significant reference is the work of Keith Sawyer. A study conducted by Sawyer (2017) exploring emergent creativity within collaborative setting, looks into how jazz musicians, with their diverse musical instruments, come together to improvise without any predefined composition, resulting in creative outcomes. Sawyer emphasizes that for creative improvisation to occur, it requires attentive and deep listening, understanding the other person (K. Sawyer, 2017). Sawyer (2010) states that “a desire to understand the individual's creativity while participating in a social event” (p. 15) is a prerequisite for group creativity. Sawyer points out that team members who have previously worked together will be able to handle challenges more smoothly. Furthermore, Sawyer states that having a clearly defined goal will facilitate improvisation and problem-solving for the team (K. Sawyer, 2017).

In the groups Sawyer (2017) examined, he emphasizes the need for team participants to be fully concentrated and focused. This concentration allows them to be focused on problem-solving. According to Sawyer (2017), the level of group flow is enhanced when individuals experience a sense of autonomy, competence, and connection with others and he adds team autonomy is consistently identified as the most significant predictor of team performance. Sawyer (2017) states that group flow is linked to two parameters: autonomy and control. Team participants should feel a sense of autonomy while not perceiving themselves as being under someone else's control. He emphasizes that groups that are flexible, actively listen, and open to innovation are more likely to be creative. Sawyer (2017) suggests that improvisation, and therefore creative ideas, actually emerge more in groups during situations of tight pressure, such as in urgent circumstances. In the nature of architectural practice, there is often a sense of urgency. Project schedules are typically tight, and there is a constant push to meet deadlines. While this urgency is not an actual emergency or a life-or-death situation, it can feel like a matter of utmost importance for teams striving to complete a project. In such situations, improvisation can become inevitable in finding solutions.

2.4. Communication and Representations in Design Teams

Communication is one of the central issue in creating shared understandings in design teams (Perry & Sanderson, 1998). Bucciarelli (1988) sees design as a social process, which is full of uncertainty and ambiguity. According to Bucciarelli (1988) participants in a group have to create a shared understanding in order to be in agreement on the most crucial issues and to decide on what to do next in consensus. However, to deal with these issues, participants need to share their ideas through communication. Bucciarelli (1988) points the difficulties of communication in teams such as different representations, different interests, different knowledge about a task, different responsibilities etc. Bucciarelli (1996) introduces the term ‘object world’ to explain the difficulties of knowledge sharing among team members. An ‘object world’ means, “domain of thought and artifact within which actors in engineering design move and live when working on any specific aspect, instrumental part, subsystem or sub-function of the whole” (Bucciarelli, 1996, p. 62). Moreover, ‘object world’ includes participants’ beliefs, knowledge, interests, and experiences (Bucciarelli, 1996).

Tools are mental or physical devices that help and enhance our cognitive abilities. The conception of cognitive artifacts, introduced by Norman (1991), relate to “those artificial devices that maintain, display, or operate upon information in order to serve a representational function and that affect human cognitive performance” (p. 17). Remarkably, design activities are highly related with many tools or artifacts such as drawings, methods, techniques, instruments etc. Design research has been dedicated to drawings and its importance in design process for a long time (Goldschmidt, 1991). Ideas must be represented to share with others and oneself (Porter & Goldschmidt, 2001). Designing is defined as “the production of a design representation” (Galle, 1999, p. 63). Design representations play a significant role in the design process and product (Galle, 1999). Design representation is identified with two fundamental roles: ‘communication’ and ‘exploration’ (Eck, 2015). There is clear understanding that design representation in teams has a central role in ‘communication’ and ‘exploration’ (Galle, 1999). Participants of a design team need to share their design ideas with others. Design representation is not only for self-communication, it is also for communication with clients, makers, users, and team participants (Eck, 2015).

Representation has been considered as central in any problem solving task (Simon, 1996). Simon (1996) states “solving a problem simply means representing it so as to make the solution transparent” (p. 132). Hutchins (Hutchins, 1995a) describes the roles of representation in the navigation system with the ‘fix cycle’, which means the basic procedures of navigation that are accomplished by a cycle of activity where the representations of the spatial relationship of the ship to known landmarks are created, transformed, and combined so, the solution of the problem become transparent. Hutchins (Hutchins, 1995a) thinks that representations are crucial for navigation because tools and local functional systems which are composed of an interaction between a person and tool. He discusses the computational activity can be distributed not only for partial results, but also through means of computation.

Representations have the potential to structure communication among team participants. Verzijl (1997) points out the importance of communication by stating that "architecture is primarily about communication" (pg. 2). The term 'communication' is defined as "social interaction through messages" (Fiske, 1990, p. 2). There are some obstacles that team participants should get over to communicate in a team devising different tactics. One is playing the devils' advocate role when team participants converged or fixated too early (Nemeth & Nemeth-Brown, 2003). In-depth discussions can encourage individuals to express their ideas in a team work (Stasser & Birchmeier, 2003). Otherwise, individuals can be influenced by the common choice of the team, which is defined as "the common knowledge effect" (Gigone & Hastie, 1993). Another obstacle is the domination of the team by the common information held in the team (Stasser & Birchmeier, 2003). Cheng and Kvan (2000) resolved communication problems in their study by using particular digital methods for sharing design ideas and concepts among team participants. To determine the appropriate technology, Cheng and Kvan (2000) analyzed and classified participants' profiles. At the same time, they took into account the required and aimed results, the tasks of the work and temporal issues (Cheng & Kvan, 2000).

Representations can be internal (in the mind) or external (material and physically perceivable (Goldschmidt & Porter, 2004). Visual imagery is the place for inner representations that designers extract and realize design formations; external representations are meant to dispose of the limitations of inner representation, obligatory because communication depends on representing ideas. For an example of external representational system in distributed cognitive system, Hutchins (1995b) investigate the use of speed bugs in a cockpit system. Speed bugs are physical tabs that are moved according to the airspeed markers to mark critical situations in a flight. However, speed bugs are tools in a cockpit system for pilots to control speed determination for landing the aircraft rather than making calculations. This external representation indicates that tools are also for using perception in a task. The speed bugs, as an external representation tools, provide a perceptual operation as minimizing the efforts in a complex cognitive system (Hutchins, 1995b). The representational structure of the speed bugs also show that external representations provide communication in a common language, such as in the cockpit system (Hutchins, 1995b). The representational systems provides the coordination of actions between individuals and tools in distributed cognitive tasks (Hutchins, 1995b).

Design is based on representations between individuals and tools (Goldschmidt & Porter, 2004). Designers can represent their ideas to facilitate interpretation or they can be represent ideas to themselves to see and interpret their ideas or, designers can reinterpret other designers' designs through representations (Galle, 1999). In design teams, shared representation has an important role. Gabriel and Maher (2000) defines the metaphor of "reflective conversation" (Schön, 1991) in collaborative design, and the reflective conversation takes a form among the designers and a shared representation of the design idea in collaborative environments (Gabriel & Maher, 2000). This 'shared representation' becomes a tool for understanding and cogitating of the shared problems, ideas, and representations (Gabriel & Maher, 2000). According to Gabriel and Maher (2000), sharing representations among team participants occurs through drawings, notes, conversations, and notations which are produced by team participants during design phases (Gabriel & Maher, 2000).

Representations can be detailed or can quickly outline initial design ideas; they can be concrete or abstract; they can be in real scale or not; they may be pictorial or text or symbolic or sketchy; three dimensional or two dimensional on a paper or computer monitor (Goldschmidt & Porter, 2004). Representations such as sketches, models, prototypes, drawings are also devices for creating shared understanding and a base for solving design conflicts because they facilitate the organization of teams' work process and collective concepts (Henderson, 1999). Externalizations of ideas produce a trace of abstract thoughts and this make some difficult tasks in design problems easier (Fischer & Ostwald, 2005).

CHAPTER 3

METHODS OF STUDY

The research adopts ethnographic field observation and interviewing as strategic techniques to capture the knowledge distribution among different participants in the context of computational design practices. It follows the outlines of grounded theory for analytic induction. The chapter provides an overview of ethnographic research, why it is adopted in this study, and its precedents in the architectural research.

Following a brief overview of the methodological approach, the chapter presents the details of data collection procedures, including field observation and interviewing, coding, analysis protocols, and strategies to achieve higher levels of reliability and validity. An extended description of the case and the context is then presented describing the architectural offices, the teams organizations, the projects and the key participants.

3.1. Approach

This research proposes that design tasks are distributed among different groups of individuals, tools, and representations in architectural profession. To thoroughly investigate the distributed cognitive system in architectural design practice, it is necessary to conduct long-term observational studies of the socio-cognitive environment where interdisciplinary interactions take place in authentic settings.

The study employs ethnographic field strategies, including observations and semi-structured interviews, as methods to capture interdisciplinary problem-solving processes in their natural context and to understand groups and people in their everyday professional lives (Emerson et al., 1995). It utilizes a qualitative research method that highlights the significance of processes and meanings (Denzin et al., 2005). Ethnographic study in design teams gives a rich set of data derived from different strategies including interviews, team discussions, incidental conversations, documents as well as non-verbal interactions (Ball & Ormerod, 2000). Ethnography has been engaged with cognitive processes, and it has been highlighted in the context and material environment of the distributed cognitive systems (Hutchins, 1995a). As a process, ethnographic research method involves extended observations of a group, through observing day-to-day lives of the people, and interviews with group participants (Creswell, 2007). In observing design teams, the researcher captures implicit and nonverbalized practices within the practice, which are otherwise unaccessible to the researcher. The team participants' behaviors provide rich data to the researcher who takes personal notes in the field and tries to understand the participants' point of view. Team dynamics create social processes that researcher must must observe and document for a long term in participants' environments. In addition, observations and semi-structured interviews were conducted to investigate interactions in design teams. This study proposes that design process of teams could be viewed as a distributed cognitive system (Hutchins, 1995a) between participants of the teams, tools, and representations. The study specifically inquires design teams' communications, modes of knowledge transfer and representation techniques in the design process.

3.2. Overview of Research Design

In the beginning of the research, two pilot studies were conducted in İstanbul. The first pilot study was conducted to observe students working in teams working in an educational setting as part of a semester course. The processes within the class and the communication within the teams were monitored throughout the semester. The second pilot study involved a one-week observational research conducted at Office A as a preliminary study for the long-term observations in the later stages of the study. The pilot studies were instrumental in the operationalization of the future steps of the research. The selection of offices for long-term inquiry was primarily based on the accebility of the office together with its location, size and structure, scale, complexity, and availability of the projects delivered.

Before the observational study, an observational research protocol was developed (Table 3. 1).

Table 3. 1. The observational research protocol.

(Source: The table is drawn by the author)

	SHORT TERM	LONG TERM
	Office oriented- Office based	Project (Job) Oriented- Team based
TIME	No Project	Part of a project
	One Week	Four Weeks
DATA COLLECTION TOOLS	Structured Interviews/Focused Interviews Surveys Office organization schema Description of tasks	Observations Semi-Structured Interviews Consultant
DATA TYPES	Verbal and visual data, photographs, records, surveys	Field observation notes (thick descriptions), photographs, verbal and visual data, records, photographs, sketches, meeting minutes, job-time schedules, online communication records (e-mails, chatting etc.), digital notes and documents, meeting minutes, algorithmic codes, CAD drawings, plan of the office, organization schema
DATA SOURCE	Office documents, Surveys, interviews	Field notes, Meeting minutes, interviews
ANALYSIS	Descriptive	Interpretative

3.3. Data Collection

3.3.1. Ethnography

Ethnography is a qualitative research approach that allows the collection of in-depth and complex social data (Fischer & Finkelstein, 1991). In an ethnographic research, the researcher focuses on an entire cultural group through extended observations of a group and/or interviews with the group participants (Creswell, 2007). The researcher spends time conversing with and observing the group, or can have a participant observer role (Creswell, 2007). The characteristics of ethnographic study provides a broader perspective of the qualitative strategy among research methods supporting a holistic exploration of a setting, context-rich detailed data, unstructured data, a focus on a single case or small number of cases, and data analysis that emphasizes the interpretation of the meanings and functions of human action (Groat et al., 2013).

Ethnographic research is not only a method of data collection, it is also a style of research, focusing on naturally occurring non-experimental situations to achieve the aims and approaches in a study (Brewer, 2003). Sommer et al. (1997) suggest that before beginning an interview study, it is better to observe subjects' everyday lives first. Ethnographic research includes two situations. The observer either enters into social setting but does not participate in people's everyday lives routine or the observer takes a participant role in the observed social setting (Sommer & Sommer, 1997). For an ethnographic study, it is important to ask "why it is important to describe" what is being researched and to interpret the cultural behavior of a certain group of people or how a group is marginalized and kept silent by others (Creswell, 2007). In ethnography, the questions are related to a description of the context, an analysis of the major themes, and the interpretation of cultural behavior (Creswell, 2007). It means that these research questions are mostly open-ended and focus on understanding the context and identifying the cultural behaviour.

Data collection for an ethnographic study is mostly based on observations. In this research, data was recorded as field-notes, interview and observational protocols. LeCompte et al. (1999) list the forms of data that are acquired in an ethnographic study as casual conversation, key informant (participant) interview, semi-structured and structured interview, questionnaire (written and/or oral), observations (nonparticipant to participant), content analysis of secondary text or visual material, elicitation techniques (e.g., looking at a scrapbook and talking about memories), and audiovisual material (e.g., audio or visual record, such as camera recording).

3.3.1.1. Field Observations

As a data collection method, nonparticipant observation offers possibilities for the researcher being a complete insider rather than a complete outsider (Creswell, 2007). The site protocol involved non-participant observations witihin situated settings. The observations were conducted with teams of architects working on design projects with computational design tools in the design process. Data was recorded as field-notes, video and audio records, and observational protocols. From the design process, all related sketches, models, digital media, photographs, and drawings in digital and/or non-digital were collected.

Observation timeline and schedule

Observational research continued throughout the day based on the teams' daily routines, starting from the beginning of working hours (9 am) in the morning at the office and continuing until the end of the day (6 pm). The observational research was conducted in the offices on weekdays and weekends, from the start of the design project to the delivery stages. The planned or unplanned meetings of the teams, communication between individuals, and spontaneous events were observed and recorded.

The time frame for this study was approximately 37 days at Office A and 23 days at Office B, starting from the beginning the design to the delivery of competition requirements. The time spent at the firms in total amounts to about 550 hours, including about 21 hours of 19 interviews with individuals and audio records in the offices, and 22 hours of teams and consultant meetings. The collected data was in the form of general field note observations, audiotaped and videotaped interviews and audio/video taped group meetings. A total of 18 interviews with individuals and 650 audio/video records with the teams were fully transcribed and analyzed.

Offices and Participants

Two offices were visited for this study. Office A was founded in 2013 in İstanbul. The team (table.3) consisted of one office leader (architect, O_aL), two team leaders (architects, O_aTL_1 , O_aTL_2), and seven interns (intern architects, O_aIA_{1-7}). The team leaders were responsible from monitoring interns' works and coordination between the office leader and the team participants. The team acquired consultancy services from a landscape architect (O_aLA) and a civil engineer (O_aCE). The civil engineer (O_aCE) was always at the office but never took a role in the team as a team participant. O_aCE was also acting as the office manager in charge of the daily running of the office. In the following table (Table 3. 2), the organizational structure of the office is presented.

Table 3. 2. Team participants by dates (*OxLy: Office Leader, OxTLy: Team Leader, OxCy: Coder OaIAy: Intern Architect

(Source: The table is drawn by the author)

Dates	Office Leader	Team Leader	Team Participants	Consultants
31.07.2017 – 14.08.2017	O_aL	O_aTL_1 / O_aC_1	O_aIA_1, O_aIA_2, O_aIA_3 , O_aIA_7	O_aLA, O_aCE
01.08.2017 – 07.09.2017	O_aL	O_aTL_2	O_aIA_1, O_aIA_2, O_aIA_3, O_aIA_4, O_aIA_4, O_aIA_5, O_aIA_6, O_aIA_7	

It is possible to sum up the list of team participants with eight job titles: *(i)* **designer** (design idea developer) *(ii)* **draftsman** (drawing of plans, sections, elevations, system detail drawings), *(iii)* **3D animator** (drawing of digital model), *(iv)* **graphic designer** (illustrating of design idea and analyzes), **engineer** (structural calculations), *(v)* **researcher** (researching sample projects and solutions), *(vi)* **architectural model maker**, *(vii)* **design drafter** (architectural rendering), *(viii)* **leader** (leading team), (ix) **coder** (code developer).

Office B was founded in 1994 and the team consisted of 37 architects. The Office takes a multidisciplinary approach to design and encourages the use of digital technological tools and methods in their projects. The first team observed at this office was structured in a hierarchical order; one office leader (O_bL), one team leader (O_bTL_1), and two architect coders (O_bC_2 , O_bC_3). After a week, the team leader had decided to assign a new team leader. The new team consisted of one office leader (O_bL), one team leader (O_bTL_2), and two coders (O_bC_2 , O_bC_3). In the retrospective research conducted at the same Office, the team consisted of one office leader (O_bL), one team leader (O_bTL_2), and one architect coder (O_bC_1) and one consultant. In the following table (Table 3. 3), the organizational structure of the office is presented.

Table 3. 3. Team participants by dates (*OxLy: Office Leader, OxTLy: Team Leader, OxCy: Coder)

(Source: The table is drawn by the author)

Dates	**Office Leader**	**Team Leader**	**Team Participants**	**Consultants**
23.07- 27.07.2018	O_bL	O_bTL_1	O_bC_2 , O_bC_3	-
27.07-15.08.2018	O_bL	O_bTL_2	O_bC_2 , O_bC_3	-
Retrospective	O_bL	O_bTL_2	O_bC_1	O_bJ

In total there were six job titles in Office B: *(i)* **designer** (design idea developer), *(ii)* **draftsman** (drawing of plans, sections, elevations, system detail drawings), *(iii)* **3D animator** (drawing of digital model), *(iv)* **architectural model maker**, *(v)* **design drafter** (architectural rendering), *(vi)* **leader** (leading team), (*vii*) **coder** (code developer). In the observed teams the roles of the team participants were initially defined in both offices. These roles did not change during the process significantly.

3.3.1.2. Semi-Structured Interviews

Within the scope of the study, semi-structured interviews were conducted with architectural design teams to explore the transfer of knowledge among team participants. The interviews aimed to examine the teams' information processing practices. Additionally, the interviews helped uncover the communication strategies and knowledge representation techniques employed by the teams. The semi-structured interviews provided a lens to understand each participants’ own descriptions of a situation and disclosed the situations and problems that could not otherwise be envisaged.

The semi structured interviews were conducted at the beginning of the observations and at the end of the design process with office leaders, team leaders, and coders. Additionaly, especially with the coders, semi-structured interviews were videotaped while they were working with computational tools. Moreover, the participants were asked to explain the visuals during interviews and team meetings. The explanations introduced by the participants were instrumental in interpreting the reasoning processes and strategies of the designers. The semi-structured interviews were face-to-face to provide a way to explore feelings, opinions and behaviors (Sommer & Sommer, 1997a). Semi-structured interviews made possible to inquire about individuals' ideas about a situation. The coders' interviews especially provided a significant perspective on how they include the digital tools in the design process.

In the retrospective study, semi-structured interviews were conducted with the coder and the office and team leaders.

All the semi-structured interviews were recorded with a voice recorder and some of them with a video recorder. Most of the team meetings were video recorded. Additionally, documents from the design process, such as participants' sketches, notes, digital files, and model photographs, were collected. The interviews were transcribed into digital files as Word documents.

Access to Project Documents and Archives

In addition to interviews and observations, the study utilized archival data stored either through physical means or through computer server data. Accessing the offices' archives was crucial in getting into all the printed documents deemed important to be stored by the offices. **Office B** stored their previous projects in their archives. Access was granted to these documents, providing valuable data, especially for the retrospective research. The displayed models within the office also served as archival material. The collected materials were labeled in the data set.

Accessing the local server systems' files was also crucial in establishing the timelines of the projects. **Office A** was using a data server system to create a digital archive. Through the server system the team was sharing files among them. Maintained on the office's online data server, this management tool allowed participants, including team leaders, intern architects, and coders teams to access all project related documentation, including competitions' design briefs, schedules and drawings.

3.4. Data Analysis and Interpretation

3.4.1. Coding and Analysis

Data analysis was conducted in three phases: description, analysis, and interpretation of culture-sharing group (Creswell, 2007). Wolcott (1994) points that to write a good ethnography is to ‘describe’ the culture-sharing group and setting:

> “Description is the foundation upon which qualitative research is built ... Here you become the storyteller, inviting the reader to see through your eyes what you have seen... Start by presenting a straightforward description of the setting and events. No footnotes, no intrusive analysis, just the facts, carefully presented and interestingly related at an appropriate level of detail.” (Wolcott, 1994: p. 28)

The observations were presented as one set of facts and descriptions in a chronological order and reporting a “day in the project process” of the design teams. Most known analysis procedure is ‘the search for patterned regularities in the data’ (Wolcott, 1994). Moreover, comparing the culture-sharing group to others, evaluating the group in terms of standards, and drawing connections between the culture-sharing group and larger theoretical frameworks are analysis methods in ethnographic study.

In the description phase of data analysis, all the data was indexed in a timeline to understand the design process (Figure 3. 1). Collected data were inscribed on the timeline as sketch, photograph, field notes, meeting minutes, video records, audio records, screenshots, and e-mails. The timeline also included information about the tools used during the design process.

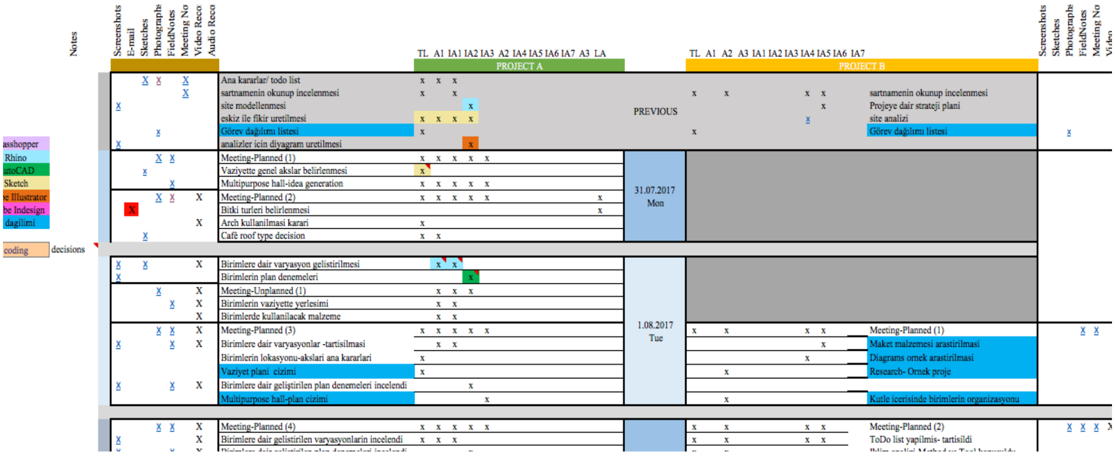


Figure 3. 1. A screenshot of the data coding of the observational research. (Source: The screenshot is taken by the author.)

MAXQDA software was used to organize and manage the qualitative data, (Figure 3. 2). One can utilize the software to analyze the transcribed data. The software has three sections which are Document System where the transcript data files can be listed, Code System in which codes can be created, listed, and be connected to each other. Code System allows to organize the data in a hierarchical structure, to take notes about a code, and to create links to memos and external files such as images or texts. Finally, Document Browser offers a window in which the researcher reaches the data to work on. After all the data organized in the timeline, all video and audio recordings of the meetings and the interviews were transcribed for coding purposes in the MAXQDA software (Figure 3. 2).

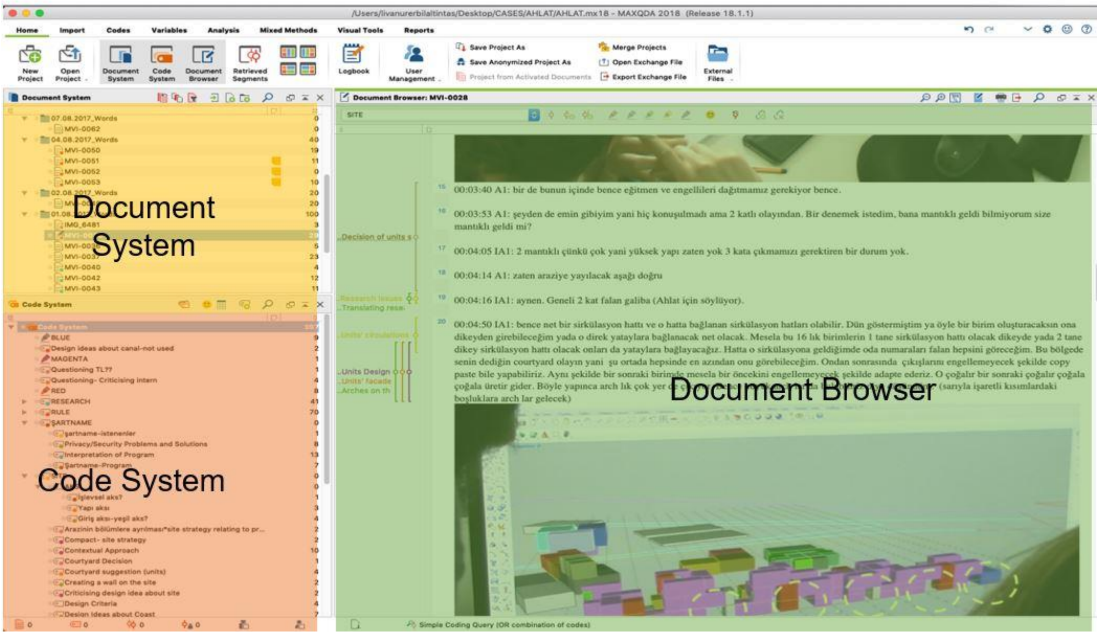


Figure 3. 2. A screenshot of the MAXQDA software showing the open coding of the observational research.

(Source: The screenshot is taken by the author.)

In grounded theory, one adopts an open coding procedure for developing categories of information (Strauss & Corbin, 1990). In the open coding phase, the text is examined, and emergent categories are identified by the researcher.

Categories that are listed according to selected phenomena, are interconnected as *axial coding* to create categories. *Axial coding*, building upon pre-established categories, refers to a series of procedures that reorganize data in novel ways after the initial phase of open coding. It involves establishing connections between categories to develop a more comprehensive understanding of the data (Strauss & Corbin, 1990).

Creating relationships between the categories and building a ‘story’ which connects categories is called *selective coding* (Strauss & Corbin, 1990). This stage is the final step in the coding process, during which substantive themes and a theory are developed based on the core categories identified. Selective coding is the process of selecting the core category, which acts as the central phenomenon that brings together all other categories. It involves validating the connections between categories and addressing any shortcomings in terms of properties and dimensions within those categories. This phase plays a crucial role in synthesizing the data and developing a coherent understanding of the overall research findings (Strauss & Corbin, 1990).

3.4.2. Codes and Categories

Coding Guide. The guide provides definitions for the set of 38 categories developed through the analysis procedures described above. The guide also uses segments from the qualitative data to exemplify each category.

In this section, the 15 super-ordinate categories with their sub-categories are listed:

1) *Design Tools* with sub-categories: making the design/form open for further manipulation, interaction through digital tool;
2) *Issues of Form Finding in the Design Process* with sub-categories: form finding through rule-based strategies, introducing sophistication in design, motivation on extraordinary/complicated design form, experimentation in form finding, manual interferences in the design process, capacity of tool control, unexpected discovery of form, prioritizing structural design over form finding, motivation to integrate computational methods, employing a formula in form finding, aesthetic or stylistic formal preferences;
3) *Intuition in Computational Design* with sub-categories: intuitive manipulations of digital models, form finding practices;
4) *Time in Design Process* with sub-categories: advantages of computational tool, schedule constraints in design decision, adopting a particular design method;
5) *Making The Design Process Transparent* with sub-categories: making design moves transparent;
6) *Representational Practices* with sub-categories: using multiple design representations, comparison of representational systems;
7) *Digital Collaboration Mechanisms* with sub-categories: having an archive at one’s disposal, coordination through tools;
8) *Design Conceptualization* with sub-categories: having general design approach;
9) *Collaborative Practices in Design Process* with sub-categories: coordination strategies in collaborative work, shared approaches and intentions within teams, interdisciplinary collaboration, engaging external parties in design process, client engagements;
10) *Precedents* with sub-categories: sources of inspirations, reference to precedents, contextualizing design ideas;

11) *Budget Issues* with sub-categories: prioritizing design idea over budget constraints, material aspects of design;
12) *Linking Parameters with Each Other* with sub-categories: introducing programmatic concerns in design;
13) *Design Approach* with sub-categories: introducing legibility;
14) *Research in Design* with sub-categories: conducting research;
15) *Design Explorations* with sub-categories: Structured and unstructured explorations in design.

3.4.3. Validity

There are several strategies to achieve higher levels of reliability and validity with qualitative methods. This research followed two strategies, namely inter-rater reliability and triangulation. To ensure the quality and rigor of qualitative research, an assessment known as inter-rater reliability evaluation is employed. This evaluation involves two or more independent coders who adhere to coding and interpretation methods, working together to establish a consensus or agreement (Creswell, 2003). The inter-rater reliability protocol was initiated after the initial categories and super-ordinate categories were generated. A coding guidebook was generated (APPENDIX C: Coding Guide) to include categories, descriptions, and samples from existing transcripts. Then, a researcher with qualitative data analysis experience was assigned a sample of transcripts to run the analysis following the same coding protocol. The coding guide included instructions for the rater to facilitate following the protocol.

Inter-rater reliability analysis was conducted for four transcripts from different segments of the video recording totaling 59 minutes long (22+1+34+2 min.; 4 recordings). The coder, then, participated in the sessions to discuss and negotiate the categories that emerged from the analyzed data set. A desired level of concurrence between the coders, based on initial and negotiated codes, was not achieved in the first meeting. However, a second meeting was conducted to discuss the codes at the end of which there was 80% concurrence. Given the complexity and scope of the material, 80% level of concurrence was considered sufficient. There were no additional meetings regarding inter-rater reliability, and the research continued with the analysis process using the agreed-upon categories from the inter-rater reliability analysis.

Triangulation is another strategy to achieve higher levels of validity. Triangulation is a procedure in qualitative research where researchers pursue convergence among multiple sources of information. Although there are many types of triangulation employed in qualitative analysis, this study employed two of them: methods triangulation and triangulation of data sources (Hammersley & Atkinson, 2003). The utilization of methods triangulation ensured the coherence of the study's findings by incorporating various data collection procedures such as interviews and field observations. To enhance the reliability and validity of the qualitative methods employed, data sources were triangulated, allowing for a comparison and cross-checking of information obtained through different means and at different points in time. Data sources were also triangulated to compare and cross-check. Obtaining data through both observation and interviews provided triangulation. In addition, field notes, collected sketches, screenshots, and photographs also played a role in the triangulation (Hammersley & Atkinson, 2003).

Realism asserts the existence of a world that is independent of the observer. From an epistemological standpoint, it asserts that we have the ability to acquire knowledge about this world that exists independently (Wilson & Keil, 1999). The research aimed to reach a transferrable interpretations in line with the observations. As long as the interpretations are realistic, the concepts may be transferrable. Therefore, the concepts may be controversial in other studies. In ethnography generalization is not the ultimate goal. Rather, realistic “thick descriptions” from which one might formulate transferrable concepts is the main concern.

3.5. Description of the Cases

3.5.1. Office A

Office A is an architectural design office founded in 2013, aimed at developing the relational thinking capacities of architectural design and its relation with design technologies. The team operates within a hierarchical structure (Figure 3. 3), and apart from the office leader, the other team participants varies based on the scope of projects. In addition to architecture students who come to the office for internships, there are three experienced architects who work there permanently.

The office spaces consist of two meeting rooms (one being the main hall), one work area/studio, one archive, one model making studio, one server and print room, and additional facilities (Figure 3. 4). The office leader's work area is separated from the team's workroom, situated in the main hall, which is also used as a welcoming area and a meeting room for visitors. The other participants' work area is situated at another part of the office, connected to the team leader's work area with a long corridor.

In the model making room, there is one 3D printer; in the server and printer room, there is one desktop and one printer, and in the studio, there are desktop computers for the team. The office leader has her own computer in the large meeting room. However, most of the interns were not using the desktop computers; they preferred to use their own laptops, which caused some problems with updating their working files unto the server system. In the main hall, which was also the office leader's workspace, there was a large model of the current project under construction. Additionally, there were a couple of pieces of furniture designed by the office. The building where the office was located was a historic apartment building with a typical Turkish apartment layout. Through the long corridor connecting the main living area, used as the reception room and the office leader’s work room, was separated from the section where the team worked. In the meeting room next to the studio where the team worked, frequent meetings took place, including the office leader.

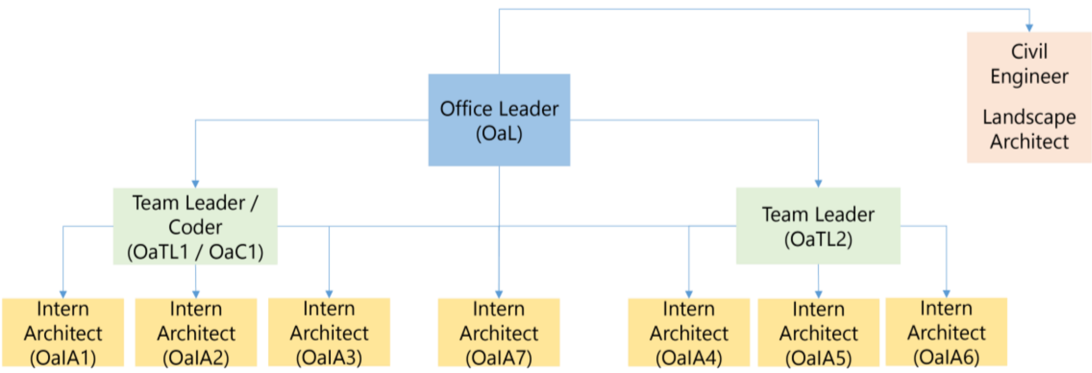


Figure 3. 3. Organization schema of Office A.

(Source: The diagram is drawn by the author.)

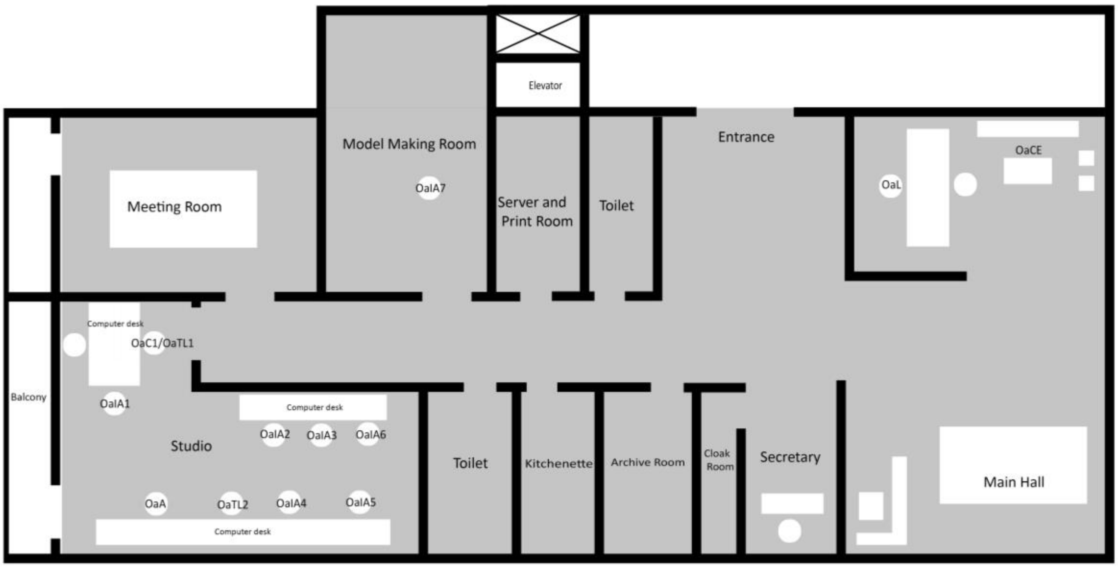


Figure 3. 4. Office plan layout of Office A.

(Source: The diagram is drawn by the author.)

3.5.1.1. Case 1: Ahlat Youth Center

The first case studied in this office was the Ahlat Youth Center competition project. The subject of the competition was the design of a youth center in a town located by the Van Lake. The architectural program of the project, with 5852 m^2 enclosed area, included 62 accommodation units, a spa, a conference center, technical volume.

3.5.1.2. Case 2: Süleymanpaşa Municipality Building

The second observed case was also a competition project. The subject of the competition was the design of a municipality building in a city center. The program of the project, totaling an approxametly 11.0000 m^2 enclosed area, included offices, conference hall, and restaurant, meeting halls, assembly hall, presidency and its private rooms.

3.5.2. Office B

Office B was founded in 1994 and at the time of the study there were 37 architects working at the office. The office takes a multidisciplinary approach and employs new technological tools and methods in their projects. The team operates within a hierarchical structure (Figure 3. 5 and 3. 6). In the office there were four core teams and each of them focused on different projects and additionaly a render team was supporting the core teams.

The office was laid out on two floors connected to each other via a gallery space. By utilizing the spatial advantage of establishing visual and auditory communication between floors through the gallery, it was possible to communicate from the ground floor to the mezzanine floor by addressing various issues through verbal and visual means. There is constant activity in the office, with people coming and going all the time, and it is possible to see books, models, and materials everywhere. In fact, even the steps leading to the mezzanine floor have books on them. The teams were separated from each other not by walls but by open shelves, books, and models. At the entrance of the office, there is a model made for WAF (World Architecture Festival) with numerous models on it. There are three stands at the elevator entrance, one with a bust and the others with again architectural models. On the ground floor, there are many models on and under the tables in the corridor area.

Upon entering, immediately to the left, there is the office leader's working desk and opposite it is the meeting room. Continuing underneath the stairs leading to the upper floor, there is the model making lab where the 3D printer and materials are also located. On the mezzanine floor, there are studios connected to an open corridor. Each studio team has arranged their offices according to their own preferences with small modifications. The fire stairs directly connected to the office were accessible during breaks.

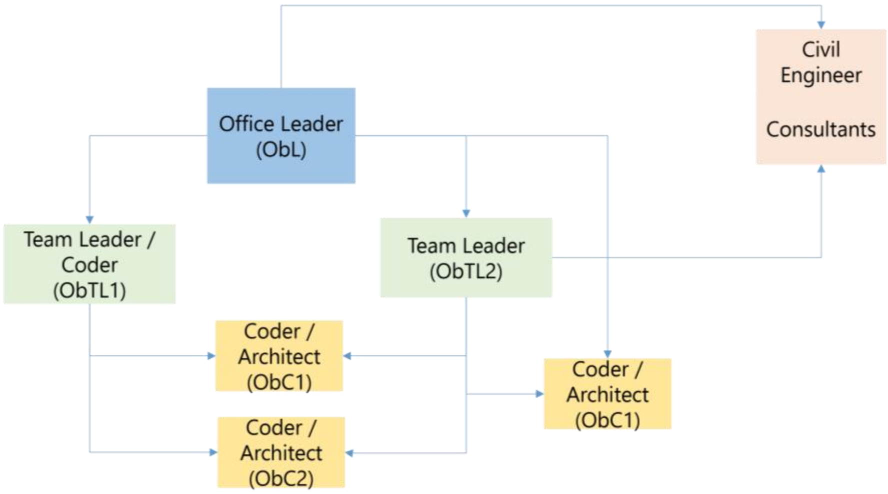


Figure 3. 5. Organization schema of Office B.

(Source: The diagram is drawn by the author.)

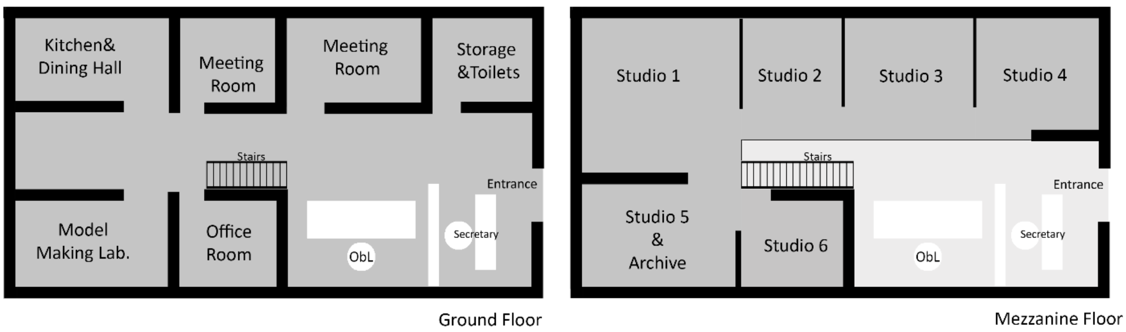


Figure 3. 6. Office plan layout of Office B.

(Source: The diagram is drawn by the author.)

3.5.2.1. Case 3: Etiler Towers Project

The first studied case at this office was an invited competition project for which the costumer was expecting an extraordinary looking architectural form. The location of the project site was İstanbul. The project consisted of two high-rise housing towers and one hotel tower together with a shopping center and restaurant facilities at the ground floor. The floor layout and the overall design was already completed by a team at the office, while a second team was in charge of the façade design.

3.5.2.2. Case 4: İTU Library Design Project

The final studied case was a library project prepared for the İstanbul Technical University. The program of the project included a public library together with an archive and reading rooms. Moreover, the library site was on the campus and an iconic design was expected by the university. In total approxametly the enclosed area of the project was 33.000 m^2.

CHAPTER 4

SERENDIPITOUS EXPLORATIONS IN DISTRIBUTED WORK IN COMPUTATIONAL DESIGN

“In the fields of observation chance favors only the prepared mind.”
Louis Pasteur

This chapter provides an in-depth account of how architectural design teams manage design development processes within a distributed cognitive system and how they handle serendipity opportunistically to generate multiple alternatives in an exploratory setting. The focus is on the specific context of design exploration that employs computational approaches to advance the design work at hand.

The chapter aims at describing unexpected and serendipitous situations that arise within the collaborative environment. The situated observations are framed and introduced around three forms of explorations: (1) Exploration by manipulating code (using specific code prescriptions), (2) Exploration by code (experimenting with forms through algorithms), and (3) Exploration with code (experimenting with forms using rule-based strategies). The first type of exploration involves making formal manipulations using software and its predetermined code, while the second type entails conducting form-related experiments through algorithms. In the third type, designers use rule-based strategies to create and articulate the formal composition of the design. In each sub-section, specific events are discussed in each type of exploration to describe the segments of serendipitous explorations. These events highlight several issues, such as setting up exploratory design tools, visualizing explored design ideas, using a variety of design tools, and distributing the design process among multiple tools and participants.

As described above, these three events largely involve form-finding experiments in the design process. In these explorative processes, it was observed that designers were in search for moments of satisfaction with the formal composition at various scales. These moments were traditionally called as "a-ha" moments, where designers had sudden insights by recognizing a satisfactory possible design solution (Akin & Akin, 1996). The qualitative data analysis suggested different types of explorations that signify the relevancies of serendipity. Within the analysis and coding, “Unexpected Discovery of Form” appeared to be the most noticeable in the coding categories. At large, it includes subcategories related to the mechanisms and the participants of a distributed cognitive system with respect to coincidental moves in the design process. Categories that highlight problems associated with “Unexpected Discovery of Form” such as “Form Finding Practices” and “Form Finding Through Rule-Based Strategies” illustrate generation of form coincidently. The narratives introduced in this chapter are based on these particular set of qualitative codes emerging from the situated observations and interviews conducted with participants.

This chapter explores three distinct approaches to generating multiple alternatives while solving design problems. Each approach involves different tools and techniques. The first approach is exploration by ***manipulating code***. This method involves using ready-made code in computational design tools like Wolfram Mathematica to create multiple design alternatives. To visualize the results, designers typically use a separate 3D visualization software. Unlike other computational design methods, in Mathematica, the coder first works with a formula and, accordingly, generates 2D graphics which is linked to the formula. If the designer decides to work on a 2D graphic, the next move, then, is to transfer the 2D lines to 3D representations within the modelling software. The second approach is ***exploration by code***. In this method, designers themselves create a script for a particular design problem using computational design tools like Grasshopper. The result of the scripting practice can be viewed and assessed within the modelling software (Rhinoceros). The third approach is ***exploration with code***. This method involves using hand sketches and digital drawing tools like Rhino or AutoCAD to specify algorithmic rules on the design solution. In thşs practice, the designers were observed to draw graphics on paper or other mediums instead of working with numerical codes on a computational design tool.

All of these approaches involve explorative acts that generate unexpected solutions to design problems. During an exploration process in design, designers both get surprised and discover a new form accidentally. These a-ha moments can change the direction of the exploration process in design. In the observed teams, the process leading up to any a-ha moment does not progress systematically or regularly. Following sections presents the three different episodes from the observations.

4.1 Exploration by Manipulating Code

4.1.1 Process

The episode in this section involves the work of Office B. It was a library project within the campus of a major university campus in Istanbul, proceeded through different levels of collaborations gradually. The office leader (O_bL) and the coder (O_bC_1) were the individuals to initiate the work. Then, the initial team with the addition of the team leader (O_bTL_2) carried out the project in the subsequent phases. At the same time, the core team (O_bL and O_bC_1) was in communication with the assigned committee which acted as the client representative compromising the university rector, the architecture faculty members, and experts. After the approval process of the design idea, the core team expanded with the addition of draftsmen, a 3-D visualizer, and engineers reaching a total of 11 participants. The project progressed in three steps: first, explorations as initial sketches and in digital tools to find a form; second, applying the selected form to further steps such as structural organizations in digital tools; third, technical and construction drawings.

Initially, the office leader (O_bL) and the coder (O_bC_1) worked closely during the early phases of design. As O_bC_1 describes, the core team worked in front of the same computer screen while O_bC_1 was controlling the computational tool. These moments of collaboration were key for the team to observe and assess the developing morphology of the design proposal. The team was able to come up with three different alternatives to convince the college administration (Figure 4. 1). Each alternative was about exploring the formal qualities of the facility, rather than fulfilling the functional program. O_bL's intention was to push for the third option (Figure 4. 1, A3), which displayed unusual formal qualities with paraboloid surfaces. The formal idea and the explorations associated with it had been on the agenda of the office for a while even before the library project, and the library project was seen as an opportunity to further explore and implement the idea in an architectural project.

A1: Grid System A2: Waffle Structure A3: Arcade System

Figure 4. 1. The alternatives of the library project. (A: Alternative)
(Source: Office B)

While presenting the alternatives in an interview with the researcher, the office leader (O_bL) clearly distinguished and emphasized the initial alternative with the grid:

00:11:42 O_bL: "Well, of course, everyone goes with grid systems-something better to comprehend-... For instance, this alternative [showing the grid alternative] is something that can be done in terms of both exterior and interior layout, it is not something that cannot be constructed. But we said that when we get such an opportunity, let's move on with something that can be taken a little further, which will not be analyzed or perceived immediately."

O_bL, expressing his thoughts on the conventional grid-based plan generation, relates the popularity of grid-based design with its ease of perception and construction. The team at Office A was motivated by the challenge to solve the design problem with a more complex geometrical system. Two main aims regarding the use of the computational tool to respond to the challenge were identified in the interviews with participants. One is the ability to produce "unusual" forms via a digital tool, and the second is the ability to observe instant changes in the form while altering the algorithmic codes.

O_bL, the office leader, defined creating complex and sophisticated forms as a "further" step. According to him, architectural forms of the future may morph from grid-based forms into organic ones. Furthermore, according to O_bL, if a form is complex enough, it is considered to be at a more advanced level. The motivation to design complex forms led to the use of an advanced software, i.e., Mathematica, which is almost a required tool for scientific research.

In the design process, to ensure close collaboration, O_bL and O_bC_1 provided a setting for exploration and experimentation in the early stages of design. Three alternatives were created, allowing O_bL to push for their preferred scheme, which required the utilization of specific tools in the design process. The arcade system was favored by O_bL due to its extraordinary and stylistic aspects. To achieve the desired qualities, a continuous formal exploration was required from the beginning. To generate the formal schema for the preferred alternative, the team used Wolfram Mathematica software, a sophisticated computing system that solves problems in various domains, including neural networks, geometry, and visualization. The team considered this computing system crucial to generating the series of paraboloid forms. Considering the capacity of the software and the intentions of the design team, the search space was narrowed down to a certain family of forms that characterizes the overall morphology of the library. However, the exploration space was still vast, allowing opportunities for various paths of serendipity in the design. The coder in this project was an architect who worked with coding languages and had knowledge of Wolfram Mathematica and Rhino software, which were asynchronously employed in the process. The Mathematica software provided the possibility to create paraboloid surfaces by processing a formula and facilitated a search space for designers by means of a series of parameters. The Mathematica software was used as a particular design tool that provided instant visualizations of geometries as participants manipulated the formula (Hyperbolic Paraboloid). The reason for developing three different alternatives was to impress the customer with the capacity and spectrum of solutions that the design office can produce in a given time. Also, having three alternatives allowed the team leader to strategically push for their preferred scheme for this particular project. O_bC1 expressed how important Mathematica and the articulation of the paraboloid forms in shaping the design.

00:03:07 O_bC_1: "first we started like this. We've been doing this for years. It's something the office has been working on. We have always worked on this formula. How can we converted it into a building? Even structurally. So, the library project is the same."

00:03:25 R: "so why were you trying this? You will get something like this" [*O_bC1 is interrupting*]

00:03:28 O_bC_1: "This is something that has been accumulated over 14 years, this is a culture of our office. You already have seen everything around in the office. It is something that we [*the office*] had been working on for 14 years."

In the interview, the coder (O_bC1) points out that the experimentations with the form led to an accumulation of knowledge that is already reflected in the office space. The accumulation of knowledge about the form is observable all the time in the office, as there are models, sketches, and other materials. Spending more than a decade searching for a form has been called a 'culture' by O_bC1. Mathematica software provided the possibility to create paraboloid surfaces by processing a formula and facilitated a search space for designers by means of a series of parameters. Furthermore, it provided instant visualizations of geometries as participants adapted and manipulated a formula.

4.1.2 Explorations

The Office B has been working on paraboloid forms for a long time. The explorations via Mathematica software as a tool have been applied as a waffle structure in an installation project named Serra Gate in 2003 (Figure 4. 2). The office leader (O_bL) mentioned in an interview that the Serra Gate project was experimental for the structural system of the form (Walker, 2014). The project has been produced in its original size. Thus, the team had experience with the form that was created through the formula in a real environment. Whether the form was applied as a waffle structure or concrete surfaces, the geometry was obtained using the same methods.

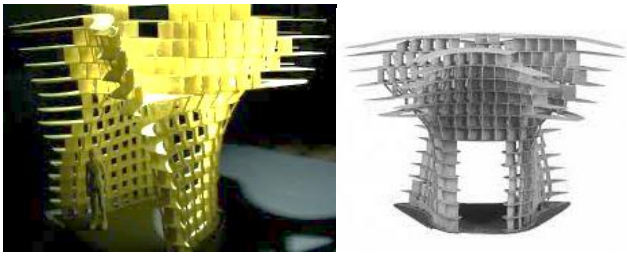


Figure 4. 2. The experimentation for the structural system for the computed form, Serra Gate project by the office B, 2003.

(Source: Office B)

Another research project was carried out in 2017 as a house project with concrete surfaces (Figure 4. 3). The house project was a conceptual design where the design team was able to apply the formula in a small-scale project.

Figure 4. 3. The previous trial with Wolfram formula, a house project by the Office B.

(Source: Office B)

Throughout these experimental projects, the team aimed at experiencing the formula and the possible form variations in two ways: experiencing the formula in the design process and experiencing the form that was produced via the formula as a space. Wolfram Mathematica software providesthe ability to articulate complex geometriesthrough various mathematical tools. The team supported the exploration process by working on the software for a long period and printing the alternatives in 3D models. Thus, the alternative forms became accessible to desingers visually and tactile.

The design team worked with Mathematica in three stages: in Stage 1, the designer adapted the ready-made formula to visualize the idea; in Stage 2, the designer manipulated the parameters within the formula while observing the 2D geometric modifications; in Stage 3, the team identified the potential geometric shapes among the set of available ones and transferred the set into Rhino software for further manipulation based on visualizations. The following screenshot shows an instance from Stage 1 in which the coder modifies the formula and observes the visualization accordingly (Figure 4. 4).

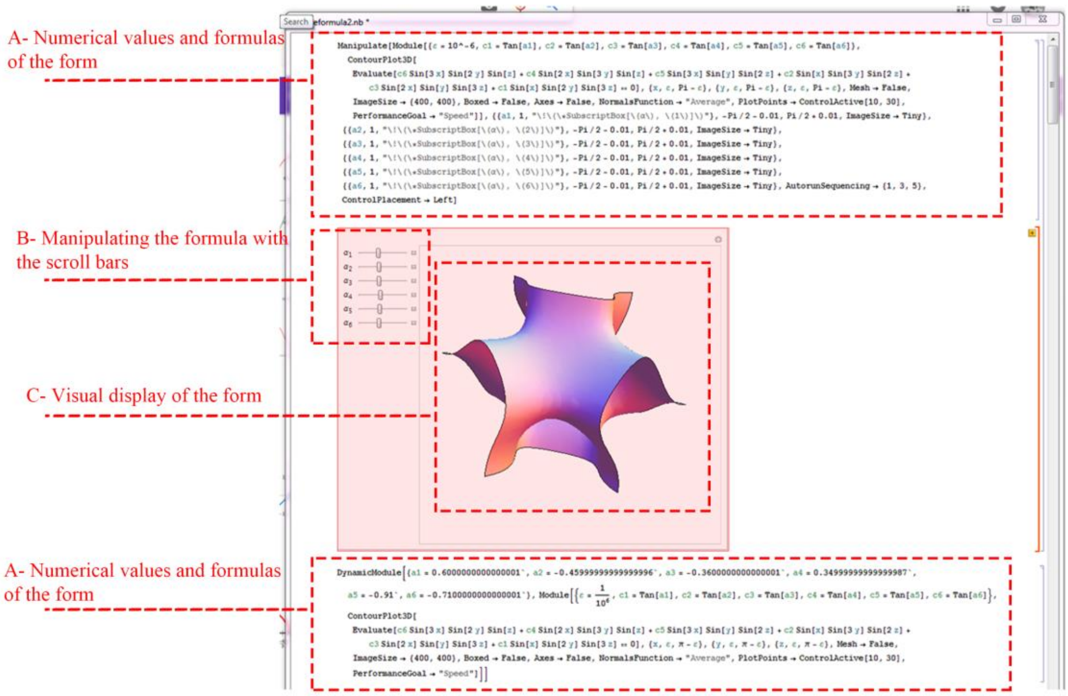


Figure 4. 4. A screen shot of Wolfram Mathematica software interface from the library project design phases.

(Source: Office B)

Zooming in to C-Visual display of the form quadrant of the digital screen on Figure 4. 5, as O_bC_1 moved the slider, and the team were able to observe the transformation of the form in 3D visually. Apart from manipulating the formula in section A on Figure 4. 5, O_bC_1 could move the slider more intuitively.

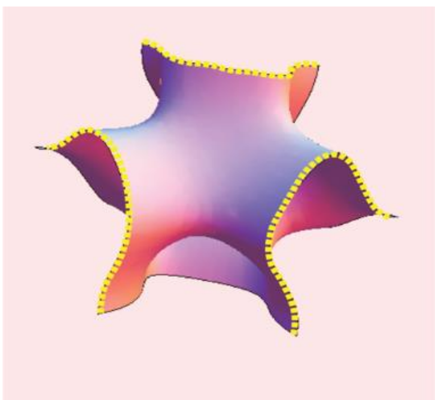


Figure 4. 5. Yellow dots show the lines that are created simultaneously by the formula changes.(yellow lines added by the author.)

(Source: Office B)

The formula instantly creates curvilinear continuous lines (Figure 4. 5), and when the coder changes the parameters within the code, the lines reshape, and the system generates and adapts the surfaces between them. While searching for potential formal solutions, O_bL and O_bC1 filtered the observed alternatives on the screen with specific, but implicit, parameters in mind. These parameters could be derived from their knowledge domains, project description and instructions, or other possible domains that could serve as criteria for the project. O_bC1 described the evaluation parameters for the project as:

00:03:46 R: "okay, you have tried on that, there was a form that you are looking for, what were the criteria in the form that you were searching?"

00:03:53 O_bC_1: "First the structural solid-void, the structure, what the form reminds us, these were very important in form finding. For example, the 16-meter axle's core… were criteria, each of them."

O_bC_1 primarily reflects her concerns about the project's structural issues in her explanations. O_bC_1 intentionally creates situations to advance control over serendipitious explorations as she provides an initial assessment of structural validity based on what is seen on screen. An interesting exploration that may not have the capacity to be built but still it can be presented to the engineers to have evaluation if it is buildable or not. The instances of serendipity does not solely rely on the interplay involving the formula and the geometric representation on the screen.The processes of observation and assessment go through specific filters and are subjected to expertise from both within and outside the discipline of architecture, resulting in the evolution of its form. It becomes a part of the socio-material environment and thus contributes to the specialization of the disciplines surrounding it.

In the dialogs above, O_bC_1 mentions the term 'memory' that establishes a relevance with the form finding strategies. The validity of the form is strengthened by its relevance to this 'memory'. O_bC_1 further explains that in the architectural design process, decision-making mechanisms are incorporated, not only based on physical requirements like structural elements, but also on societal data, such as collective memory. By replicating the factors that influence decision-making, the design team effectively expands the exploration space for potential design solutions.

O_bC_1 searches for and observes possible alternatives and saves each and every output for the next step, which involves formal manipulations of the digital model in Rhino software. This gives another opportunity for exploration to the designer while working in Rhino. It is not fully explicit when or how a designer identifies a certain iteration and further develops the formal qualities in Rhino. However, it is possible to claim that the architectural design criteria, involving structural and programmatic issues, are also in play when designers observe and study the set of emerging geometries. The available geometries generated by the Mathematica software were quickly assessed on the spot based on the voids created through paraboloids. The assessment involved opportunities for daylight penetration, qualities of an architectural space and enclosures, and affordances for a structural element which was perceived as anarcade system. The geometries created by the Mathematica software (Figure 4. 6) were assessed, qualified, or eliminated by the coder before bringing the set to the larger team's attention.

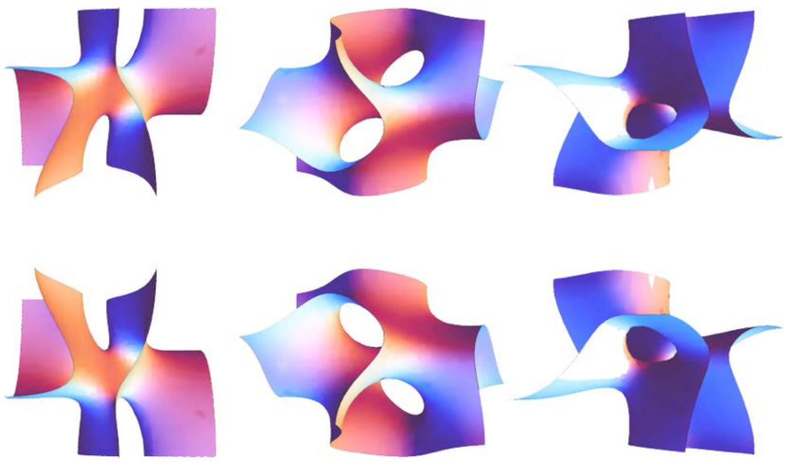


Figure 4. 6. Alternative forms that are created in Mathematica.
(Source: Office B)

Once an alternative form was created in Mathematica, O_bC_1 imported the selected forms into Rhino to place them on a structural system and arrange the form. After obtaining a 3D form in the modelling software, O_bC_1 explored the outer shell of the building by creating floors at different levels (as shown in Figure 4. 7 and 4. 8).

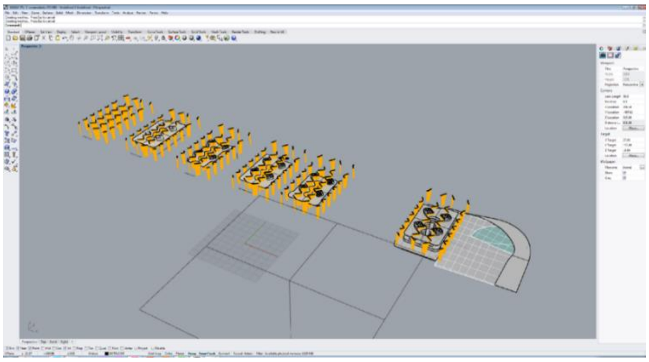


Figure 4. 7. Structural system applications to the geometric form on Rhino as 3D view.
(Source: Office B)

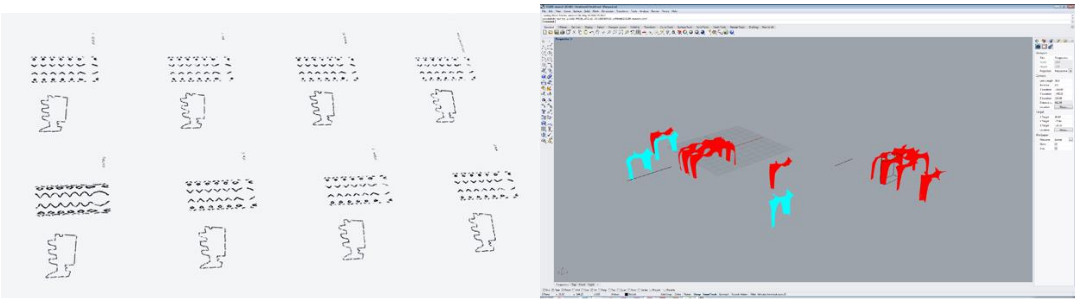


Figure 4. 8 The structural system designs of the form in Rhino. (left) plan drawings of the shelter and the structural system in different levels, (right) adapting the form in Rhino with possible structure elements.
(Source: Office B)

The design team created structural system designs for the form in Rhino, as shown in the plan drawings of the shelter and the structural system at different levels on the left in Figure 4. 8. On the right, O_bC_1 adapts the form in Rhino by incorporating possible structural elements. In the semi-structured interviews, O_bC_1 explained how she adapted the mathematica model in Rhino:

00:04:40 R: "So, you obtained this form, and you mentioned that you transferred it to Rhino. Did the form change there? If so, how did it change? Or was it transferred as it was?"
00:04:42 O_bC_1: "Of course, for example, openings are created, axes are adjusted according to meters. We can't have a 40-meter axis span; it should come with certain increments like 16 meters, so we do it accordingly. We created window openings, placed them inside, and adjusted them. Then, based on architectural decisions, we reformulated it."

O_bC_1 created a more intuitive and unexpected exploration area by manually manipulating the form she obtained in Mathematica. In the semi-structured interviews, the office leader expressed his intention to use Mathematica. According to his words, the form that was achieved is not possible to imagine by sketching.

00:01:15 O_bL: “Here it is! that's it! [pointing at the sketch in Figure 4. 9] This ceiling is poetic but it's useless, it's very poetic, you know, it's not 2D. This is 3D. This is also not 2D but 3D. After we make it like this, we can get it in the 3D printer. You've seen this one’s achieved…. Now that's unimaginable. It's not something that will happen with the human mind. We can imagine, we can think, but it [*software*] develops, reproduces, we use digital technology for this here.”

In the dialogues above, O_bL explained that the form achieved through the software would not have been possible to imagine by sketching alone. O_bL compared sketching tools to computational design software to illustrate how the latter makes it possible to create imaginative forms. According to O_bL, hand sketching has limitations in expressing and developing the ideas in one's mind, making computational tools necessary. However, ObL also mentioned not having the full capability to control the computational software that the team used in the form finding process for the library project. Therefore, O_bL frequently monitored the alternatives generated through Mathematica and their transformations in Rhino. The produced forms were evaluated and developed by the team by being repeatedly represented in different tools.

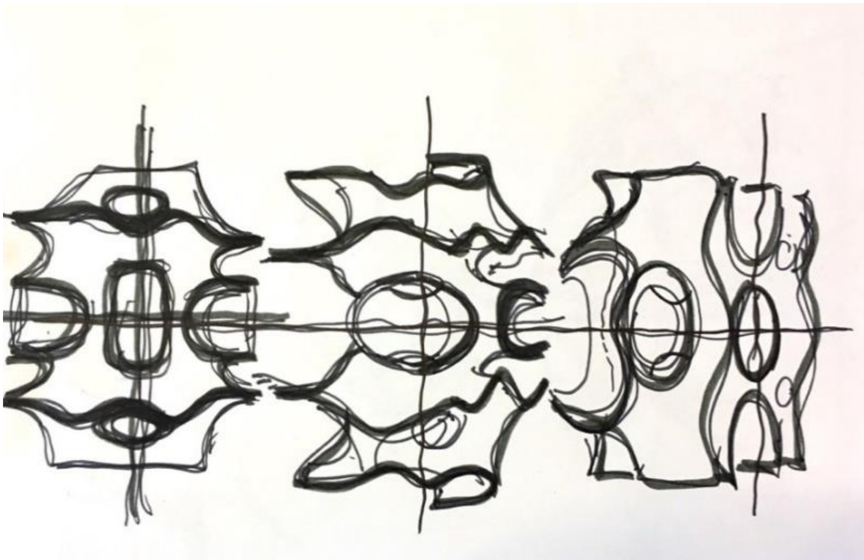


Figure 4. 9. O_bL’s sketch of the library project.

(Source: Office B)

The following sketches represent the exploratory phases of the form in the application of the generated form in the software. At this stage, the design process has evolved into a sketching exercise, which is usually heavily incorporated at the beginning of a design. The team has attempted to adapt the form obtained digitally to the structural system through sketching. In Figure 4. 10, the team searched for possible openings to allow daylight inside by sketching various options.

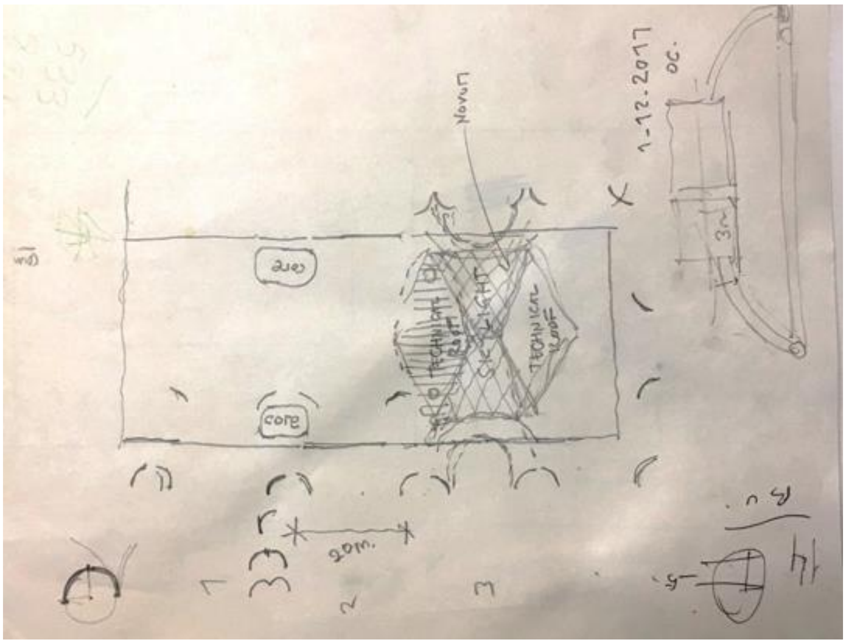


Figure 4. 10. The sketches about the skylight on the roof, developed by the office B’s team.
(Source: Office B)

The hand sketches also helped inform the design of the enterance area while applying the form in Rhino. The outline of the ground floor was sketched on paper to explore possible entrance designs and their forms. Figure 4. 11 shows the exploratory process that the design team carried out on paper through sketches. Before reaching this stage, the team had made the main decisions using Mathematica and Rhino software. They had used digital tools to search for answers to questions such as what the external shell would look like and where and how the structural elements would be.

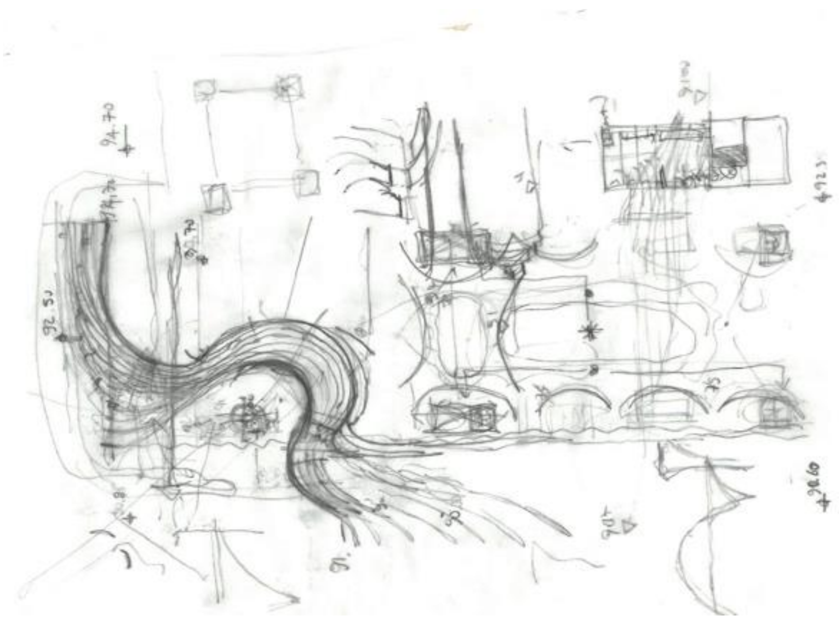


Figure 4. 11. Exploring the entrance according to the exterior shell, developed by the office B’s team.
(Source: Office B)

The team continued their collaborative efforts in exploring the entrance of the building through sketching. While digital tools like Mathematica and Rhino allow for control over the form-exploring process, sketching provides a more practical means of participation. However, O_bL mentined that there were certain points where it was difficult to proceed with hand-drawn sketches, and they had to rely on software tools to move forward.

4.1.3 Serendipitious Exploration in the Library Design

The design team adapted a computational tool intentionally because the team was motivated to try out different alternatives in the numerous options of the computation world. Wolfram Mathematica Software allowed the team participant to search alternatives intuitively while keeping a close control through algorithmic representations embedded and permitted within it.

By using Mathematica and Rhino concurrently and interactively, the team participants had the opportunity to observe the coder's work concurrently with the visualizations of the formulas. The concurrent and interactive use of the software packages further supported the exploration. The design team intentionally adapted a computational tool because they were motivated to conduct experiments and search through a vast range of computation options. Wolfram Mathematica software provided a precise search space to the team, allowing them to control the form with a tendency towards sensory perception.

The team transferred the geometry to Rhino and initiated another series of manipulation to generate a structural unit by means of stretching the edges of the existing geometry. The curvilinear surfaces that were transferred into Rhino Software were further refined through manipulating the geometry on the digital model. The modifications aimed at generating a shell that covers the library to house the architectural program and to develop a structural system for the facility. The team's strategy was to refine a single geometry in isolation, then multiplying the forms by adding, rotating, and mirroring in Rhino. The team preferred to 3D print all potential alternatives to keep track of their exploration space, and keep available alternatives in sight. The intention was to create a tangible archive for the project that they defined as the 'research project'. By generating a 3D printed catalogue, the possibilities of serendipitous exploration were enhanced.

Printing in 3D enabled the team to visualize the alternatives they explored in the software at a fixed scale. Keeping the 3D printed versions of the alternative forms in the exploratory search during the design process allowed for recording of the knowledge and presentation of the ideas in the office. In field observations, the team participants frequently encounter or intentionally examine previously prepared project models and prints in the office during their daily routine. The team constantly saw models, prints, and all projects that were still being worked on but not yet completed. The presence of many things representing knowledge and experience produced in the office sometimes served as an external memory repository and simultaneously a continuing source of inspiration. Considering the products of the work, a sort of archive-office was created.

In Figure 4. 12, the 3D printed models produced in the progress of the form creation are shown together in a time order. Each production was actually a designer's a-ha moment at that moment. In this exploratory process, the design team wanted to document each form they captured as a knowledge of experience, beyond and above just the search for form.

Figure 4. 12. 3D prints of the library design alternatives.
(Source: Office B)

O_bC_1 explained that the first four models in Figure 4. 12 (numbered between1-4) were not produced with Mathematica. After Image 4 (Figure 4. 12), the team decided to continue something more complex:

00:08:20 R: "Are these images in the order that we are looking at now? There seems to be quite a jump from that to this" [*talking about the image numbered 4*].

00:08:21 O_bC_1: "Yes. We said let's do something completely different. This scared us a bit because we are not good at handling these types of solids [*switched to Image 5*]. We asked ourselves if we could achieve something by using these forms in 2D."

In image 6 (Figure 4. 12), the team completely focused on the structure and planned to integrate a separate shell from the interior on top of the library. The team began to form Image 6, which they think balances exactly what they wanted in terms of arches, structural elements, and "*fullness*" which means the density and quantity of structural elements in the library. The team was at a stage where they captured a form with its structure that can be differentiated according to the usage of the space and where they can place amphitheaters and other programs. O_bC_1 explanied that they were aiming arches and an amphi in the design so, they eventually explored a module that they wanted to work on (Figure 4. 12, Image 7). With the aim of designing a portico that would allow to create a “reading” space, they used natural daylight through the opennings of the shell. After creating a module, the team started to work in Rhino. As they moved from Image 7 to Image 8 (Figure 4. 12), their concerns shifted towards replicating the module they had created to fulfill their spatial and structural requirements. Initially, their focus was solely on the form using Mathematica, but with Rhino, *applicability* of the structure became the dominant concern. Until the last visualization Figure 4. 12, image 15), the team worked in Rhino following applicability and the design criteria.

A progressive design process was carried out between Mathematica and Rhino. The potential form considered in Mathematica was reworked in Rhino and brought together with their design criteria. Therefore, this iterative process often allows for explorations and coincidences. When selecting from the forms created in Mathematica, they intuitively selected the potential final form of the shape after modifications in Rhino and made a decision based on that. So, the team made decisions intuitively, along with ambiguity, unforeseen results also emerged. In the trials produced as a 3D print (Figure 4. 12), the process changed course at times (Figure 4. 12, image 4) and turned towards different search paths.

The team used Mathematica to incorporate serendipity into the design process by using the software to generate a large number of possible design solutions based on a set of parameters and constraints. This allows architects to explore a wide range of design possibilities, including unexpected and serendipitous solutions that may not have been initially envisioned. By incorporating a degree of randomness into the generation of these design solutions, architects can also introduce an element of chance that can lead to unexpected explorations. O_bL express his thoughts about computational design and Mathematica as follows:

00:20:14 O_bL: “…I am not a practical user of digital software, but I know the logic behind it and can understand it. With algorithmic and parametric systems, we are trying to achieve something, what is our goal, what can we achieve, why... it is something that emerged out of necessity, a requirement. It does not just appear suddenly…”

O_bL, pointing that computational design is a necessity, talks about a collaboration that not only contributes to the thinking process, but also reveals situations that cannot be thought of without the software. This collaboration between the computational design tool and the designer is also supported by multiple representation systems such as 2D and

3D visualizations and numerical representations in computer screen, and continuity of this representations into an architectural model as 3D printings. As represented in Figure 4. 13, the design team used different representation tools synchronously and asynchronously during the design process.

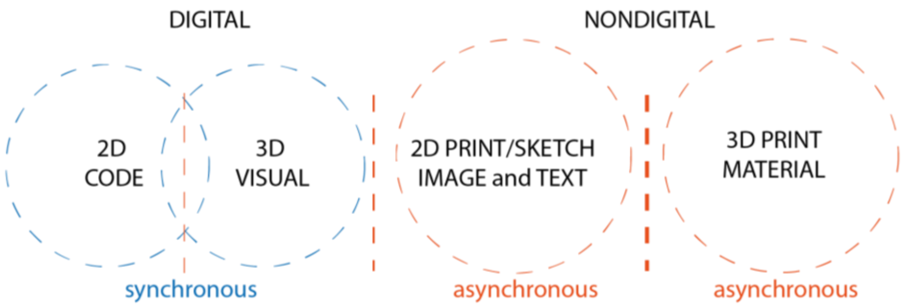


Figure 4. 13. Digital and non-digital representations as synchronized and asynchronized in the library design process.
(Source: The diagrams were drawn by the author.)

Using parametric modeling tools supports designers to have multiple alternatives to be represented in various representational forms. These representational forms act as visual stimuli that are key elements in creative processes (Goldschmidt, 2014), and they can be considered as basis for serendipitous explorations. Expertise is related with number of alternatives a designer produce (Akin, 2001) but, the novice designers who are also coders might create even more with parametric modeling tools than experts. Parametric modeling tools re-form, reproduce, and transform the alternative.

According to O_bL, parametric tools are not only producers of the imagined ideas, the tools might introduce "*unimaginable*" design ideas. The fact that the unimaginable is calculated by the software and presented to the designer is a form of exploration. Computational design has enabled architects to explore new design solutions and to push the boundaries of what is possible in architecture. However, computational design also poses a challenge for architects, as it can lead to a rigid and deterministic design process that does not allow for the kind of unexpected outcomes that can occur through chance encounters. This creates a contradiction between these two situations computing the boundaries of the research space and doing exploration in it. While computation provides a rigid and defined framework, making explorations and progressing intuitively within this framework presents conflicting situations for the designer. When unpredicted results emerge within the defined framework of design parameters, it can lead to discoveries. Alternatively, if the tools being used are not in full control of the designer, the process may progress in total randomness by not having complete mastery of the computation tool, failing to clearly define design boundaries through computation, while keeping the designer in an ambiguous search space.

However, the unpredicted ideas might be situations that should not be imagined, for instance; because of impossible structures. Architects, as a result of their formal education, may be eliminating some ideas without even imagining them by separating buildable and unbuildable designs in their minds. However, the team leader (O_bTL_2) expressed his thoughts about unbuildable structural situations as follows:

00:07:19 O_bTL_2: "yes, the building rotates when you switch the slider. Okay, the building is rotating, but while rotating the building what happens to the structure of it? From this starting point, you designed a thirty meters console on the fiftieth floor. Who will carry this? So, we're going to hang it on clouds? How is it going to be? How will you design the facade when it is rotated? ...You don't think about all these issues because for you, now, the building is slider at that point. … Either you will be very skilled, you need to be able to think faster than computer while doing it, what will you see when you do it, so you will **not leave it to chance**! I think the biggest problem in this parametric design is when it is left to chance… of course they can make very good things out of **coincidences**, but you may not know what will **happen** before you **guess**. Sure, but... the results that come out **without any predictions**, I think more than half of it fails. Because it starts to go not in line with the principles you set at the beginning. It runs away and starts to deviate from the road." [*Emphases are added by the author*.]

O_bTL_2 emphasizes that the lack of control and prevalence of negative results in computational design tools often lead to wasted time and effort. According to O_bTL_2, it is necessary to be able to immediately understand the feasibility of the form seen on the computer screen in reality. However, most coders are young and inexperienced architects, and they are not able to monitor what is computationally produced.

On the other hand, O_bL stated that parametric design tools allowed them to imagine the unimaginable ideas. According to ObL, starting to use computational tools a long time ago can enable thinking about much more complex systems:

00:21:03 O_bL: "... **but it's related to thinking and producing with this software**. They have entered into more smart, more complex systems since childhood..." [*Emphases are added by the author*.]

Getting used to using computational design tools and thinking in the way required by the tool is an advantage for O_bL, but the earlier one starts learning computational way of thinking, the more effective it becomes to design complex forms and systems.

As O_bC_1 stated at the beginning of the design process, they considered certain structural aspects as criteria.The criteria for developing algorithms mostly involve situations where serendipity is taken under control. The structural solution's rationality is preliminarily considered through the algorithm at the beginning of the design idea creation. Serendipitous encounters of potential alternatives are not accepted as they are brought to the fore by the computational tool. They are passed through certain filters and subjected to purposeful evaluations by architects within and outside the discipline. The explored form evolves and is sculpted, becoming part of the socio-material environment of the office. The serendipitously explored form becomes a reflection of the convergence of purposes, approaches, and socio-material configurations.

The nature of sketching involves a reflective action (Schön, 1992) for the designer during the drawing process. In the interviews, the coder and team leader mentioned that they worked directly on the software without producing many sketches initially. The team explored multiple alternatives in a random way as a form of digital exploration. Seeing the form that emerges with each change of the slider on the Mathematica screen and playing with the slider again is a kind of digital reflective action.

The team improvises actively between spatial, structural, and conceptual issues as they work on Mathematica and Rhino. Serendipitous explorations arise during these improvisations. O_bL does not proceed with an already decided form but takes an experimental approach, with no preconceived outcome in mind, thereby making room for serendipity. As they conduct repeated experiments, explorations increase, and the chance for serendipity also increases. When novel computational tools such as Mathematica is integrated into architectural design, a new creative process emerges, where two different domains overlap. Computation-based and design-based approaches are combined, and sometimes insight may come into play in areas where designers have no complete control. In such cases, Mathematica may not be fully under control, allowing unexpected design solutions to be explored by the professionals in situated contexts.

Mathematica is a powerful computational software system that is also adopted by architects for its ability to generate complex geometries and perform advanced numerical computations. Design is an iterative and exploratory process that architects proceed sometimes intuitively and/or through improvisation, while computational tools may not go hand-in-hand with this intuitive process. Generating a large number of possible design solutions is attractive for architects. This allows architects to explore a wide range of design possibilities, including unexpected and serendipitous solutions that may not have been initially envisioned. By incorporating a degree of randomness into the generation of these design solutions, architects can also introduce an element of chance that can lead to unexpected explorations and discoveries. However, the limits and diversity of a computational tool will still be within the boundaries defined by the architect. In the case being presented in this seciton, the team opted to make decisions based on visuals rather than proceeding through code when they were not fully proficient in Mathematica.

In Mathematica, the team created the form by pulling the slider and made random movements to explore 3D visuals. However, to understand the usefulness of these explorations, they needed to follow these steps: 1) a professional should observe the process of manipulation at all times to opportunistically benefit from possible alternatives, 2) further developing some of these alternatives through software packages such as Rhino considering structural and other hard constraints, and 3) discuss and develop a structural qualities with an engineer afterwards. Therefore, the forms obtained at first, evolve during the process as they progress and constantly face new situations and interventions. Throughout all of these changes and steps, there are often back and forths. Computational tools are powerful software systems that can be used to generate complex geometries and perform advanced numerical computations in architecture.

4.2. Exploration by Code

4.2.1. Process

In this section, the design of tower structures for a mixed-use complex is presented. The episode involves the team in Office B. The observations of the design processes were carried out by two different team leaders (O_bTL_1 and O_bTL_2) at different periods. This section is focused on the design process of the second team leader (O_bTL_2). The tower project had previously been analyzed in the office, but it was revisited because it was requested to be reconsidered in terms of the façade system and overall form. The team started to search for a new formal composition while trying to remain loyal to the plan solutions of the previous design. As a first step, the team leader (O_bTL_2) generated sketches about the f acade orientation of the towers (Figure 4. 14). Right after the sketches, the team made a physical model of the site (Figure 4. 15).

The aim of making the site model was to try and see the opportunities of the site. After the physical model was completed, the coder (O_bC_3) shot a photograph of the site model and traced the visibility axes on a CAD software (Figure 4. 16).

The following figures represent the design process from the sketching phases to the digital tools.

Figure 4. 14. (left) O_bTL_2's sketches about the view of the towers. (right) The purple lines show the view angles in the site.
(Source: The photographs are taken by the author. Place: Office B)

Figure 4. 15. (left) O_bTL_2 and the coders working on the site model in the model laboratory. (right) New site model; generated by the O_bTL_2.
(Source: The photographs are taken by the author. Place: Office B)

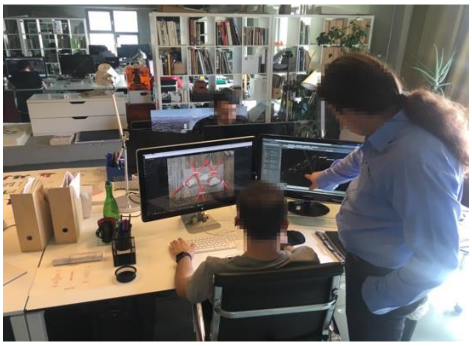
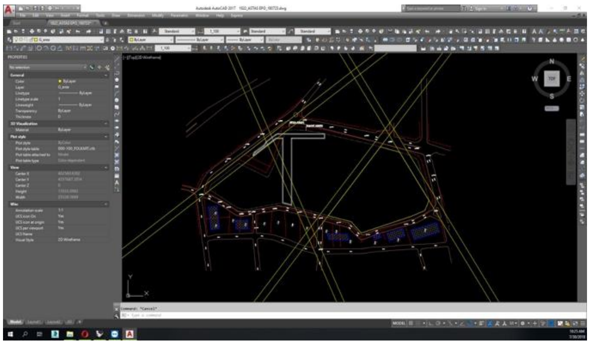
 

Figure 4. 16. (left) The coder (O_bC_2) is working on the site model photographs in CAD software. (right) CAD file of the O_bC_2, showing the view axes which are directly drawn from the model photograph.

(Source: The photographs are taken by the author. Place: Office B)

Upon the decision of the office leader (O_bL), the new team leader (O_bTL_2) reviewed the previous process and then sent the coders (O_bC_2 and O_bC_3) to the project site to gain firsthand experience. Afterwards, the coders could integrate their observations about the site and the parameters specified by O_bTL_2 into the design.

4.2.2. Explorations

The design process of the towers was carried out using Grasshopper and Rhino simultaneously. Both coders (O_bC_2 and O_bC_3) were experienced in both tools. In the first stage, O_bC_3 started to develop a generic code to guide the formal features of the tower. In the second stage, O_bC_3 adapted and applied that generic code to different levels of the towers, which were constantly changing from floor to floor.

O_bC_3 related the significant steps while working on the generic code. First, the team aimed at ensuring control over a hypothetical geometry composed of circles. They wanted to place the elements that would form the facade on the circles, and that frame the openings for the on the façade system. The hypothetical geometry, the circles, was on the horizontal plane, and they were creating the generic code that could allow to be transformed into other desired geometries. In the observations, O_bC_2 followed four steps: (i) creating a section of the towers; (ii) creating geometric elements on the section of the towers; (iii) adjusting the rotation of the elements according to the computer mouse scroll movement; (iv) adjusting the superimpositioin of the elements relative to each other.

Firstly, for step (i), O_bC_3 created a section of the towers that focused on the relationship between six floors of the towers. O_bC_3 created two main circles as the base and the top and evenly divided the space between them. He was trying to develop a vertical surface between the base and top circle lines (Figure 4. 17).

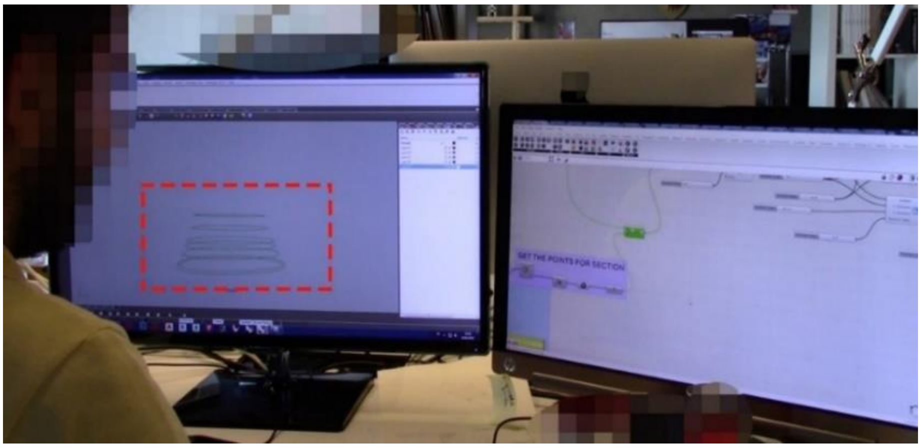


Figure 4. 17. O_bC_3 is creating a generic code (left) Rhino screen is presenting the circle lines (right) Grasshopper screen is presenting components that O_bC_3 created.
(Source: The photographs are taken by the author. Place: Office B)

In the second step (ii), O_bC_3, first identified points with equal intervals on circles (Figure 4. 18). Then, he placed rectangular prisms on these points and observed the movement control of the rectangular prisms when he changed their rota tions (Figure 4. 19).

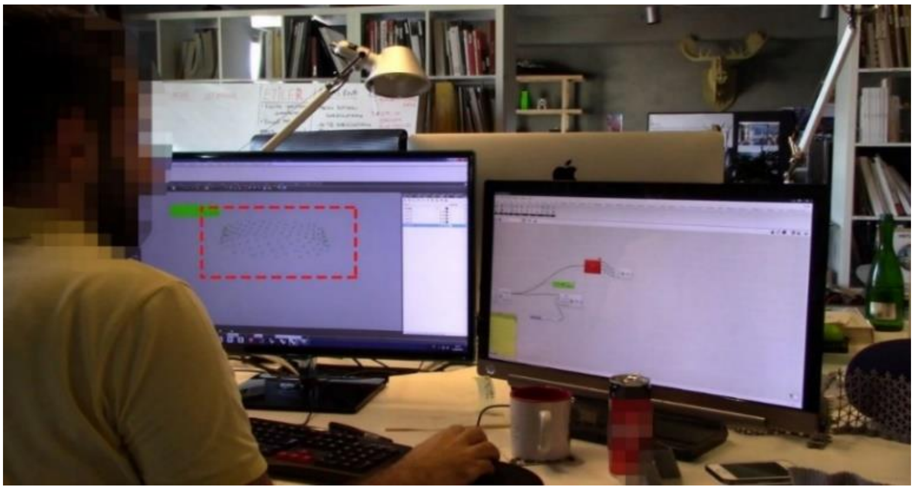


Figure 4. 18. O_bC_3 is creating a generic code: (left) Rhino screen is presenting the points at circle lines (right) Grasshopper screen is presenting components that O_bC_3 created. (Source: The photographs are taken by the author. Place: Office B)

While working in Grasshopper, O_bC_3 was also thinking out loud and describing the problem he was experiencing:

00:02:17 O_bC_3: "hmm, because the expectation of the command we call Orient is actually geometry, meaning its inputs, one is the geometry, the other is the source, and last the target, where we will place it. Here, in the source, you are specifying any base point, such as the center point of a brick, and the source as well. We match it with where that vector location turns. I don't understand why the vector always gives the same direction... how am I going to do this? I have encountered this problem before. For example, these frames here, each one rotates... I sliced them before. But they are dividing the contour because they are on a certain number of curves, so each direction's different... maybe if I convert them to curves, it might work... but I need to go back to the beginning. Hmm, yes, it will be like what I did here. For example, I put a 50-distance brick and I will do the same thing again. Why didn't it cut properly... there's something strange here, what did I multiply it by... it didn't give me what I wanted. It always gives the same sequence, very strange."

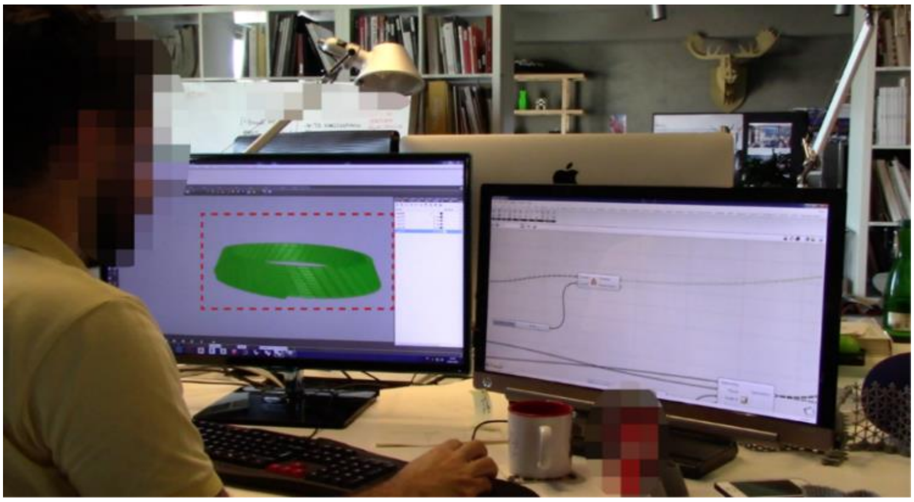


Figure 4. 19. O_bC_3 is creating a generic code: (left) Rhino screen is presenting the circle lines (right) Grasshopper screen is presenting components that O_bC_3 created. (Source: The photographs are taken by the author. Place: Office B)

O_bC_3 described the towers façade they were working on as "mediocre", with regard to the design qualities (Figure 4. 19). To overcome this, O_bC_3 followed two additional strategies, (iii) adjusting the rotation of the rectangular prisms, and (iv) adjusting the superimposition of the rectangular prisms. The surface was made up of rectangular prisms which are thought as panels and/or volumes on the towers. O_bC_3 was trying to control rotation motion of the rectangular prisms while the slider was moving on Grasshopper. With this adjustment, anyone in the team could observe the movement of the rectangular prisms as precisely as possible. To do the control adjustment, O_bC_3 enters randomly selected numbers into the slider in Grasshopper plug-in, Graph Mapper (Figure 4. 20). Thus, O_bC_3 regulates the rotation motion via computer mouse scroll button. In the beginning the scroll was moving between two points A to B, hence, only two different positions of the objects were observable in the 3D view. After adjusting the mouse scroll, O_bC_3 added more points allowing the object to be positioned in different alternative locations along the circle, between A and B, points such as C, D, E, F, etc. When rotating between 0 and 90 degree points with a single movement, only two different positions were possible, whereas having five different positions creates more alternatives (Figure 4.21). Therefore, the more precise and controllable the sliding is, the more different positions the rectangular prisms can be in. Based on the adjustments of the tools,

O_bC_3 had an opportunity to observe alternative positions of the rectangular prisms on the towers. Thus, O_bC_3 had expanded the exploration space for the façade design.

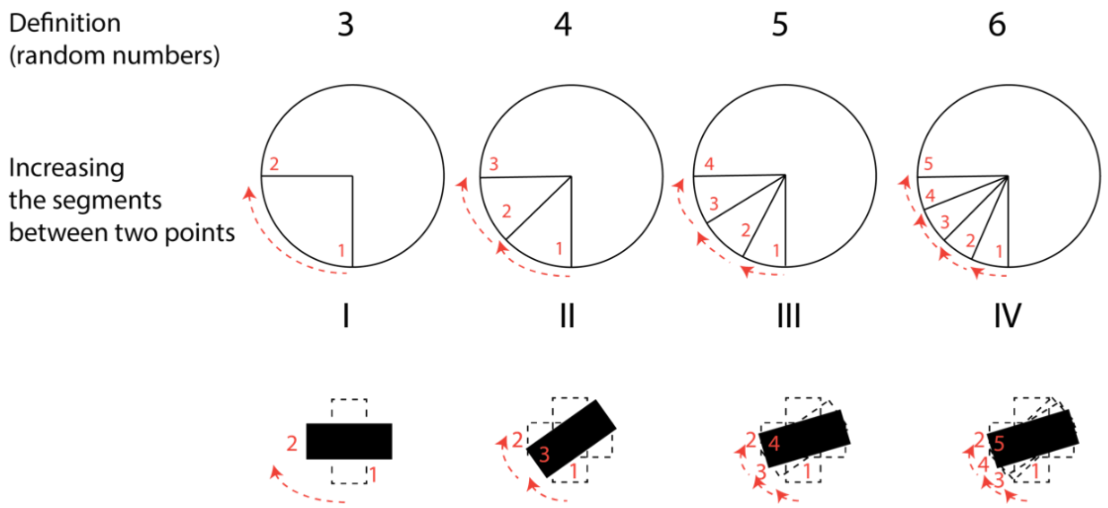


Figure 4. 20. O_bC_3 made adjustments to the scroll speed of the computer mouse. Step I involves rotating the rectangular prisms 90 degrees when the mouse is moved once. Step IV involves rotating the rectangular prisms 90/5 degrees when the mouse is moved once.

(Source: The diagrams were drawn by the author.)

When rotating between 0 and 90 degree points with a single movement, only two different positions were possible, whereas having five different positions creates more alternatives (Figure 4. 21). Therefore, the more precise and controllable the sliding is, the more different positions the rectangular prisms can be in. Based on the adjustments of the tools, O_bC_3 had an opportunity to observe alternative positions of the rectangular prisms on the towers. Thus, O_bC_3 had expanded the exploration area of the façade.

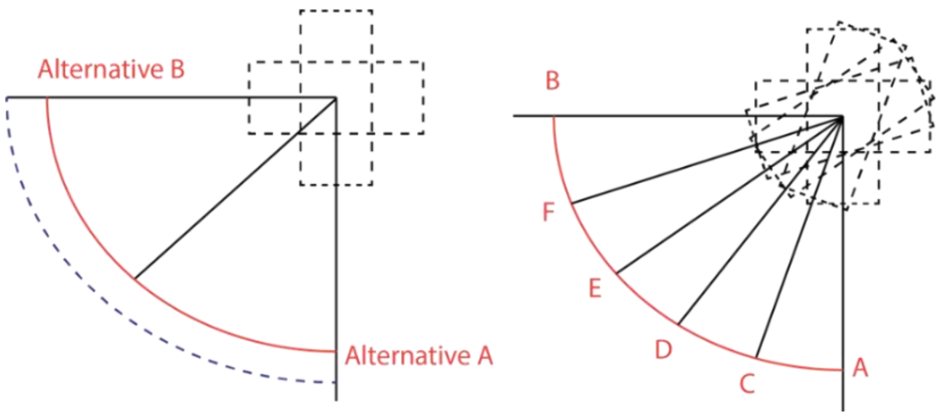


Figure 4. 21. Rotating the rectangular prisms while adjusting the tool.
(Source: The diagrams were drawn by the author.)

In addition to rotating the rectangular prisms, O_bC_3 also wanted to perform a 'stagger' sliding motion. "Stagger" is a command that O_bC_3 wanted to implement in order to prevent the rectangles above the vertical circles from aligning with the top and bottom rows of the prisms (Figure 4. 22). This would involve sliding the rectangular prisms to the right or left so that they were not aligned with the top and bottom rows, apart from rotating around their centers (Figure 4. 22 and 4. 23). As a result, the second coder (O_bC_2) who works with generic code would be able to create different levels of solid and void on the facade as he or she move the slider.

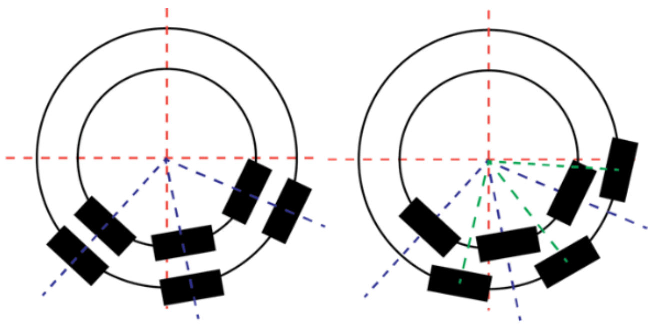


Figure 4. 22. Plan view of the circules with rectangular prisms while adjusting the tool with the comment ‘stagger’.
(Source: The diagrams were drawn by the author.)

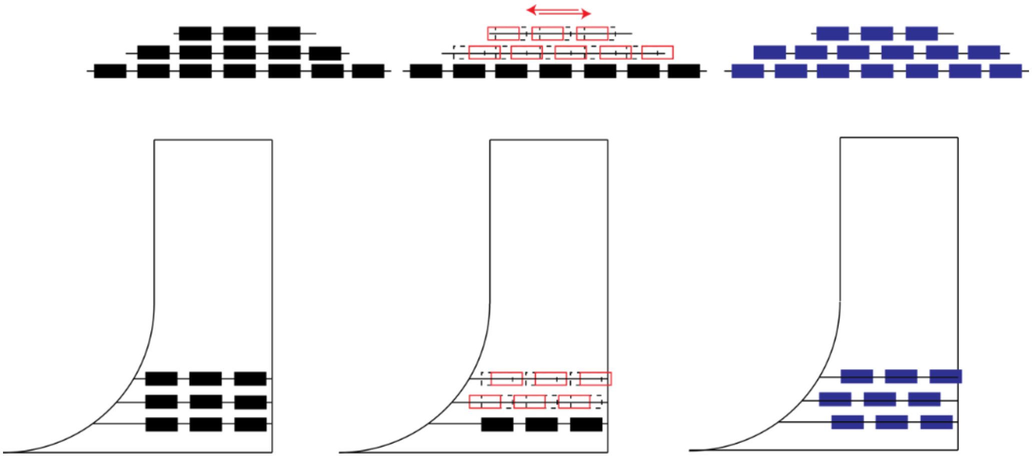


Figure 4. 23. Sections of the circules with rectangular prisms while adjusting the tool with the comment ‘stagger’.
(Source: The diagrams were drawn by the author.)

O_bC_2 and O_bC_3 routinely explored online resources to review newly developed plug-ins, and were downloading and saving those that could potentially benefit their project or be useful in the future. Occasionally, the coders were incorporating a newly developed plug-in into their design, or conduct research to find a specific solution. Sometimes the coders were advancing their design based on a plug-in they came across, while other times they were researching solutions they wanted to achieve in their minds. O_bC_3 was conducting experiments by associating other components with the Graph Mapper plug-in. O_bC_3 frequently examined other works published on online platforms. Based on these focused-research, O_bC_3 was experimenting with the ideas he gained and the solutions he found. O_bC_3 learned that he could obtain more alternatives by applying Range component on Graph Mapper and he started working on it. O_bC_3's strategy was to create the surface with objects that are arrayed in an order. To control the array, Grasshopper's plug-in[1] Graph Mapper[2] (Figure 4. 24) and Range[3] were used to generate the surfaces of the façade design. Graph Mapper is one of the plug-ins that is developed for Grasshopper to enable control of multiple objects' rotations following a value range. Graph Mapper plug-in is a function that can decrease output numbers proportionally and also modifies and cut-offs distance data according to the graph shape and range (Davidson, 2014).

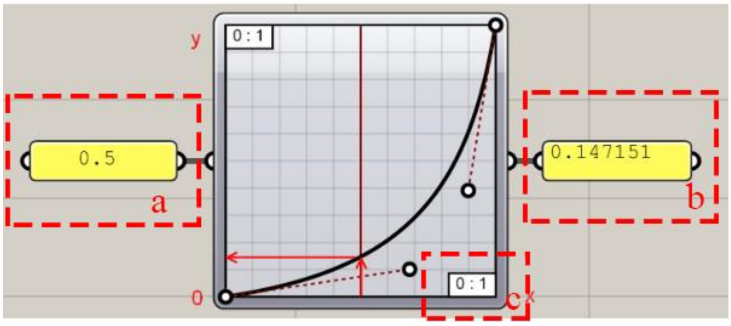


Figure 4. 24. A screenshot of Graph Mapper component a: number slider input; b: number slider output; c: is Range that the coder entered.
(Source: The screenshot is taken by the author)

Graph Mapper starts to work as when a value is inputted for the X axis value [right corner down]. Then, the value will be mapped onto the Y axis [left corner up] by using the curve. The curve is controlled by the numeric values of the sliders[4]. Graph mapper

[1] **Plug-in,** also called add-on or extension, computer software that adds new functions to a host program without altering the host program itself. Grasshopper plug-ins are mostly developed by the users of the software and shared online. Anybody who is interested with the software can develop and upload a plug-in, or download via online searching. Plug-ins are kind of digital band-aid for the software where users could not solve the problem directly from the software. When a user faced with a problem while designing, the solutions might be shared already on an online web pages such as food4rhino.com etc.
[2] **Graph Mapper:** First input in Grasshopper. After calling the graph mapper, the coder makes connections between Graph Mapper and other plug-ins to control output. It is defined as: "the Graph Mapper is a two-dimensional interface with which we can modify numerical values by plotting the input along the Graph's X Axis and outputting the corresponding value along the Y Axis at the X value intersection of the Graph. It is extremely useful for modulating a set of values within an institutive, grip-based interface." (Payne, 2015, p. 49).
[3] **Range:** is defined as: "a range creates a list of evenly spaced numbers between a low and a high value called the domain. A Range component divides a numeric domain into even segments and returns a list of values" (Payne, 2015, p. 225).
[4] **Number Slider:** is defined as: "a slider is a special interface object that allows for quick setting of individual numeric values. It makes possible to change the values and properties through the menu, or by double-clicking a slider object" (Payne, 2015, p. 225).

can also be controlled through other components (Figure 4. 25) that effect the 3D image of the design.

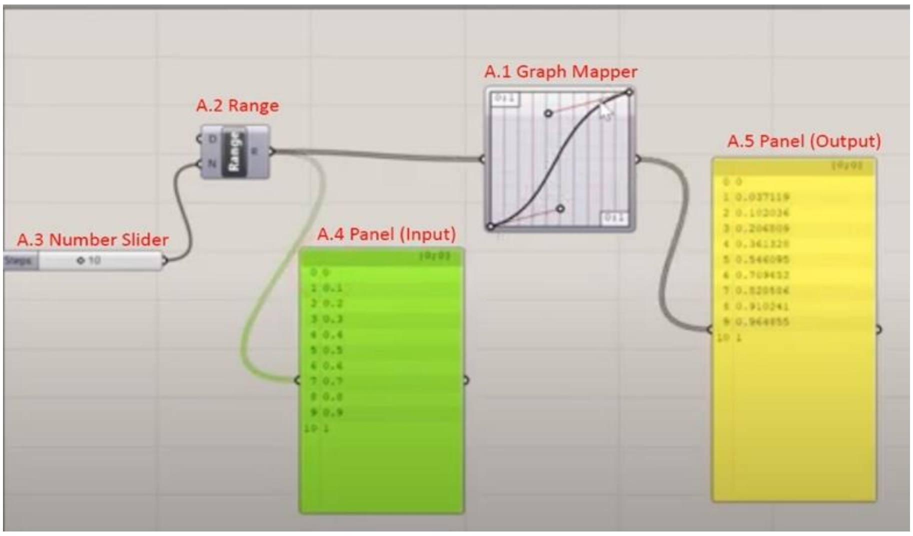


Figure 4. 25. A screenshot for Graph Mapper instillation on Grasshopper.
(Source: The screenshot is taken by the author)

Throughout these trials and experiments, $O_bC_3$experimented with the Graph Mapper plug-in by following tutorial videos and using plug-ins as a guide. As a result, O_bC_3 was able to develop a method for creating a generic code in Grasshopper to design patterns.

00:14:29 O_bC_3: "Otherwise, this works, we can extract something from it with just one... we can create it with just one pattern. Actually, it does the same thing... they did it for only one of them, then we add one more to it.
00:15:24 R: If you give more than two...
00:15:39 O_bC_3: Hmm... I added one more. I did something I don't understand.
00:16:58 O_bC_3: Maybe that's why... haha, look, it worked now. It's about dividing something into different numbers, I divided it into even numbers, for example, even..." [*Interrupted by a visitor*]

In the observations, O_bC_3 followed four distinct steps to compute the façade design, which are illustrated below (Figure 4. 26):

Panel (Input) and Panel (Output): is defined as "a panel for custom notes and text values. It is typically an inactive object that allows you to add remarks or explanations to a Document. Panels can also receive their information from elsewhere" (Payne, 2015, p. 115).

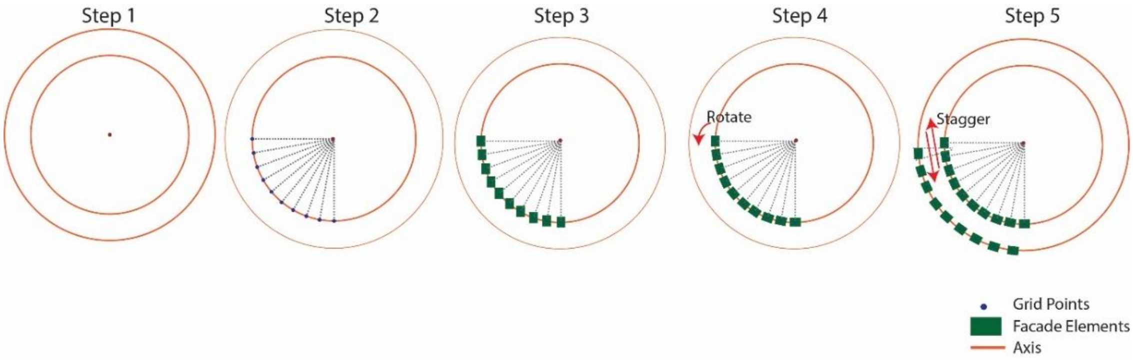


Figure 4. 26. O_bC_3's formula generation about facade design.
(Source: The diagrams were drawn by the author.)

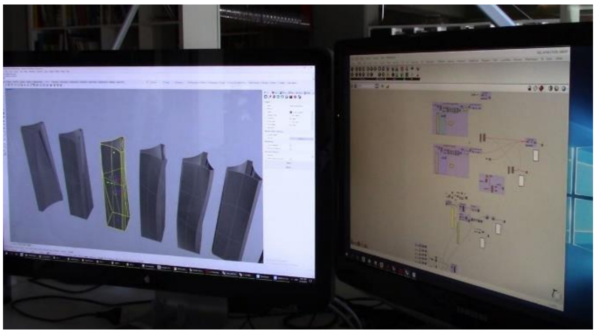
In Step 3 (Fig.4.26), O_bC_3 applied a specific component, namely Orient, into Graph Mapper which required a value and range for the rotation of the objects. In Step 4 (Fig.4.26), the Graph Mapper was applied to another plug-in called Stagger, which controls the arrays of the objects on each row. Inserting the value and modifying the parameter through the sliders provides instant visualizations of the geometries. The exploratory setup was made available by the functionality and capacity of the software.

In the second step of the design process, the second coder, O_bC_2 took over the control. O_bC_2's first strategy was application of the code developed by O_bC_3 on existing plan layout which was developed earlier. However, O_bC_2 started to change the geometry of the base and top plates of the towers to find the best geometric formation.

O_bC_2 started working with the generic code sent by O_bC_3 by first drawing a square plan on the base of the tower and a triangular plan on the roof. O_bC_2 then combined these two geometries using the 'loft' command. The geometries O_bC_2 drew on the base and roof were purely for experimental purposes. His goal was to create as much bending as possible to allow a rotation on the facade. In the following dialogs O_bC_2 describes how he applied the generic code to the towers:

00:00:17 R: "So, for instance… let's talk about how you created these forms."
00:00:42 O_bC_2: "…well, I draw a circular shape. Then I thought, how I can give this circular shape an angle in the best way possible, so that it can see the sea."
00:00:59 R: "So, is our goal to see the scenery?"
00:01:01 O_bC_2: "The view, our goal is to see the scenery. I drew lines from the center of it."
00:01:15 R: "Are you drawing them randomly or do they have specific angles?"
00:01:18 O_bC_2: "No, I drew 90-degree angles and 120-degree angles here. Sorry, I drew a 90-degree angle and brought it here. Then I thought, if I divide this circle into 3 equal parts with 120-degree angles, I might have given the smallest angle to face our scenery. This side facing the

scenery might be the most useful and efficient. Then I added an arch… that corresponds to 120 degrees. I rotated it 120 degrees and moved it inwards a bit.” (Figure 4. 27)



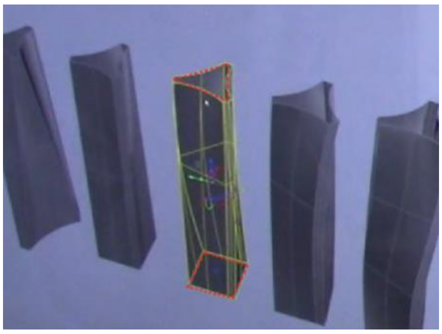


Figure 4. 27. O_bC_2 is creating a surface between the base and the top of the tower (left). O_bC_2 is experimenting with different geometries for the base and roof to explore the most amount of bending possible in the towers (right). (The red dashed lines show the base and roof geometries.)
(Source: The photographs are taken by the author.)

In the following conversation O_bC_2 explains how he tried out different rotations for the towers’façade to the scenery. O_bC_2 assigns random geometries to the top and bottom and observes the situation (Figure 4. 28). According to the 3D outcome he viewed, he further changes the shapes and make adjustments.

00:02:33 O_bC_2: “We could have also started from a triangle. But I started from a circle, I don't know why, I started from a circle. At first, I wanted to bring the base of this to a triangle from a circle, but then we changed our minds and started the base from a triangle at the bottom. After drawing this, I joined all of them together and we got this triangle. Then, I copied it to the top floor, rotated it 60 degrees and moved it slightly inward to fit inside, so we can easily measure our square meters. If we look now, the top floor is 650 square meters and the bottom is 1000 square meters, and 1100 square meters.”

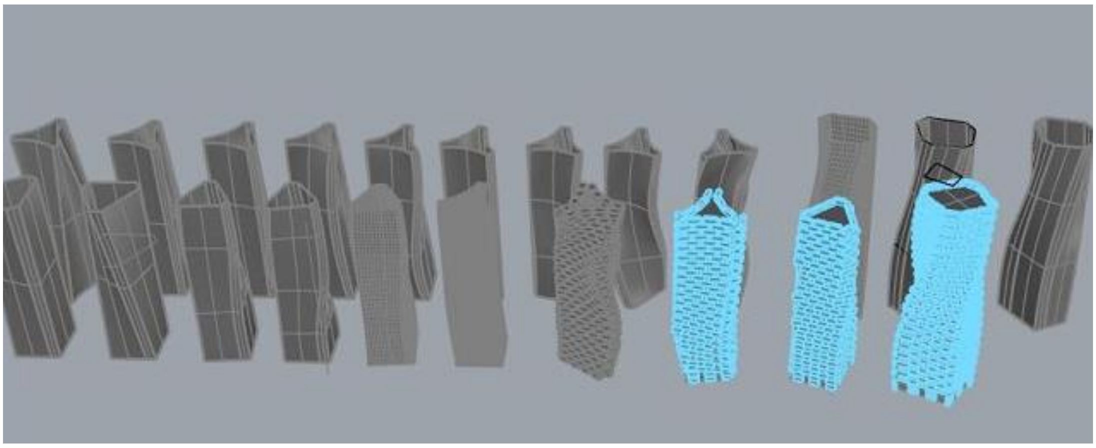


Figure 4. 28. Trials and explorations for the best rotating form for the façade, developed by O_bC_2.

(Source: Office B)

After identifying the geometries that provided the most bending, O_bC_2 started experimenting with the facades using the developed generic code (Figure 4. 29). O_bC_2 related the tower formations and worked on to the previously developed generic code in Grasshopper, they started observing the units' facades. The team wanted each unit to provide a solid-void pattern and spaces that could become either the facade or the terrace. While pulling the slider in Grasshopper, O_bC_2 started observing the changes on the 3D view on the screen.

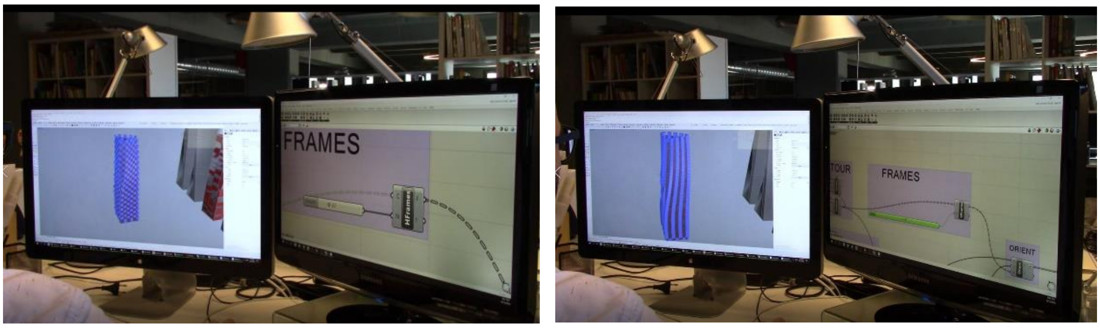


Figure 4. 29. O_bC_2 exploring the form and facade by adjusting the slider in Grasshopper and observing the movements of the facade elements, which are controlled through the generic code.

(Source: The photographs are taken by the author)

Afterwards, O_bC_2 tried to create the pattern on the facade by playing with the angles, while also attempting to direct as many units as possible towards the view. O_bC_2 was trying to discover the correct angle through experimentation. During the observations, while trying multiple angles, he arrived at a formation with a specific angle that he satisfied with.

00:03:32 R: “Why are you rotating it 60 degrees here?”
00:03:35 O_bC_2: “Now I'll explain that. The building we drew is facing this direction, but our view is on this side. I rotated it so that we can see the view more comfortably.”
00:03:51 R: “So, the one you started on the base isn't facing the view, but as you go up, you turn towards the view?”

Then, O_bC_2 started thinking aloud and explaining his thoughts while working:

00:03:56 O_bC_2: “No, no, now I'll explain. First, I generally turned our building towards the view, then I drew our triangle and said, "What if we started from the triangle, or from the circle, or from the square?" I'm going from the square to the triangle, or from the triangle to the upper triangle. Then I lofted them together, and I said, what if I go straight from the square to the triangle? And this is the façade we got. If we do a little rotation in between, how about from the square to the circle? Sorry, from the square to the rectangle. I gave up on the circle because it might be difficult to plan the inside of the circle, and we wouldn't catch the right angle, the widest facade. I did a slight rotation for the change in angle as we went up... then I did a little more rotation and tried to give it a different movement... Let's make a soft transition from the circle to the triangle. I added two triangles and got a soft transition from here to the triangle. Then I changed my mind and said, "What if we start with a triangle from the bottom?" I started with a triangle from the bottom and lofted it, then in the middle circles, I tried to add some movement to the building so that it wouldn't be too flat going up... I made a small adjustment in the building's height and made a sine transition.”

In the conversation above, O_bC_2 is explaining the exploratory process of designing the towers' façade. O_bC_2 started with a circle as the base shape, but later changed his mind and decided to start with a triangle on the lower levels. O_bC_2 rotated the triangle 60 degrees to better fit the view he wanted to orient towards. O_bC_2 also experimented with different shapes and transitions between shapes, such as going from a square to a triangle or a circle to a triangle. O_bC_2 made adjustments to the design to create movement and avoid having the building look too flat, while also used a sinusoidal transition in the height of the building to add variation. Overall, the design process involved trying out different shapes and transitions, and making adjustments until he found a design that the team were satisfied with.

O_bC_2 mentions that he started with a circle as the base shape for the building but, eventually, decided to go with a triangle. Trials of changing the shapes at the bottom and top provided chance explorations in the façade design and in the form of the towers

(Fig.4.30). Similarly, the small adjustments he made to the towers' design as it went higher may have led to unexpected and serendipitous explorations that ultimately shaped the final form of the building.

One of the intentions of the team participants was manipulating the design outcome intuitively by using manual operations on Rhino. The dialogs between the team leader (O_bTL_2) and the coder (O_bC_3) illustrates this:

00:09:40 O_bC_3: "If we go around widely, this thing that an ongoing thing, maybe we'll break away from here, you know, these two become a group, these two become another group."
00:00:15 O_bTL_2: "Exactly. Maybe it would be more proper to take it as a whole and go as sculpturing it."
00:08:38 O_bC_3: "It's not exactly as smooth as possible like that."

During this explorative process, O_bC_2 wanted to experiment with a different floor and ceiling geometries to try out alternative options for the pattern they wanted to create using the generic code, along with the bending in the form. In some of the trials presented in Figure 4. 30, O_bC_2 presented the team and the office leader (O_bL) with the orientation of the façade and the ratio of solid-void patterns for their evaluation. As a result of all these trials, the team decided on the design as seen in Figure 4. 31.

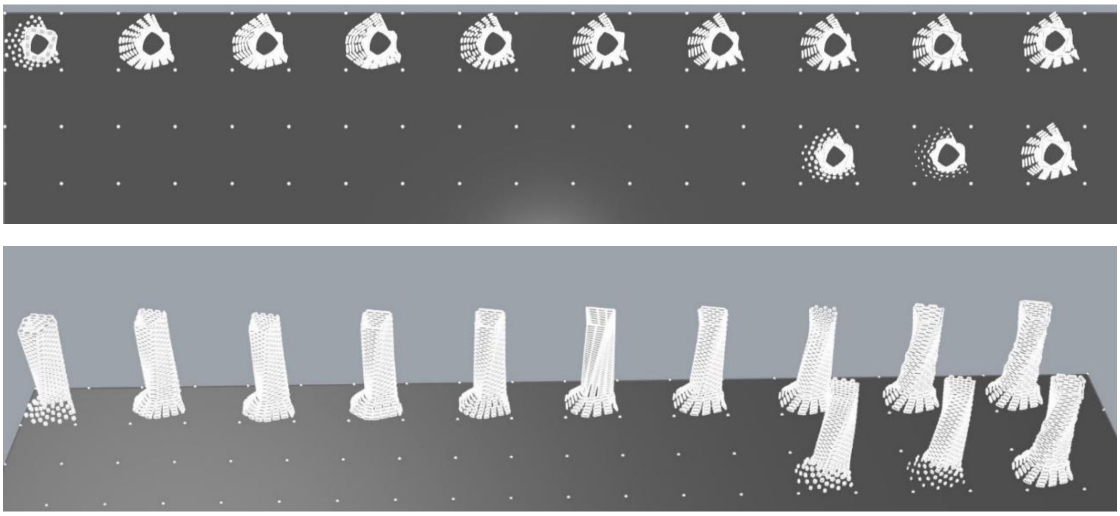


Figure 4. 30. Top and 3D view of the alternatives of the towers, developed by O_bC_2. (Source: Office B)

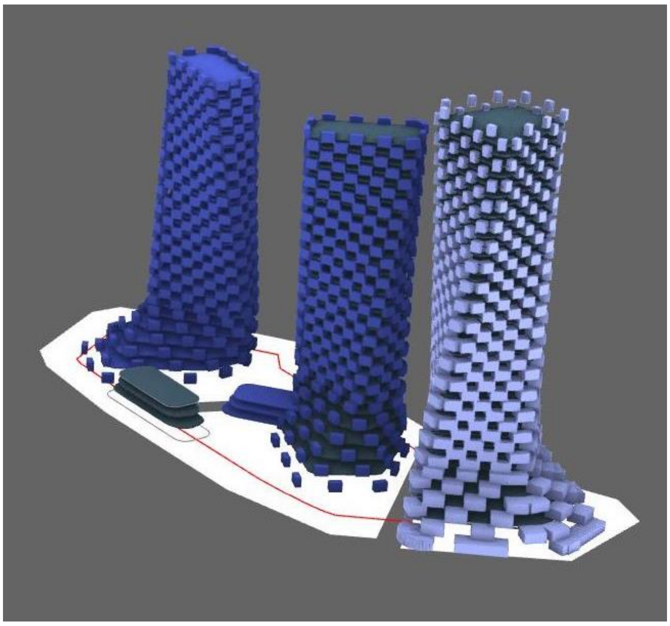


Figure 4. 31. The last version of the towers after appliying the generic code, developed by O_bC_2.

(Source: Office B)

4.2.3. Serendipitous Explorations in Tower Formation

In the tower project, the tools enabled the designers/coders work concurrently with the visualizations of the formulas. Hence, the office leader (O_bL) and the team leader (O_bTL_2) had the opportunity to observe what the formulas represent, and monitor the evolution of design While changing the parameters in Grasshopper, the coders (O_bC_2 and O_bC_3) shift the slider button as they observe the result on the Rhino screen for instant manipulations. The team leader and the coder observe the monitor and when they notice a potential alternative the coder pauses the slider and saves the alternative formation on the screen. Eventually, the integrated system -comprising of human and non-human members- generated a number of alternatives to be further assessed through other forms of representations.

The intuitive manipulation of the digital models were mainly governed by stylistic concerns. The formal manipulation of digital models was based on design intentions such as creating a smooth façade. The manual manipulations of the digital model by the designer, such as dragging an object on a 3D model in Rhino, can be seen as the team's efforts to associate the design results, which are generated by chance by the computer, with their design intentions.

The distribution of the tasks and workload among the coders both accelerated the process and enabled them to make further explorations by encountering different situations at each stage. For example, while O_bC_3 was developing the generic code, he examined many different ready-made components in Grasshopper until he reached the algorithm, he developed to control the units on the facade, and gained ideas that he could incorporate into the design as he examined them. While trying out these new components, he contributed to the design by exploring alternatives accidentally. O_bC_2, on the other hand, applied both a bending effect through defining top and bottom floor geometries, and a pattern to the towers through the generic code. While applying these two, he randomly applied different geometries to the floor and ceiling to achieve a geometry that could obtain more bending.

4.3. Exploration with Code

4.3.1. Process

In this section, the youth center competition project's design process in Office A, exemplifying exploration with code, is discussed. In the initial phases of the design problem, O_aC_1 and O_aIA_1 developed multiple alternatives. Each of the designers preferred to first sketch, then translate the scheme to the digital design tools. The office leader (O_aL) set regular meetings every day during the design process when the alternatives developed by O_aC_1 and O_aIA_1 were presented and discussed with all the team participants. In the first meetings, O_aL defined some rules for the design solution, such as creating a border on the site by using the units, or using arch formations to linkthe units. In this section, the design of the accommodation units' alternatives prepared by O_aC_1 is presented and discussed.

In the beginnings of the design process, O_aL separated the site into three zones (Figure 4. 32). The rationale behind zoning the site related first functional separation, second to the possibility of using the different parts of the site in different seasons, and third to leave the cost as pristine as possible. Zone 1 was reserved for accommodation units, Zone 2 for conference hall, and Zone 3 is the coast.

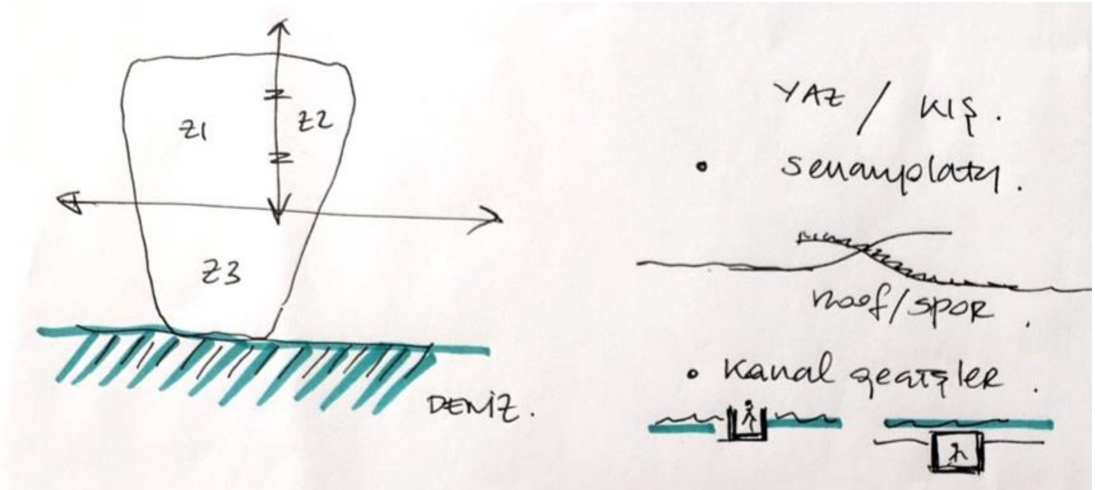


Figure 4. 32. The site was separated three zones: accommodations, social area, coast, developed by O_aL.

(Source: Office A)

Following the zones created by O_aL, O_aC_1 started to sketch alternatives of site layouts (Figure 4. 33). In the sketches, O_aC_1 was in search for a mass organization, introduced by terms like "*pixilation*" and "*puzzle*" (Figure 4. 33). In the interviews with O_aC_1, she explained the terms by referring to "hierarchy" or "rhythm".

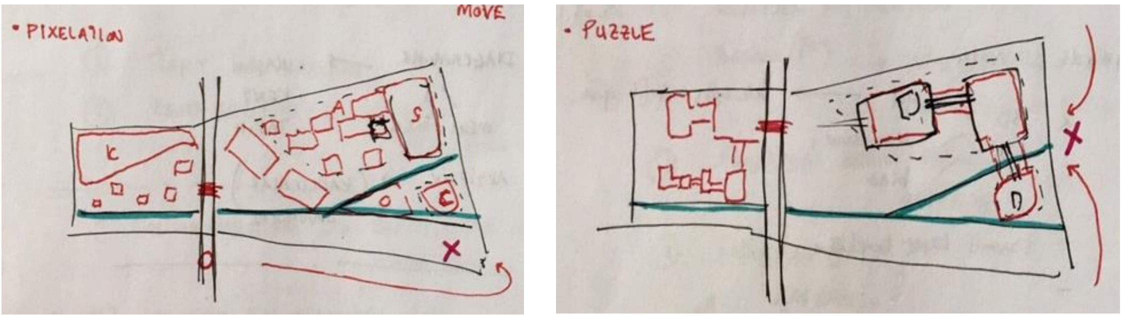


Figure 4. 33. The site mass organization sketches by O_aC_1.

(Source: Office A)

These terms were instrumental in the way O_aC_1 computationally organized and monitored the design process. She was advancing in a systematic manner and planning each stage. Although she knew how to use digital computational tools, O_aC_1 first generated sketches in her notebook and planned the initial steps of different possibilities. Then, she started experimenting on the computer to see the implications of this conceptual explorations. In the following section, O_aC_1's exploration is explained.

4.3.2. Explorations

O_aC_1, who is both a coder and a designer, started to design through traditional sketching on paper with algorithmic methods. In the design process of the accommodation units, O_aC_1 developed the units based on a 4x8 meter module. Afterwards, O_aC_1 started to search for rules that would govern how these units will be combined and sketched several alternatives via diagrammatic representations (Figure 4. 34). The motivation of setting a series of rules was creating multiple alternatives within the frame of the criteria.

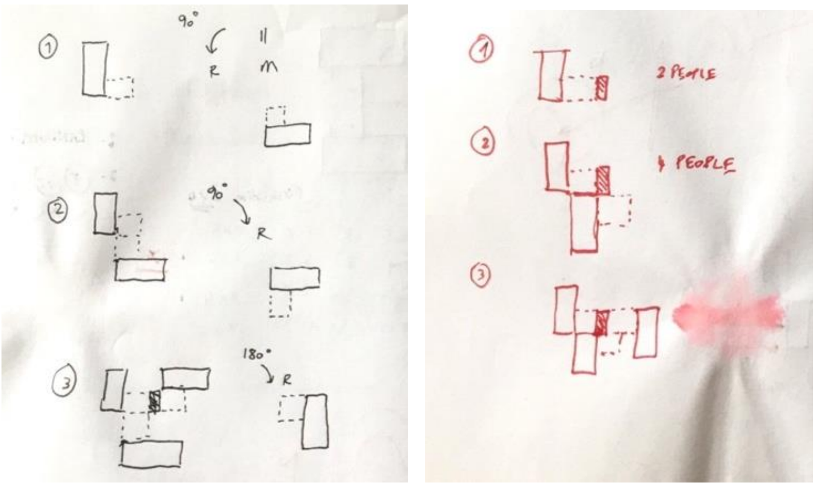


Figure 4. 34. Sketches about the rules, developed by O_aC_1.
(Source: Office A)

As it is illustrated in the following figure (Figure 4. 35), the rules that O_aC_1 develops during the design process of the units were created in two different sets. After O_aC_1 determined each rule, she transferred the rule diagrams to digital environment and adapted the units to their real sizes. While creating this rule set, O_aC_1 wanted to develop rules for every possibility and explore alternatives.

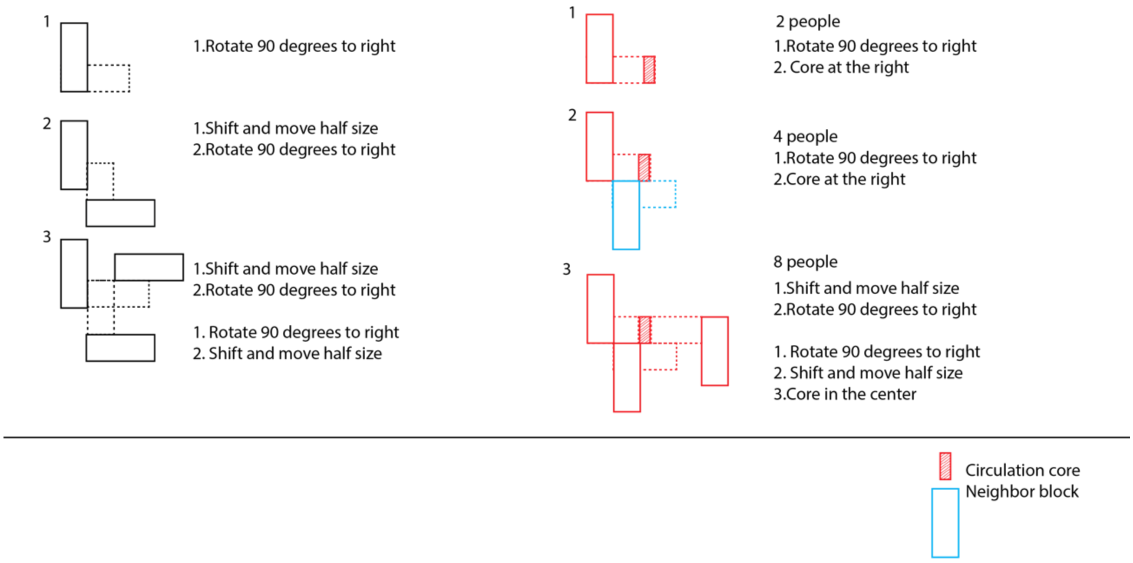


Figure 4. 35. The illustrations and explanations of the rules that developed by O_aC_1.
(Source: The diagrams were drawn by the author.)

As it shown in the sketches and illustrations (Figure 4. 35), O_aC_1 derives two sets of rules which mainly differ according to having an open core or an inner core. While setting the rules, some criteria were considered in the background. The criteria were determined in the earliest meetings with all team participants. The size of the units was determined in the team meetings as 4x8 meters. O_aC_1 used mainly rotate and mirror commands to have inner courtyards inside of the units. While applying rotation and mirroring commands, O_aC_1 was aiming to have a straight wall on one side of the units' border. Later, the units' border was mirrored, and a street was created between two lines. At the same time, O_aC_1 was also sketching site organization alternatives. Following the site sketches and the unit size decisions, O_aC_1 started toing and froing on the tools which are sketching, 2D drawing, and 3D digital modeling (Figure 4. 36). O_aC_1 tried to visualize different alternatives with hand gestures while trying to create a connection between the computer screen and sketching.

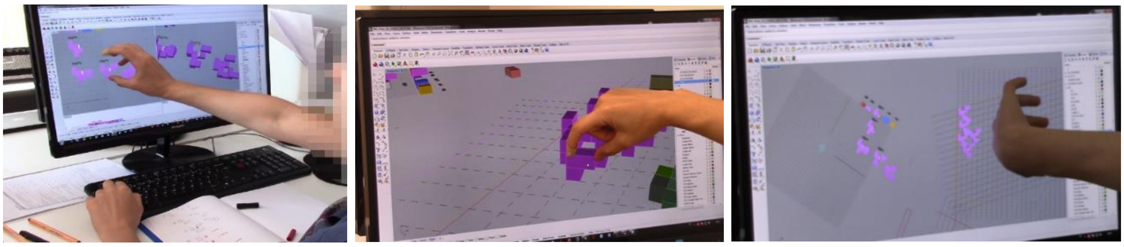


Figure 4. 36. O_aC_1 is creating an exploration space with the acts between the computer screen and the sketchbook.
(Source: The photographs were taken by the author.)

O_aC_1 extends her exploration space from traditional tools to digital environment while switching between sketching and Rhino 3D drawing. Her sketchbook was mainly used as a reminder and tracing the ideas that come to mind at first. Following the 2D sketches of the initial ideas, O_aC_1 re-presented the idea through the digital tool as 2D drawings. After modifying with exact sizes and re-arranging the units, O_aC_1 observed the rules that she created in 3D on the site layout. During the progression from sketching to 2D digital drawings, and to 3D digital drawings, O_aC_1 was chasing possible variations of the units' organization. By using more than one tool, O_aC_1 increased the chance of coming across unexpected alternatives while expanding the exploration. Moreover, the initial design idea morphed into three different design alternatives via three different design tools (Figure 4. 37).

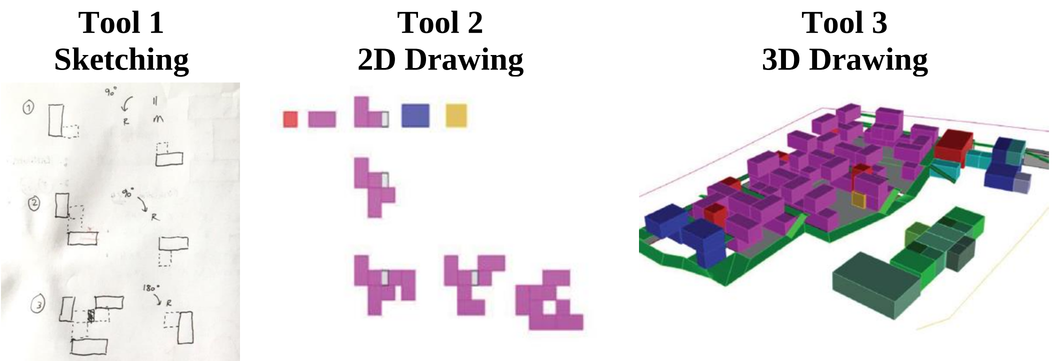


Figure 4. 37. Multiple tool using while setting an exploratory space by O_aC_1.
(Source: Office A)

During the Tool 1 stage in Figure 4. 37, O_aC_1 initially considered three different alternatives while sketching. By asking questions such as "what if it is like this, what if I try this, what can I achieve," and by introducing rules, she created a rule set. At the Tool 2 stage, O_aC_1 drew the sketched alternatives in 2D in Rhino. However, the work was not just about transferring them. According to the specifications, O_aC_1 also created various types of units: units for students, units for instructors, and units for disabled users. At this stage, O_aC_1 had further narrowed down the exploration area.

In line with the decisions taken at the meetings with others in the Office and with her personal inclination, O_aC_1 was trying to integrate a number of issues into the design. The considered issues comprised of 'randomness' and 'regularity' with seemingly opposite consequences. While trying to balance these two issues, O_aC_1 was running the design process using different commands such as random, mirror, and rotate. In the following dialogs, O_aC_1 and O_aIA_1 discussed the randomnesswithin the situation:

00:00:01 O_aIA_1: “I thought the same thing. For example, someone gave you a key, A23. It's difficult to find it if you don't have any direction. In holiday resorts or similar places, for instance, they say your room is here, and then you turn back from that point. At least, there is a guiding line for you. I cannot see this trio anywhere else in the visual [Figure 4. 38, highlighted in yellow]. Most likely, that's why someone will see the man here. Or, for example, the man over there is different from anywhere else.”

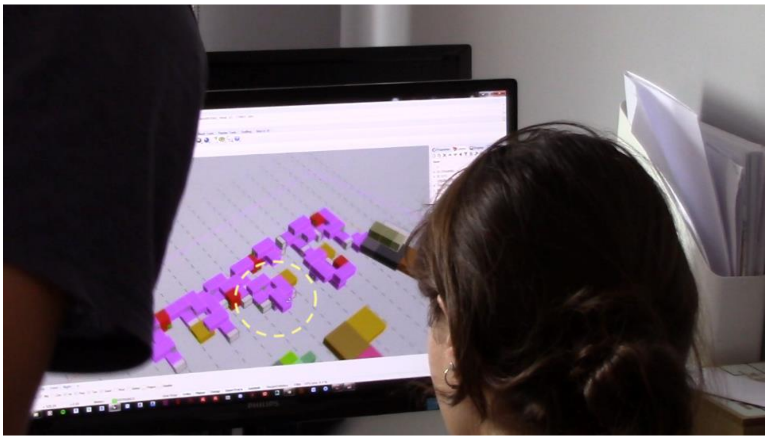


Figure 4. 38. O_aC_1 and O_aIA_1 are evaluating the design proposal on the Rhino screen. Yellow dashed lines [added by the author] highlight a group of units that the team participants focused on in the discussion.

(Source: The photograph was taken by the author.)

In this dialogue, O_aIA_1 says that O_aC_1's alternative is not uniform and the same, and this situation is actually a good advantage. At this point, O_aIA_1 starts to explain how O_aC_1 explored the forms randomly:

00:00:49 O_aC_1: "I really did it with random, mirror, rotation, etc... For some reason, it feels more reasonable to do it this way on the second attempt. We're going to multiply this, or I don't know, there will be a unit of 8. I think you put 8, 6, so you made a unit according to the rule. For it to be more solid, we made 8 and we did 6, we made 10, Maybe, 3 units, perhaps, disintegrate [she is explaining by rotating with her hands]. Because if we do the same thing all the time, will it get boring? I am thinking about that..."

O_aC_1 says it is "more reasonable to do it this way on the second attempt," indicating that they are working by way of trial and error rather than following a strict rule. O_aC_1 tries to place the units using movements like rotating, mirroring, as if playing with Lego, based on the volumes considered earlier in the process. O_aC_1 explains how to proceed in a completely random manner without strict rules. However, as she mentioned in the subsequent sentences, she also discusses parameters that narrow down and guide the exploration space; solid void balance, avoiding the repetition.

Afterward, when combining a few units together, she tries to replicate the module only horizontally, aiming to achieve a complex arrangement without repetition. O_aC_1 states:

00:27:05 OaC1: "I'm only duplicating without even rotating them, it's complicated enough."

During the design process of the units aimed at achieving a complex and non-repetitive arrangement, OaC1 followed her own design ideas in determining how to assemble the units initially. Later, in the stages involving how these units would come together in the site, OaC1 worked within a framework established by O_aL during meetings Figure 4.39).

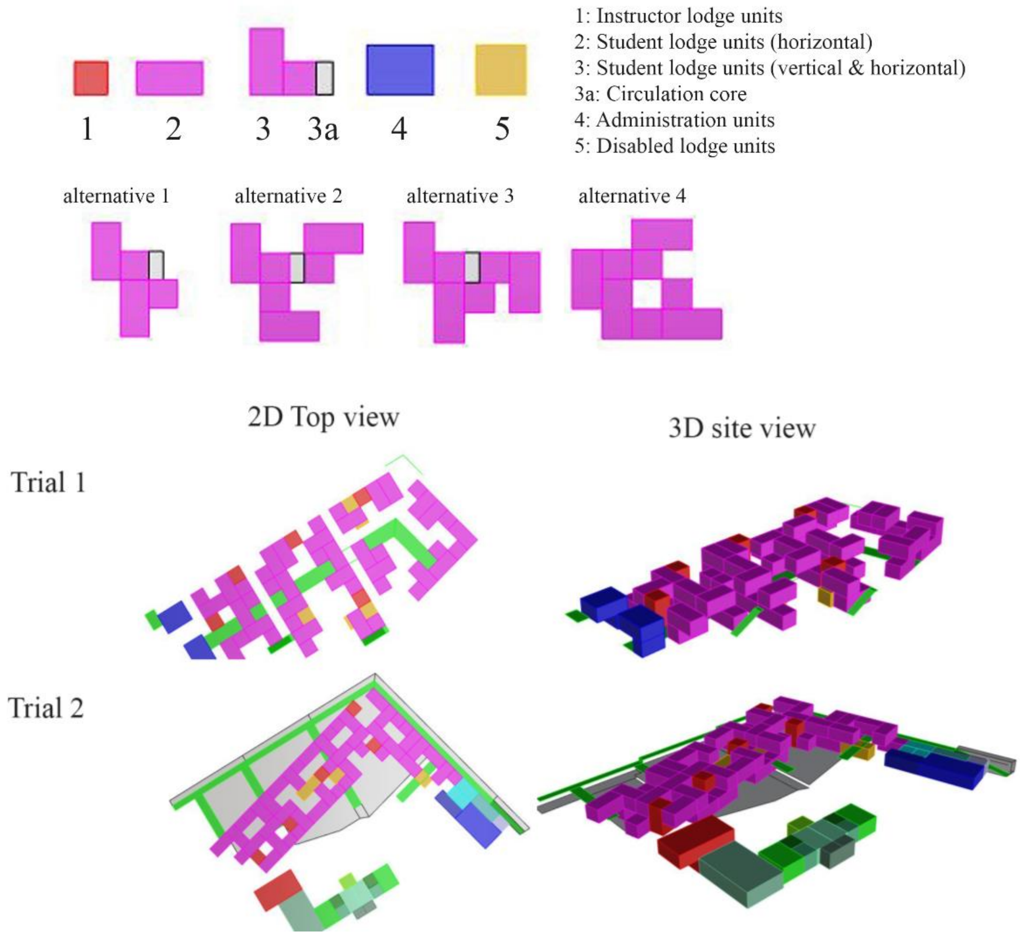


Figure 4. 39. (top) 2D drawings of the rule trials on the units and exploration of alternative unit organization. (bottom) some of the alternatives of the 2D and 3D drawings on the site.

(Source: Office A)

The situations that provided the exploratory environment during the design process for creating all these alternatives were as follows: firstly, while working with the O_aC_1 sketch, she transferred her mental process onto paper using a familiar and traditional tool for her. Then, using a similar method, she continued to transfer her 2D drawings from paper to Rhino, supporting the process with gestures during the ebb and flow between paper and computer, maintaining a secondary exploration. Afterwards, when O_aC_1 placed the "unit-core" arrangement on a framework on the site, she initiated another trial-and-error process. At this stage, as approaching the final decision, the team frequently discussed the outcomes in meetings.

4.3.3. Serendipity in the Unit Composition

In the youth center design process, O_aC_1 draws the dimensions of each unit as square meters on a 2D plan. Then she tries to create a set of units combined around a common space. After creating various types of combinations, she creates a site layout by duplicating them. However, the interesting thing that O_aC_1 does here is the effort to transform the combination procedure into a rule. O_aC_1, who has also coding expertise, was trying to determine the rules with a parametric approach. Considering numerical values and stages, O_aC_1 switched back and forth between hand sketches and 2D digital drawings.

In digital tools being able to quickly transform and visualize the form studied in the design process facilitates exploration. However, digital programs do not allow accidental or unplanned or unintentional discoveries. While digital design tools provide clear transformations in the design object through the precision of sliders, can making the transformations manually allow us to see the intermediate options? Or does physically manipulating a model, turning it by hand instead of rotating or mirroring it in the digital environment, by chance enable us to perceive the possibilities?

In the case presented, the designer extends the explorative spaces by adding different media and environments: sketching, 2D drawing, 3D drawing, and gestures. Within all these interactions, O_aC_1 intentionally benefited from a coding logic regardless of whether it is enacted digitally or not.

4.4. Discussions

The cases presented in this chapter establish the significance of design exploration by computational tools following the paths of knowledge propagation across the distributed system. The emerging significant point in this chapter is that in the way computational tools are used in current architectural offices, design exploration proceeds by mutually supportive contributions of experts who provide informed judgement and novices who are knowledgeable about coding and can thus produce many alternatives.

Within the distributed cognitive system, exploration in design proceeds through translations from one representation to another, a process primarily managed by coders and designers. Although this exploration is systematized with coding, multiple human and nonhuman coponents of the system ensure serendipity.

In the cases, the exploration of the unexpected solution undergoes while monitoring the exploration through set rules. The designers, while using computational design tools such as Mathematica or Grasshopper, play with the parameters and observe the 2D or 3D visualizations instantly on the screen to explore various solutions of the problem. The visualizations consist of enormous number of alternatives so, recording each step of the design process is essential. Thus, the team leaders could follow the evolutionary process of the design and take full advantage of the exploratory setting.

The team leaders, who have a 'prepared eye'(Goldschmidt, 2014), benefits from the production of alternatives that facilitates a serendipitous exploration. Preparedness is related with expertise (Goldschmidt, 2014) and the expert works only with visual representations. However, the novice puts the requirements into the coding and uncloaks all the possibilities. In this collaborative work, the expert has a role in evaluating the alternatives and turning into a solution. The creation of the alternatives is held by the novice. The generation of alternatives has been said to be related to expertise as Goldschmidt (2014), but knowing how to evaluate alternatives and being 'prepared' leads to a successful outcome (Goldschmidt, 2014; Kamprath & Tassilo, 2019). The novice designer/coder and the computational tool together enable a more ambiguous exploratory setting in which each alternative is quickly produced and archived. This ambiguity could present a wider exploration space for the expert to base the informed judgment.

In the studied cases, the teams sought an articulation of a rationale in design decisions. Between serendipity and rationality, designers try to create solutions that fit their design intentions. In this explorative environment, having multiple participants may be an advantage in terms of having various perspectives and ideas. Although architectural design teams tend to sustain creative input in their work, they may sometimes have to limit the effort dedicated to creativity due to time and managerial situations.

Some have interpreted design as planning (Rittel, 1971). Designing with computational design tools requires a level planning through generated algorithms in opposition to designing with sketching, which is a reflective action (Schön, 1991). Computational design tools require a visual translation of the outcomes of the algorithm whereas sketching is already a visual representation. Computational design provides enormous numbers of alternatives. Computational design tools might be enhancing creativity just like sketches do but differently. Sketches primarily enable multiple interpretations through their ambiguous, and syntactically and semantically dense ordering (Goel, 1995) while computational tools produce multiple alternatives. Computation presents an exploration space supporting ‘A-h-a!’ moments (Akin & Akin, 1996) in the creative process through generation of multiple alternatives.

Unconscious processes have been found to be related with creativity (Dreyfus, 1999). Sketching as well as computation could trigger these processes. Improvision, for instance, is a creative act that happen in a collaborative environment (Sawyer, 2017). Andel (1992) states that “a computer program cannot foresee or operationalize the unforeseen and can thus not improvise.” (pg.28). Computers cannot be surprised (Andel, 1992) because computers can calculate all the alternatives. Computers do not try different perspectives because they are not curios or they do not have a sense of humour (Andel, 1992). Creative things cannot be planned, or predictable (Sawyer, 2017). Sawyer (2017) expresses that innovative ideas cannot be planned or predicted; it is realized when it happenes. Only it needs to be allowed to emerge (K. Sawyer, 2017). For something to emerge, there must be a space and flexibility to explore. Computational design is based on parameters and specifications which designers define. The more flexible an algorithm is, the more alternatives it produces. However, the wider it is, it will be more far from design criteria. Improvision by using different medias such as various computer software and different tools such as sketching, model, digital model etc. allow designers to explore design solution (Laurel, 2003).

The cases presented demonstrate the effect of the computational tools in exploration in design process. Remarkably, computation involves rigid and accurate calculations, and precise measurement. On the other hand, serendipity seems to require the opposite. Louis I. Kahn (1930) expresses his thoughts about measurable and immeasurable aspects of a design:

> “A great building, in my opinion, must begin with the unmeasurable, go through measurable means when it is being designed, and in the end must be unmeasurable. The design, the making of things, is a measurable act. At that point, you are like physical nature itself, because in physical nature everything is measurable—even that which is as yet unmeasured... But what is unmeasurable is the psychic spirit. The psyche is expressed by feeling and also thought and I believe will always be unmeasurable. I sense that the psychic existence-will calls on nature to make it what it wants to be. I think a rose wants to be a rose. Existence-will, man, becomes existence, through nature’s laws and evolution. The results are always less than the spirit of existence.
>
> In the same way, a building has to start in the unmeasurable aura and go through the measurable to be accomplished. It is the only way you can build. The only way you can get it into being is through the measurable. You must follow the laws, but in the end, when the building

> becomes part of the living, it evokes unmeasurable qualities. The design involving quantities of brick, method of construction, engineering is ended and the spirit of its existence takes over." (Kahn, 1930; p.11)

As Kahn stated, it is important to start a building design by considering immeasurable elements, expanding the realm of exploration, investigating the design from a broader perspective, and taking into account the socio-material environment. Architects cannot fully calculate every aspect of a structure, so they may seek interdisciplinary support in the dimensions they want to calculate and comprehend. While they may attempt to calculate each step in some cases, it has always been enticing to see unexpected or unforeseen alternatives emerging from the situated activities involving design representations. Therefore, architects frequently have the desire to sketch and scribble.

In the presented cases, the architects and the coders have expanded their exploration processes and domains by utilizing tools and methods that allow them to broaden their exploration. Not being fully proficient in the tools has made them more open to coincidences but has also presented them with more alternatives. Instead of inputting rigid numerical values to achieve a specific form, they have used the tools in a more intuitive manner, taking advantage of the aspects that can be manually manipulated within the tool. For example, playing with sliders, unintentionally connecting components found through search engines, and making an effort to combine traditional methods with digital tools have greatly facilitated the achievement of various forms in the presented cases. The computer presents the design product as if it is finished (Turkle, 2009), therefore it leaves no room for ambiguity. The designer seeks to leverage the flexibility of tools to increase the space for exploration.

Designers have a purpose in their mind while engaging in design problem whether ill-defined or well-defined (Goldschmidt, 2014). Regardless of whether the purpose is ambiguous or clear, designers intuitively search a solution back and forth between associative and analytic thinking which supports creativity (Gabora, 2010). Although many alternatives are produced through computational tools, a 'prepared mind' (Goldschmidt, 2014) can evaluate alternatives and turn them into a creative serendipitous moments. The office and team leaders in the cases were in a position to monitor each emerging alternative while the coders worked with parametric modeling tools to sustain the process of formal exploration. As the coders manipulate the parameters of the formula, the tool continuously and synchronously generates alternatives that are evaluated by the team leaders.

Just as sketching, digital-doodling on computational tool and 3D visualizations is also iterative processes which progresses through coding, observing, and re-coding. Instead of progressing with an already envisioned result-oriented approach in the design process, the teams were trying every possible way they can discover during the process. In the cases, the team participants repeatedly went back to the beginning and continued to try repeatedly. In this way, going back again and again increases the chances for serendipity.

In the design process, unexpected design solutions can emerge as long as ambiguity prevails. Coders, while using computational design tools, combine their architectural knowledge with developing coding skills to utilize information and experiences from multiple disciplines. Establishing interdisciplinary connections and promoting with a holistic problem solving approach are often recommended (Dyer et al., 2021). Uncertainty can indeed arise from a lack of adequate knowledge about a particular domain, as mentioned in the study by Dyer et al. (2021). However, in the cases presented, the inadequacy of knowledge about the computational design tools actually encouraged the exploration of new alternatives while utilizing those tools.

Serendipitious explorations are powerfull for design problems (Suwa et al., 2000). In the youth center project, for instance, the coder preferred sketching in the early phases and after she transferred the initial ideas to the computer. Both of the steps were open to unexpectedness in means of interaction with the tools. (Goldschmidt, 1994)

The act of sketching on a paper can reveal more information than what was initially put into it (Goldschmidt, 1994). However, in the cases presented, the act of manipulating geometry through computational tools were observed to introduce additional alternatives. This is made possible by the emergence of new combinations and relationships among the elements of components in Grasshopper or Mathematica, the modification of parameters in code, and the utilization of other applications added to the design process, such as the tools like Grasshopper, Mathematica, and Rhino. These unexpected explorations may arise during the process of sketching and using computational tools.

Yu et al. (2015) claim computational design is a dynamic and rule-based process. Computational thinking relies on setting and organizing the rules for development of forms but, designer does not obviously define a final form (Poulsgaard & Malafouris, 2017). In the youth center project, the coder while setting the rules did not clearly know the final form. In computational design, the final design solution does not need to be precise. However, in the design process, the interruption of the seeing-drawing-seeing process can lead to an end design arbitrarily.

CHAPTER 5

LEGIBILITY OF DESIGN IN COLLABORATIVE COMPUTATIONAL PRACTICES

This chapter presents and discusses the design process of two separate architecture offices where two teams of architects were tasked with generating a schema in early phases of architectural design primarily using computational design strategies. The intention of the design team leaders in both cases was to make the design idea -and design process in general- more legible and transparent for individuals within the team as well as others including clients and consultants. Through situated observations of these collaborative computational practices, how design ideas are represented and externalized in a distributed cognitive system with the intention of achieving a legible schema to guide the design process were observed.

5.1. Legibility to Communicate to Others

This section presents two different sections from Office A, each from a different project. The first one pertains to the design process of the accommodation units in the youth center project. Secondly, a section from the facade design of the Municipality building project is presented. During the design meeting that was conducted with the coder, interns, and the office leader, the concept of "legibility" was introduced by the office leader as a design criterion to the team.

5.1.1. The Youth Center

In this section, an episode from the youth center project is discussed in detail. The participants to work on this project at Office A includes O_aL, O_aC_1 and O_aIA_1. In the initial phases of the design problem, the coder (O_aC_1) started to design in a rule based approach. The task was to develop a site plan by generating a compositional rule to organize pre-determined lodging units for the youth center. The design team had made some key decisions during their regular meetings since the beginning of the competition process. Based on these decisions, O_aC_1 started working on a layout for the units on the site. The team participants, including O_aC_1 and an intern (O_aIA_1), were making efforts to establish a rule in their dialogues. In an unplanned meeting with O_aC_1 and O_aIA_1 the following dialogues unfold showing how the rule based design process has proceeded (Figure 5. 1):

00:00:01 O_aIA_1: "I thought the same thing. For example, someone gave you a key, A23. It's difficult to find it if you don't have any direction. In holiday resorts or similar places, for instance, they say your room is here, and then you turn back from that point. At least, there is a guiding line for you. I cannot see this trio anywhere else in the visual [Figure 5. 1, highlighted in yellow]. Most likely, that's why someone will see the man here. Or, for example, the man over there is different from anywhere else."

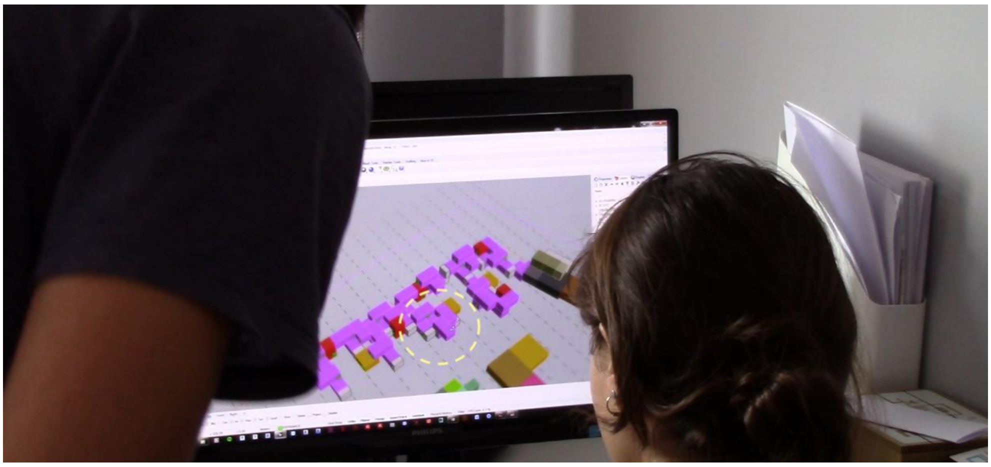


Figure 5. 1. O_aC_1 and O_aIA_1 are evaluating the design proposal on the Rhino screen. Yellow dashed lines [added by the author] highlight a group of units that the team participants focused on in the discussion.
(Source: The photograph was taken by the author)

In the dialogue above, O_aIA_1 expresses his concern about wayfinding in the architectural layout. While sharing his own experiences, he mentions that the spatial organization itself should guide the users through the facility. O_aIA_1 positively views the "non-repetitive" layout in the coder's alternative and states that as a result, it will be easier for the user to find their way in the proposed layout. At that point O_aC_1 starts to explain her proposal:

00:00:49 O_aC_1: “I really did it with random, mirror, rotation, etc... For some reason, it feels more reasonable to do it this way in the second attempt. We're going to multiply this, or I don't know, there will be a unit of 8. I think you put 8, 6, so you made a unit according to the rule. For it seems to be a little fuller, we made 8 and we did 6, we made 10, Maybe, 3 units, perhaps, disintegrate [she is explaining by rotating with her hands]. Because if we do the same thing all the time, will it get boring? I am thinking about that...”

O_aC_1 mentions on one hand that she will replicate the masses according to a certain rule, but on the other hand, she believe that if they progress in exactly the same way, it will turn into a highly repetitive design and could become 'boring':

00:01:44 O_aIA_1: “Exactly, that's exactly what I was going to say. For example, we can make it for 3 people, 10 people, or 8 people relatively [unclear]. Initially, we needed to have a crowd here [Figure 5. 2, referring to the part marked in yellow]. Just to make it more public, to bring it closer. You can put 2 of 15 here and 2 of 10 here, and then it goes 8, 8 after a certain point.”

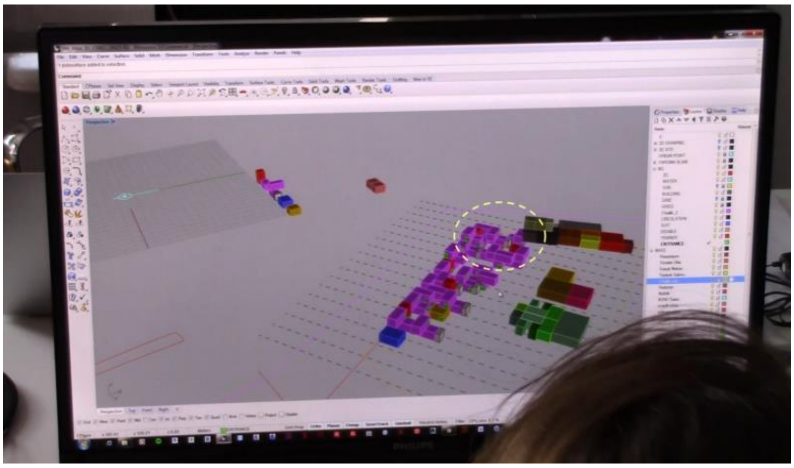


Figure 5. 2. O_aC_1 and O_aIA_1 are evaluating the design proposal on the Rhino screen. Yellow dashed lines [added by the author] highlight a part of the units that the team participants focused on in the discussion.

(Source: The photograph was taken by the author)

In the continuation of the dialogue, O_aIA_1 agrees with O_aC_1 and suggests a more dense composition in some parts and a sparser one in others. This way, they try to find a solution to the problem of 'boring' by introducing varying levels of density within the composition.

00:02:10 O_aC_1: “I suggest we do it that way. [unclear] There shouldn't be just one. Let there be multiple, and they should be in multiples of 8, 12, 16, four times our number. Of course, it will be readable [legible], but at least you will look at it twice and say, 'hmm, there's a rule here.'”

In the above sentence, O_aC_1 expresses that what they create with the 'rule-based' approach may not be understood by a designer and/or a user at first, and that recognizing it on the first attempt is crucial. They emphasize the importance of a rule-based design being understood by its users in the dialogues:

00:04:50 O_aIA_1: “I think there can be a clear circulation line and circulation lines connected to that line. Remember, I showed you yesterday, we'll create a unit like that, where I can enter vertically, or it directly connects to the horizontals, it will be clear. For example, these 16 units will have one circulation line vertically or two vertical circulation lines, and we will connect them to the horizontals. In fact, when I reach that circulation, I will see room numbers and everything. In this area, it's like the courtyard concept you mentioned, so at least I will be able to see that in all of them. After that, we can even do copy-paste without obstructing the exits. Similarly, in the next unit, for example, we can adapt it in a way that doesn't obstruct the previous one. It multiplies, the next one multiplies, it keeps multiplying and producing. By doing this, the arches (Figure 5. 3) take up a lot of space. I thought we could also find large arches [archs will be added to the gaps marked in yellow].”

$O_aIA_1$suggests advancing the design by duplicating the units in a repetitive manner after obtaining an initial combination. This way, the design will follow a set of rules and will not have a monotonous repetitive pattern.

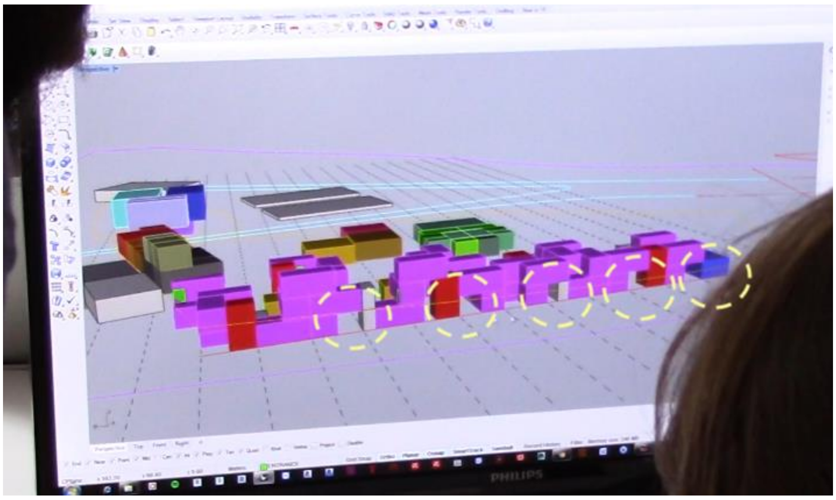


Figure 5. 3. O_aC_1 and O_aIA_1 are evaluating the design proposal on the Rhino screen. Yellow dashed lines highlight a part of the units that the team participants focused on in the discussion [yellow dashed lines added by the author].

(Source: The photograph was taken by the author)

After their brief conversation, O_aC_1 and O_aIA_1 had a meeting with the office leader (O_aL) to discussrearrangement of the units based on a site layout drawn by O_aL (Figure 5. 4).

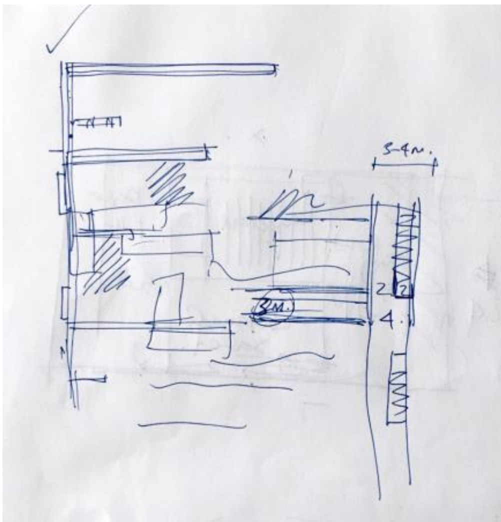


Figure 5. 4. The site layout that drawn by O_aL in the team meeting.

(Source: Office A)

The next day, they had another meeting during which O_aC_1 presented the alternatives she had generated:

00:04:50 O_aC_1: "Well, I tried a bit, but it didn't really work out. In general, I just repeated a unit. It creates a mirror effect. And there are three emerging scenarios."
00:05:17 O_aL:" Why is there such a thing?" [*While examining the drawings*]
00:05:25 O_aC_1: "Are you asking about the green part or the opening?"
00:05:27 O_aL: "Here, we're creating an axis like this [*referring to the inverted L-shape from above*], forming a central courtyard. Is there an obstruction here?"
00:05:34 O_aC_1: "Actually, it's just that, on the ground [*unclear*], especially with the bridges above [*unclear*], since I made the courtyard in general for all units, yes, a courtyard is formed here as well."
00:06:00 O_aL: [*quickly zooming in and out repeatedly*] "because in this diagram and in O_aIA_1, the things we discussed yesterday weren't done, we'll be using the diagram we discussed yesterday. It's important for the buildings to be in that flow. When you distribute them, it starts turning into a completely different project. If we draw two lines here, for example, it will just continue downwards in this flow. If we offset one, we want to emphasize an outer edge in this flow, especially when you distribute the masses, if you place obstacles here, it elongates like this. The direction of the buildings is wrong. I should see a green area outside here. I should see a dense purple [*referring to the color of the units' layer*]. And then it should gradually resolve towards this direction."

O_aL criticizes them for not continuing based on the sketch site layout (Figure 5. 5) that was intended to guide the next steps in the process and explains how they should design the units in an organized manner. They emphasize the need to design in a way that gradually dissolves, while also providing a set of rules for the arrangement.

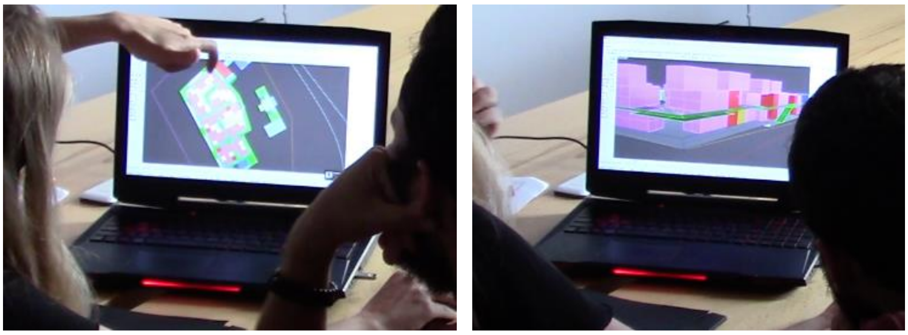


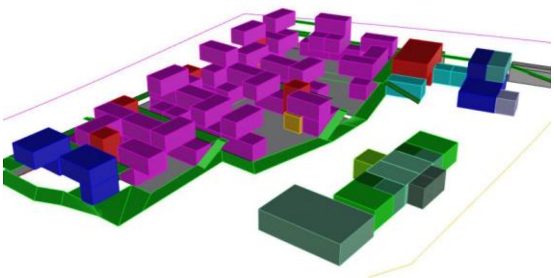
Figure 5. 5. O_aC_1 presenting her alternatives to the O_aL and the team participants, plan view of the alternative (left), perspective view of the alternative (right).
(Source: The photographs were taken by the author)

After criticizing the density of the circulation line [drawn with a green layer] and units [drawn with a purple layer], O_aL explains that their main goal is to establish a rule.

She restates the design will become understandable and legible only through a rule to be followed in creating the masses:

00:10:44 O_aL: "It seems that the elevation level is not in the right place in this area. The elevation should be located somewhere else. For example, if you say that it should have a two-way relationship... let's not go for three [referring to three levels]. Let's finish it directly on the second level. One thing is certain, it should be a completely enclosed mass [referring to the social space]. We should be able to move inside without going outside. You need to compactly arrange them in a way that we can no longer break them apart. That's what I mean. **So, what is the rule?** The thing that bothers me the most is that **I can't read it immediately**. I mean, I can't read the distribution rule, so to speak. Yesterday, everything was more legible, I mean in the proposals. Let's open yesterday's suggestions in yours (Figure 5. 6) [*meaning his computer*]." [*Emphasis added by the author*].

O_aL's main complaint was that it was hard toimmediately grasp the idea behind the alternatives and stated that the proposal was an unclear one in terms of the legibility of the architectural design intention. She requested to take a look at the previous models together with the team to have a better comparison.



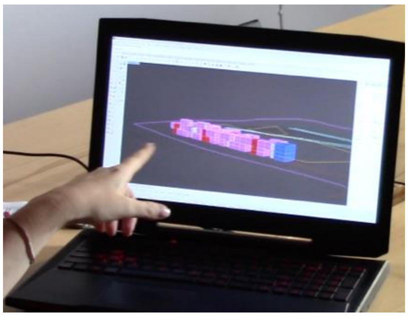


Figure 5. 6. O_aC_1 previous work that O_aL found more legible.
(Source: The photographs were taken by the author)

After opening the previous work, O_aL started explaining the rule and what she meant by "reading" the design:

00:16:13 O_aL: "Now, without a clear rule, when it's random, it becomes as if we just placed things randomly without any purpose. No architect can understand it. They cannot interpret something they cannot read. That's what I'm trying to say. They should be able to understand it at a glance. Today, I couldn't understand it immediately when I looked at it. Can I explain myself? I couldn't grasp the rule and its purpose. Yesterday, when you showed me your design, I could understand what you were thinking without you even explaining it. But now, it's lost. It has become too random. It's the same with yours [referring to O_aIA_1]. When you go into too much detail, it obscures the main idea. I can see that, and now we need to find a way to rescue it. Let's see what we can do."

According to O_aL, architects should be able to read and interpret the design, understand its purpose and intention. The rule establishes coherence and order within the design. Each unit is placed and related in accordance with this rule. However, apart from being understandable and legible by other architects, it is also important for the designers themselves to have a clear design understanding and for the spaces to be easily comprehensible. O_aC_1 and O_aIA_1 set rules both because of their intention to implement a computational approach and to achieve a legible design (Figure 5. 7).

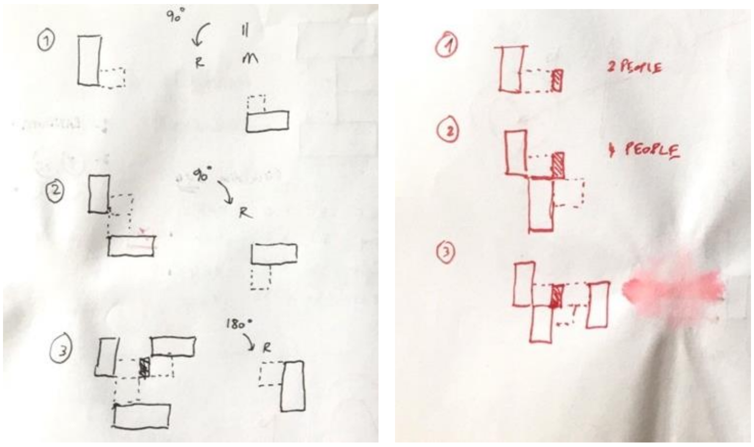


Figure 5. 7. Sketches about the rules, developed by OaC1.
(Source: Office A)

The illustrations in the Figure 5. 7 expresses the rules that O_aC_1 was trying to set in the organization of masses for the youth center. In the left row step 1 is the basic unit of the system which is copied and rotated in 90 degree to the right; at the step 2 the unit is shifted along half of the edge of the rectangle and rotated in 90 degree to the right. At the step 3, O_aC_1 adds to step 2 and she rotates the unit 90 degree to the right and shifts half of the long edge of the rectangle. In the right row of the Figure 5. 7, O_aC_1 adds a circulation core (a rectangle with hatch pattern) and adapts the rules according to the core. O_aC_1 begins to think that at this stage there will be a two-story structure, and tries to observe the emerging outcome when the upper and lower floors are superimposed on the site, along with the relationships between them. In the Figure 5. 8, the sketches of O_aC_1 about the units were illustrated according to explanations in the interviews.

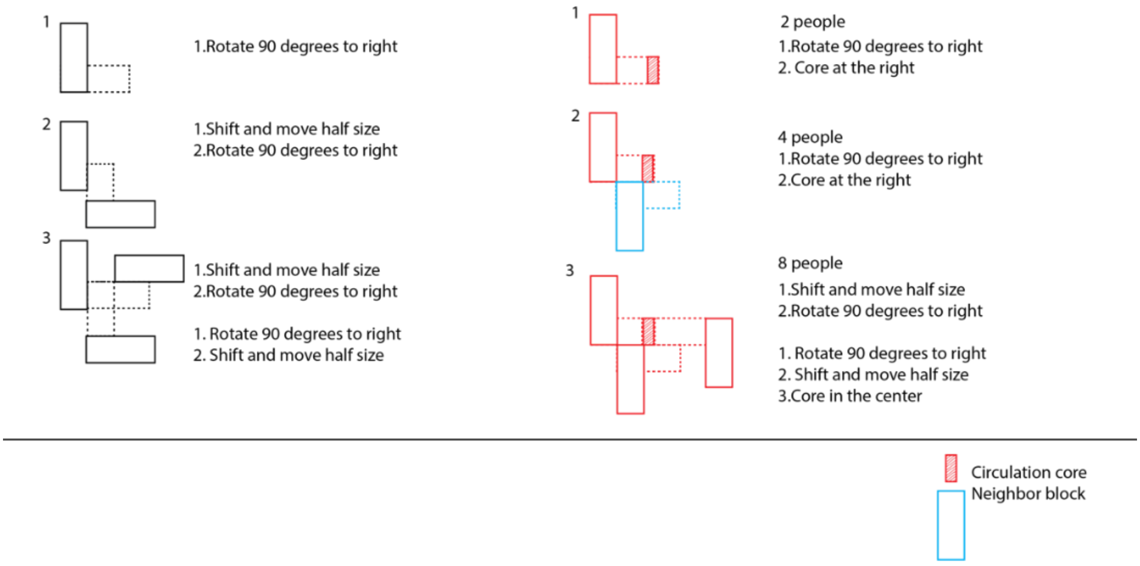


Figure 5. 8. The illustrations and explanations of the rules that developed by OaC1.
(Source: The diagrams are drawn by the author)

The illustration (Figure 5. 9) presented below depicts the working principles and stages of a coder. It showcases the design movements each unit undergoes at each step.

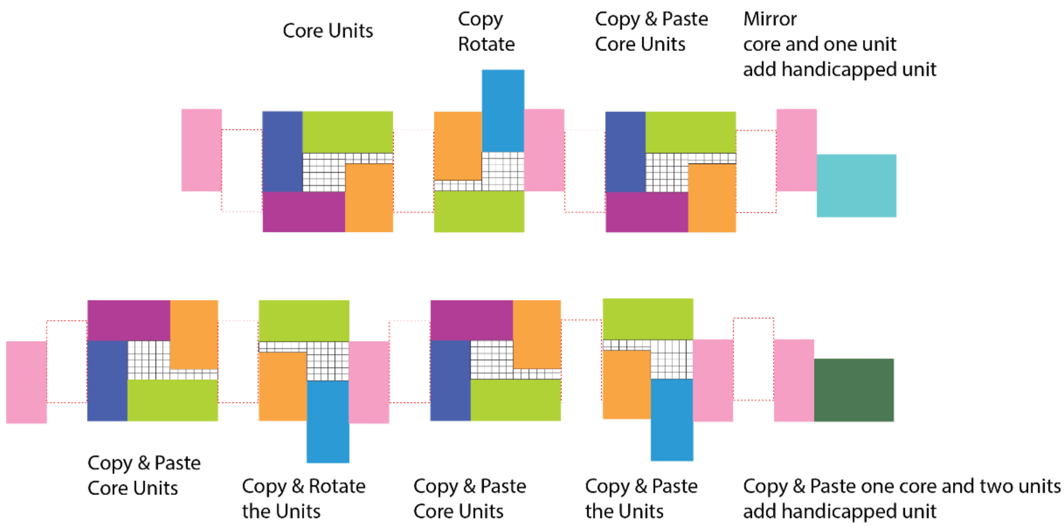


Figure 5. 9. The illustration of the final submission's unit organization.
(Source: The diagrams are drawn by the author)

At the end, following the rules they established, the team reached a solution and submitted the site layout in Figure 5. 10 at the end of the competition process.

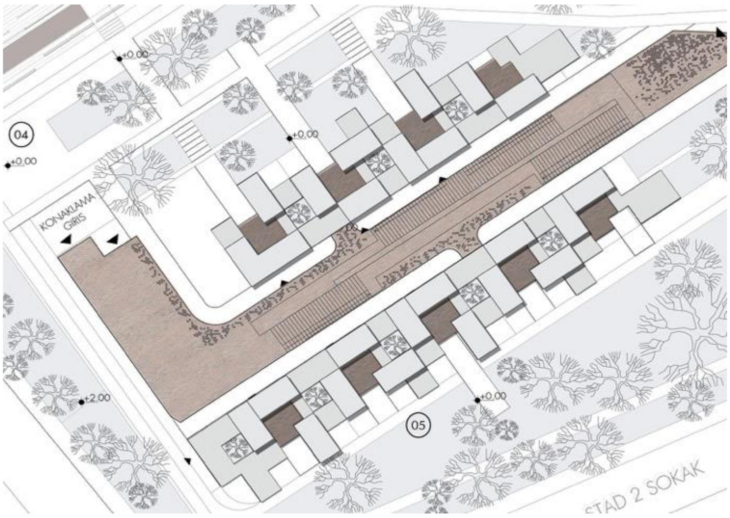


Figure 5. 10. The final submission of the site layout and the units’ organization to the competition.

(Source: Office A)

The rule-based search for arranging the units was also extended to the design of the "arch" arrangement between the units and the layout of glass bricks on their facades (Figure 5. 11).

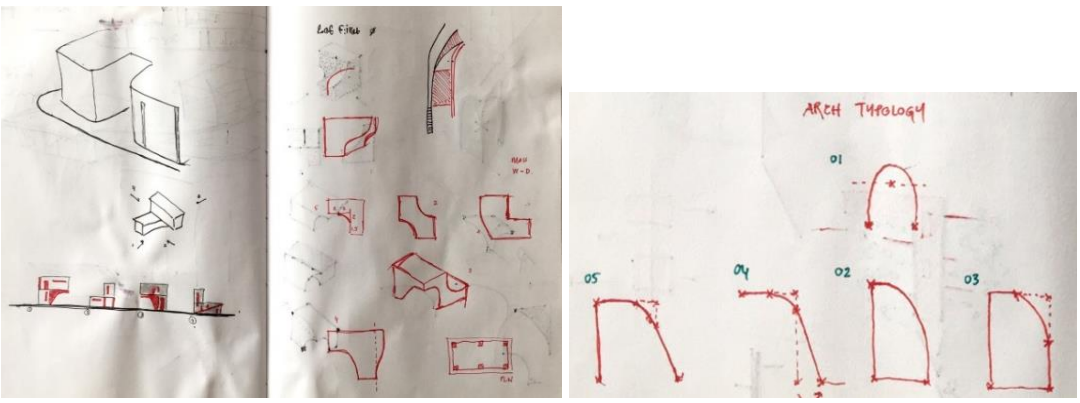


Figure 5. 11. (left) arch typologies, developed by O_aC_1, (right) adapting arches on the units’ façade as 3D sketches.

(Source: Office A)

The rules were also established to integrate arches into the design of the façade. The integration of arches was guided by the following criteria: (1) placing arches on terraces and (2) using arches on passages between the units. In the sketch (Figure 5. 12, left), two alternatives for designing arches were utilized in different parts of the project: alternative 3 was used on the units, while alternative 4 was employed in the design of the social area's roof. After adapting the arches to the façade, O_aC_1 worked on the location and forms of the windows. Firstly, O_aC_1 sketched and determined the placement of openings (Figure 5. 12, left), and then developed façade sketches to visualize the openings with the arches (Figure 5. 12, right).

For the openness, three different patterns were generated which are square, rectangle, and shifted rectangle. The patterns for the openness were developed in Grasshopper by writing scripts (Figure 5. 13). In the Figure 5. 14 four different situations of the openings on the façade, which are associated with the interior spaces, were illustrated.

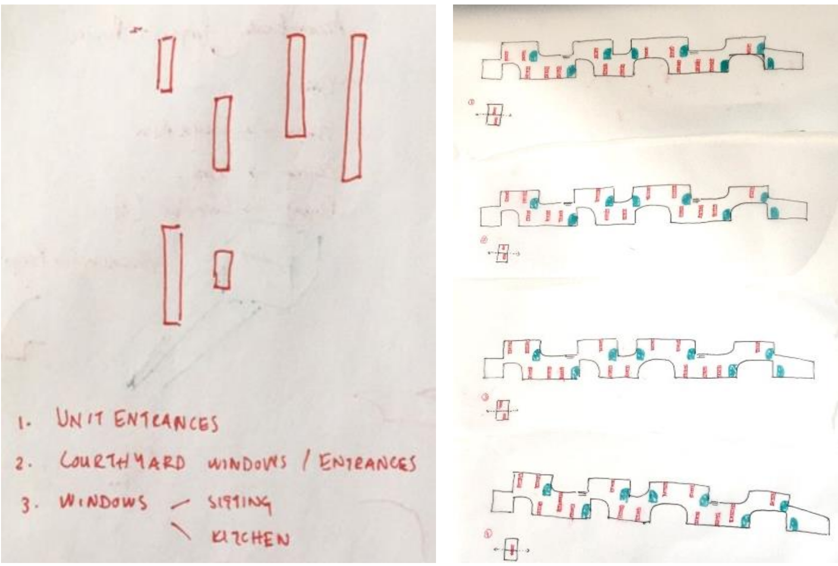
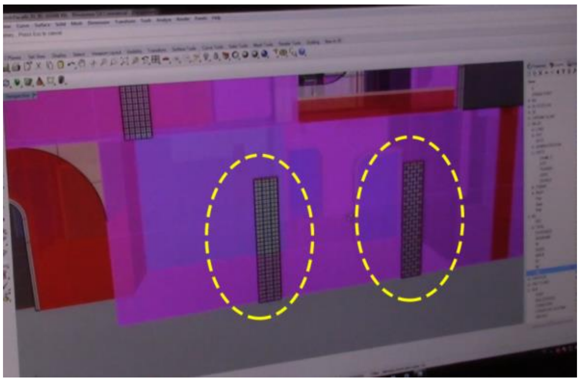


Figure 5. 12. (left) sketches of opening unit alternatives, (right) sketches of opening alternatives on the façade.

(Source: Office A)



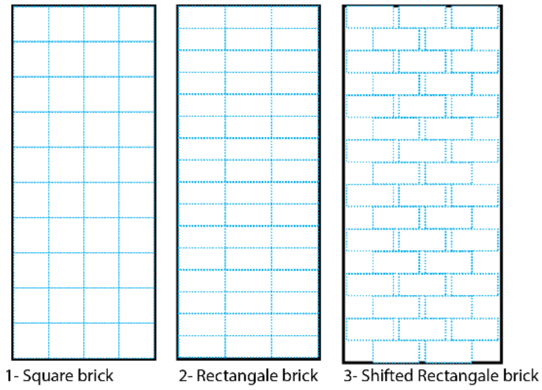


Figure 5. 13. (left) the Rhino screen while O_aC_1 is appliying the glass bricks on the opennings, (right) the illustration of the glass orders' according to the O_aC_1 proposal. (Source: Office A)

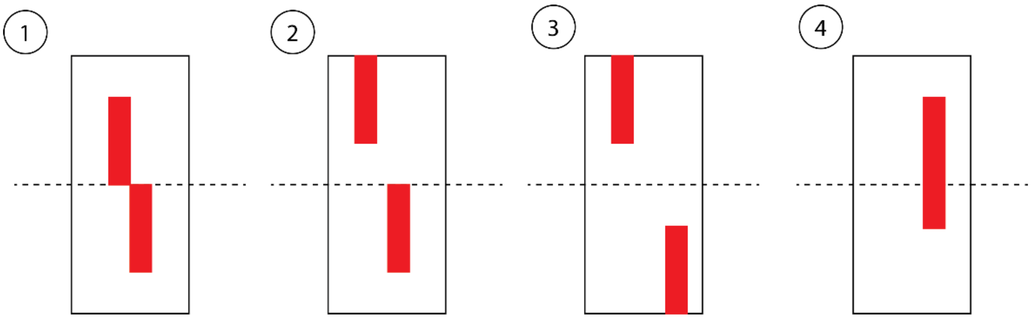


Figure 5. 14. An illustration of the opennings' variations on the units' façade. (Source: The diagrams are drawn by the author)

In the explanation below, O_aC_1 describes how the openings she designed on the facade are connected to a series of rules and presents the sketches O_aC_1 created to explore alternatives. Depending on the purpose, the openings O_aC_1 will propose vary according to the characteristics of the spaces within the units. Therefore, the openings on a kitchen façade differ from those on a bedroom façade, supporting the aim of the design to be legible.

O_aC_1 :.. "Although I can't remember the exact architectural program of a room, the opening with bricks was roughly applied for the exit door from the living room to a balcony, the main entrance door from the foyer to the room, and the skylight from the kitchen and bedroom. Again, when these 4 elements were settled in a resolved unit, they automatically found their place in the overall whole.

They are not completely random, free sketches…for reasons related to surprise, preserving the axis of symmetry or aesthetic concerns. Do not extend the rule of the main unit to the overall. The difference between these sketches is that proportions are tested. In other words, it is the idea of testing this proportional situation in general, such as whether the door opening should be zeroed to the corners or not, whether the brick walls should start from the ground, whether they should be in the middle, in the middle, or should they lean against the ceiling".[*manuscript by the coder (O_aC_1) on e-mail at Jan 3, 2019, 10:34 am*]

The final delivered facade is presented in Figure 5. 15 which includes arches and openings.

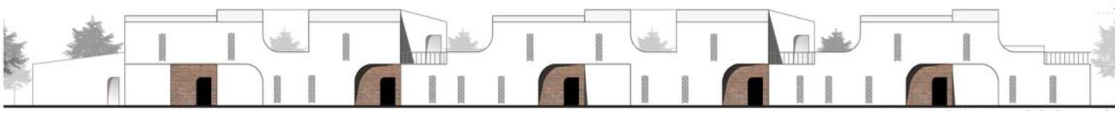


Figure 5. 15. The arches and the opennings on the units' façade.
(Source: Office A)

O_aC_1 explained the design idea which was initiate with a 'mathematical logic' and worked out in hand sketches. Following the logic, the coder applies a 'derivation method' and repeats the logic according to available tools in Grasshopper.The team was motivated to be explicit and legible about the design idea in creating and articulating a rule. O_aC_1 explained the reason of creating a rule in order to achieve unity in the form:

O_aC_1: … "Originally the goal is: to create all the rules on the smallest building block and then generate that unit and get the overall form. Why we do this: 1. To generate a 'regular' form, randomly generating one form, each different from the other, means that it does not have a unity. The mass formed at that point is a unit in itself. However, in each of our designs, we originally aimed to produce an overall mass subject to the rules with a repeating unit. 2. This is a competition, we have a very short time. Actually, in all projects, time is very short and producing a full unit of everything and generating this unit has always made our work easier. Which is actually a kind of mass customization, which has a counterpart in real life. The whole of the units following a rule relieves the whole process from project to manufacturing." [*manuscript by the coder (O_aC_1) on e-mail at Jan 3, 2019, 10:34 am*]

O_aC_1 explains the reasons for progressing in the design in a rule-based manner by highlighting two important points. Firstly, even though it is a competition project, it is impractical to propose completely different units when it comes to actual construction. She emphasizes that the competition process has limited time, and therefore, it is reasonable to proceed in a serial manner, aiming for mass production to reach a solution efficiently. This explanation together with the concept of legibility mentioned by O_aL, which refers to the legibilty and the ability of an architect to easily understand the design intent, explain why a rule-based design process is preferred. O_aC_1 initially approached the problem using a computational design approach and then completed the process using digital tools. In the presented case, the team leader (O_aL) and the coder (O_aC_1) worked in collaboration from the very initial steps of the design process. The team was motivated to apply computational design methods while creating design idea. Thus, each of the design idea was legible for the team to criticize and to develop it further.

5.1.2. The Municipality Building's Façade

In this section, a design episode, during when a formula generation in computational design tools is the dominnat activity, is presented from the municipality building. Office A participated in a municipality building competition, by using computational design tools to specifically generate the façade design of the building. For part of the façade, the office leader (O_aL) defined a set of rules to create a surface composed of glass bricks and brick (Figure 5. 16, a). In the team, O_aL and the coder (O_aC_2) sustained the transparency in idea sharing in the sketching phases with clearly identified prescriptions (Figure 5. 16). On the sketches, the percentage values in numerical format were sketched to describe the effect of the openings on the surface of the wall. Following the instructions (Figure 5. 16, a), then, it was O_aC_2's turn to process the sketch by resketching the idea (Figure 5. 16, b), to come up with a formula on Grasshopper (Figure 5. 17). Both the paper and the software were used simultaneously as the tools to translate a certain form of architectural knowledge (the wall) into mathematically represented one (percentage for gaps), then back into an architectural visual (perforated wall surface). O_aC_2 explained what the rules he applied in Grasshopper:

O_aC_2 : "The rule primarily prescribed the percentage of bricks in any particular section of the façade following percentages of 25, 50, and 75. In other words, out of the 100 bricks in a rectangular portion of the façade, either 25, or 50, or 75 bricks are subtracted creating an effect of dissolution of the solids. There was also a 90% section [the middle area], where 10% of the bricks were randomly eliminated. Since OaL's sketch didn't have a specific measurement for the area, I adapted them to the original size of each wall using her sketches while preserving the proportions. If the input code is 50%, we multiply the total number of bricks in that area by 1/2 and randomly select and delete them. The randomness in the selection already ensures that it's not an equal selection, making the process completely random. In the random selection option, there is a parameter called "seed" that creates variations, but even when the same number of bricks is selected, instead of having AABAA, for example, it becomes AAAAB". [*manuscript by the coder (O_aC_2) on e-mail at May 15, 2018, 5:58 PM*]

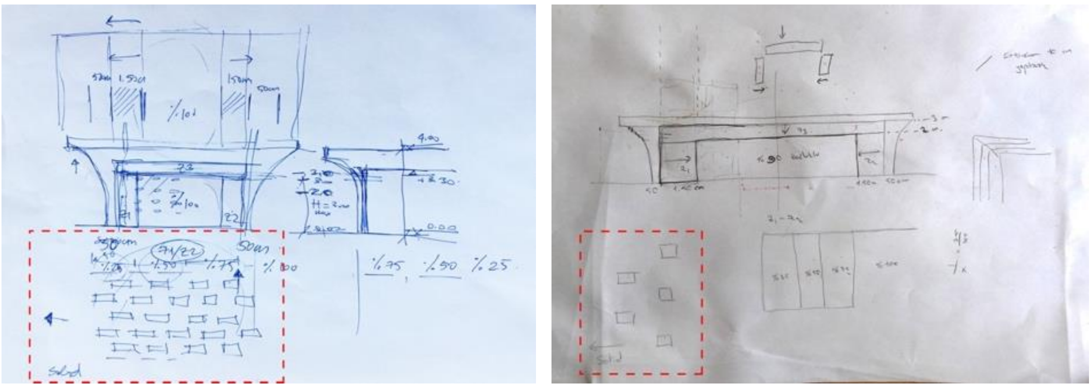


Figure 5. 16. (a. left) Descriptive sketches developed by the office leader; (b. right) resketching the idea before parametric definition by the coder (O_aC_2).
(Source: Office A)

In the meetings with O_aC_2, the office leader (O_aL) emphasized that with the glass bricks it is possible to get enough daylight. With the percentage values they initially determined, they aimed to arrive at a desired distribution of voids on the façade wherever it is needed. However, while the distribution of all these gaps contains a randomness, they need to also taking daylight inside at necessary parts, reflecting random glass brick orders. To have randomness, O_aC_2 needed to set parameters on Grasshopper. At the end of the design, the façade developed as in the Figure 5. 18.

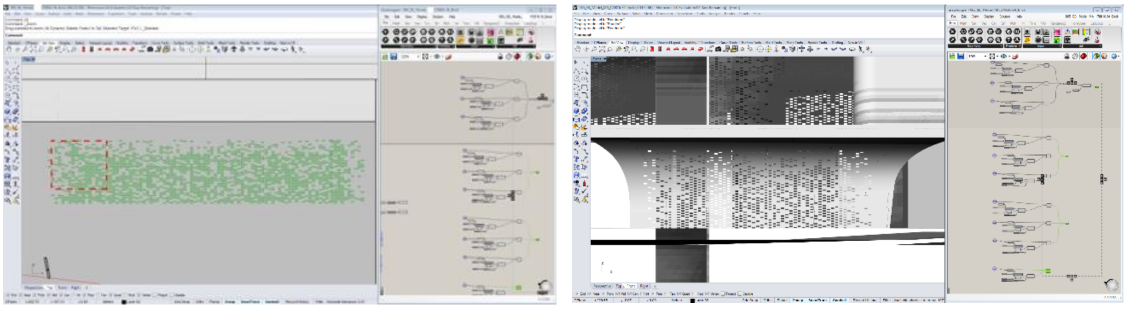


Figure 5. 17. Parametric definitions in Grasshopper developed by the coder (OaC2).
(Source: Office A)

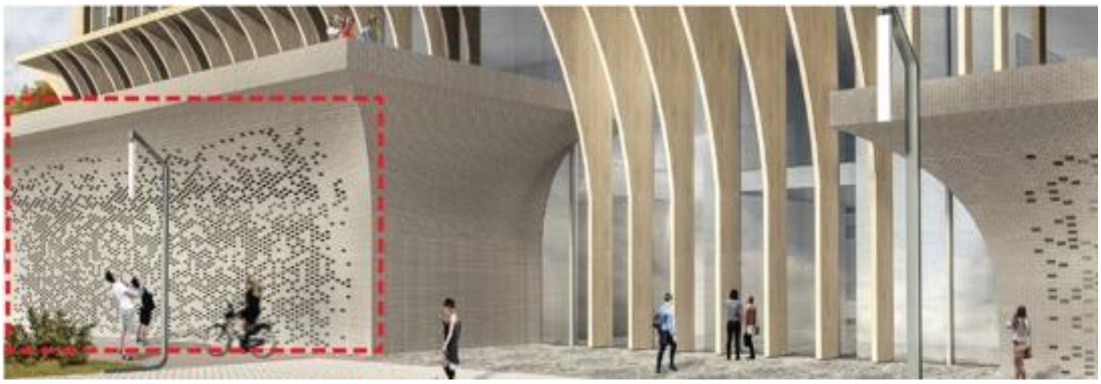


Figure 5. 18. Last version of the façade design. Drawn by the coder (O_aC_2).
(Source: Office A)

O_aC_2 explicitly set the parameters that explains density of the voids or glass bricks. However, the location of each void was not clearly defined in the computational formula and in the sketches that are developed by O_aL and O_aC_2. In the computational phase of scattering the voids, an expert eye could decide on the last version of the façade.

In this case, 'prepared eye' (Goldschmidt, 2015) of the team observed possible alternatives and selected the best idea for the design problem. 'Preparedness' can be related to the awareness of the explored alternatives. O_aC_2 mostly worked on the coding screen with numerical representations of computational tools. At the same time O_aC_2 and O_aL observed and monitored the visual representations of the rules that they input into the software on the 3D visualization screen.

5.2. Legibility to Track Design Process

This section presents an episode from the towers project of Office B. It highlights the office leader's criticisms of the design process during a meeting that took place just before Office B's first team leader was replaced by the second leader.

5.2.1. Towers Project

In this section, an episode from the towers project is discussed. In the tower project, two different team leaders sequentially led the coders in Office B. The first team leader (O_bTL_1) pursued a distinct formal approach to the search for form with the coders. However, during a meeting, O_bTL_1 reviewed everything the office leader (O_bL) team had produced so far, including sketches and 3D digital work (Figure 5. 19), and made a change. The new team leader (O_bTL_2), along with the team they were involved in, initiated the following dialogues:

00:00:01 O_bTL_2: "Let's go with a narrative, friends. Now we'll draw traces like these, let me quickly place them with a sketch."

The new team leader O_bTL_2 was trying to set the course of the design process and he was explaining them to the coder. At that moment O_bL jumped into the conversation and started to criticize the previous works:

00:00:05 O_bL: "There is an acceptance of being identifiable, for example, there is no analysis of the road, no analysis of a thing, I'm going interrupted..., and they are working on the building within that. These kids from the formalist Pinterest don't know where to start because."

Criticizing their formal approach, O_bL said he wanted to understand the origin of the form. Pinterest is a website to provide designers with an immense number of images, used mainly for accessing architectural precedents, and allows searching entirely on images that typically serves as a model for designers to consider in their ongoing design work. O_bL implies that the coders were only searching for visuals and were unnecessarily influenced by Pinterest images.

Figure 5. 19. The sketches and prints from the tower design process of the first team leader.

(Source: Office B)

O_bL state that a proper connection with the site was not established, and a completely independent design had emerged.

00:00:30 O_bL: “It's coming out like this wrong. The terrain doesn't create the ground floor thing, so these masses are forming, you know…”
00:00:50 O_bC_3: “Well, we actually worked on the form based on the existing condition.”
00:00:52 O_bL: “I don't see which existing condition you're talking about here! Look, I said once to show the evolution of it! You should say “here it is!”, then you jump to this, then you jump to that, then you jump to that. There is no such thing here, so the cause-effect relationship is broken!”

O_bL’s main emphasis was on the “cause-effect” relationship which he thought as key in generating the design solution. In this “cause-effect” relationship, O_bL was seeking the logic of the idea as a process that was clearly illustrated. Thus, the design solution could be legible for themselves and to the other designers. When designing with computational design tools, O_bC_2 used to apply an algorithm that would adapt to the entire design. However, the leaders wanted the design to be resolved in detail. For example, when the developed algorithm was applied to the entire facade, it should exhibit regional variations. Furthermore, O_bL added that design should be related to the environment, incorporating the following considerations:

00:01:20 O_bL: “Analysis, analysis, starting from scratch without any previous analyses, there are assumptions here. Because of how we shape these assumptions into a concrete form…”

During emphasizing the importance of analysis, O_bTL_2, the team leader, expresses his understanding of O_bL’s requests and explains how they will be implemented. At the same time, in the same conversation, O_bC_2 also states that they have already conducted the analyses.

00:01:36 O_bTL_2: “ Okay, as I mentioned, I will place these components in a certain direction, mentioning their positions and orientations. After that, I will ask you to lift them up accordingly.”
00:01:49 O_bC_2: “Well, actually, we have already made them.”
00:01:52 O_bL: “What have you done, man? Show me, show me what you've done. Where is it? What? I mean, I don't see anything here. I just talked a while ago, was it all pointless? I'm talking in vain, you're not listening to what I'm saying, and nothing means anything because it doesn't come from anywhere. Something should be based on something else. Look, down there, there are plenty of seashells. When seashells try to reach a necessary state, they hit a rock at some point, lose their edges, go sideways, and their tips get shattered. It's trying to move without being exposed to anything. I'm showing you this (Figure 5. 20). It should move within a cause-and-effect framework. Did you go to the right place? You're doing things without even seeing where you're going. I'm astonished, you're astonishing me.”

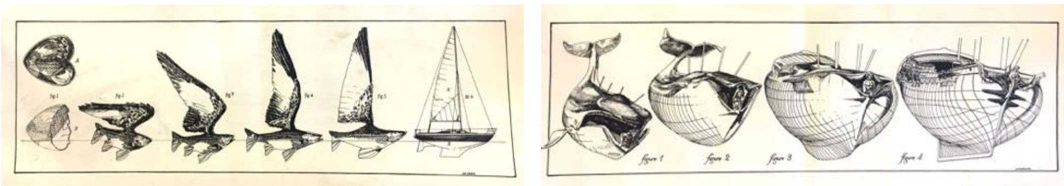


Figure 5. 20. O_bL shows these images to the team to express his thoughts about what evolutionary means to him.
(Source: Office B)

After conveying O_bL's requests for a form with a clear evolutional path, the team embarked on a design exploration considering the environmental factors through site analysis. As the floors of the towers rose, they introduced a twisting effect to establish a formal gesture that links the tower to the features of the site. Additionally, in order to document the process, they used Grasshopper with sliders to record each change they achieved, aiming to document the evolutionary process (Figure 5. 21).

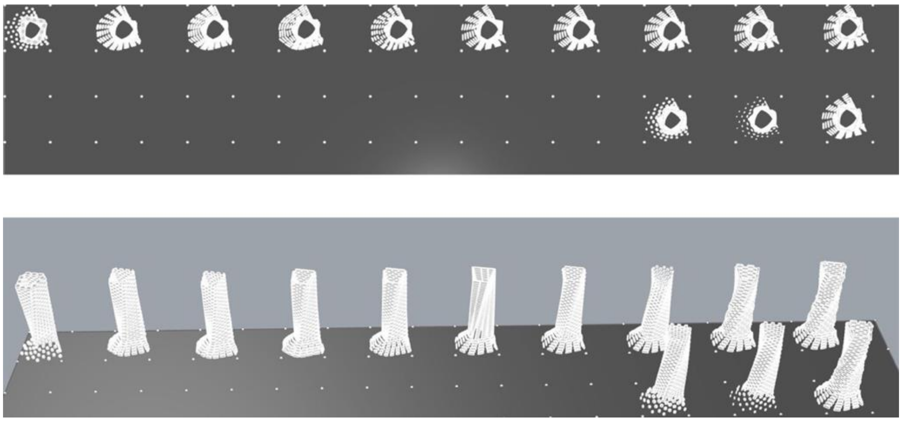


Figure 5. 21. Evolutionary process of the proposal. Developed by O_bC_2.
(Source: Office B)

Indeed, recording an evolutionary process can be closely associated with validating or legitimizing a design idea. By documenting the iterative changes and transformations through the evolutionary process, the team can showcase the thought process, experimentation, and development behind the design concept. This documentation serves as evidence of the design's evolution, helping to justify and support the design choices made throughout the process. For them, having a documented narrative adds transparency, credibility, and a level of rigor to the design approach, making it easier to communicate and gain acceptance for the final design solution.

5.4. Discussions

The observations above indicate that the teams mainly need to be explicit and legible in the design process in sustaining legibility among team participants and between the team and the other architects.

This chapter establishes the significance of the discussions on legibility in the sense of knowledge propagation within/among designer/s in the design process. Pursuant to observations and interviews, the findings demonstrate the concept of legibility in design as a pivotal issue in collaborative computational design practice. As offered by the qualitative analysis of observational notes and interviews, the term legibility emerged mainly around two routes which occasionally overlap in design discussions: (1) Concerning clarity in communication and articulation of design intentions, legibility was valued to make the developing morphology understandable to others; (2) legibility was also equally valued when designers want to explain and represent the logic of computational operations that govern the formal evolution of design. Briefly, the former involves reasoning about the formal qualities of the architectural product, whereas the latter is about making the formal evolution within the process legible to third parties.

Legibility is a term used in urban design and it is defined as the ability of an environment to be organized in such a way that it can form a recognizable and coherent pattern (Lynch, 1960). The main aim of designing a city as legible is to create a recognizable path for the users who experience the city from entrance to the end as a journey. In a similar manner, it was observed how the idea of legibility come into play as the individual or collective design work must be legible in every step of the design process. The rationale behind being legible in a distributed team dwells on the justification of design intentions for oneself and for the other participants. Legibility is necessary in every level of architectural design process, such as, understanding existing situation of the design area, understanding of design idea in the design process, clarity among team participants, communication between the team and consultants in the design process and construction process, and legibility of the design idea for the users. In the Figure 5. 22, the relation of an explicit logic and the design product is illustrated.

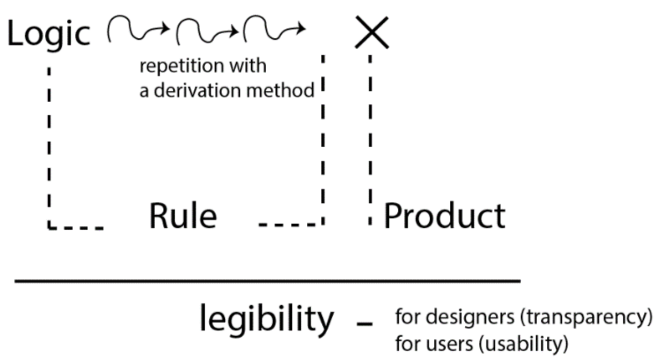


Figure 5. 22. An illustration of legibility in a rule-based design.
(Source: The diagram was drawn by the author)

Design is a creative process and the relationship between creativity and idea diversity is strongly emphasized (Guilford, 1973). In design teams, individuals seek legibility to support a productive communication within an ambiguous creative setting for which divergence is often desired. To be legible, the team participants used sketches on paperas a tool to generate and translate ideas. The team participants aimed to be more cognizable by making sketches either to understand their own thoughts, or to transfer them to the other participants. For example, in the sketches produced by O_aC_2, the main idea was conveyed on a paper, and then the parametric design tool was used to produce different alternatives in line with that main idea in a way to expand the exploration space. Thus, initial sketches become a legible guide for the coder. On the other hand, O_aL developed a sketch to express the idea to O_aC_2. Then, the coder re-sketched all of what he heard during the drawing phase from the team leader and what he saw on the sketch developed by the team leader. Re-sketching appears to work as a legibility tool. Coder performed the re-sketching step as a synthesis phase of the initial sketch, thus he outlined Grasshopper definition.

To have a better understanding and coordination in the design process, the team needs to increase awareness in the system. Moreover, to support collaborative work in a system, transparency of tasks and operations are necessary. Heerwagen et al. (2004) describe the awareness as 'back-channel information' in collaborative work. Each of the participants' high awareness is important for dynamic undertaking of tasks by different team members. In the distributed work environment, tasks are coordinated dynamically to sustain the process. Dynamism is highly needed together with transparency in any operation. Hutchins (1995) mentions 'dynamic tasks' in the system to tackle with a problem dynamically because of 'overlapping knowledge' of the team participants. The office leader (O_aL) and the coder (O_aC_2) kept the transparency in the sketch phases with highly described recipes (Figure 5. 16, a). The team leader draws sketches with numerical percentage values to describe the effect of the openings on the wall design. Then, O_aC_2 re-sketches the idea (Figure 5. 16, b) and applies it as a formula on Grasshopper (Figure 5. 17). In the coding stages, the team leader was periodically checking the process on the Grasshopper. So, mathematical expressions and drawings were used interactively between the team leader and the coder. O_aL and O_aC_2 share overlapping knowledge domains so, O_aL could easily follow Grasshopper process.

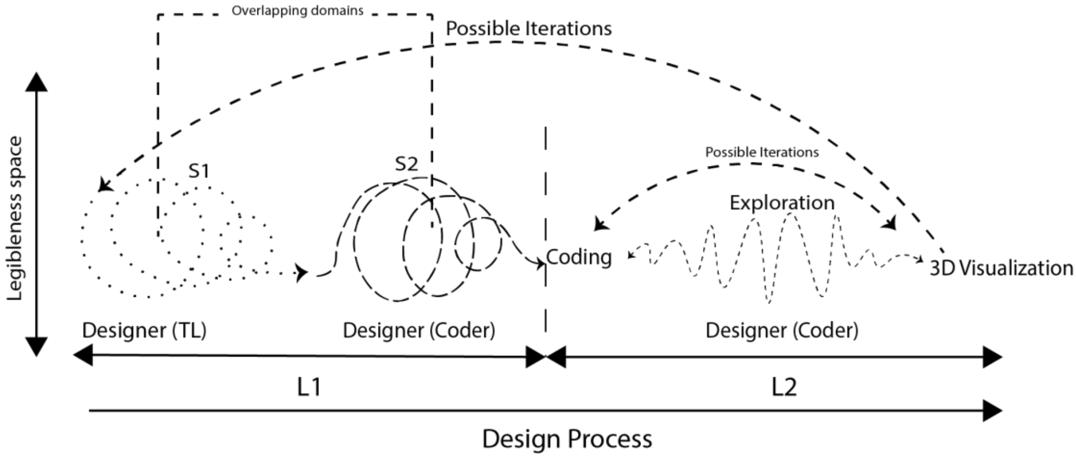


Figure 5. 23. An illustration of legibility space in the design process. L1: Legible production in initial design idea, L2: Legibile production in computational tool, S1: Sketching by office leader, S2: sketching by coder.
(Source: The diagram was drawn by the author)

Interaction and communication within teams, by its nature, need to be as explicit as possible in the complex nature of the design process, as it is suggested by Sawyer (2017) while explaining the creativity within imporivisation teams in theather. Sawyer states that the first prequisite for creativity in improvision is the openness to listen and hear your team members, which he calls deep listening, together with building on their ideas (p.37). In the above cases, this was sustained through a search for legibility. Yet as Sawyer (2017) suggests creativity can never be predicted nor planned, because if it could have been than the outcome would be expected therefore not creative, and that it can only emerge. Based on innovation studies in big industries, Sawyer argues that companies only realize the significance of an innovation and how it happened only after it has emerged. If so how could creativity within teams be ensured and enhanced? Sawyer suggests that this is a double-edged sword that the teams need to keep a close eye on keeping planning, in the studied cases legilibity and rule-based design descriptions, versus improvisation, in the studied cases, serendipity. His proposal indicates an interactive iterative process between episodes of planning and improvisation stating that:

> "The improvisational teams didn't exactly wing it, either. They engaged in short bursts of planning that alternated with improvisation; in other words, they distributed design activities throughout the execution process… The key to innovation is always to manage a subtle balance of planning, structure, and improvisation" (Sawyer, 2017: p.41).

In an improvised collobaration sustained with legible rules, team participants need to be legible to conduct a fluid process. As Sawyer (2017) claims, when there are no clear objectives team members of an organization are lost:

> "The group members then are more likely to be in flow while working toward such a goal if they've worked together before, if they share much of the same knowledge and assumptions, and if they have a compelling vision and a shared mission. One study of more than five hundred professionals and managers in thirty companies found that unclear objectives became the biggest barrier to effective team performance" (Sawyer, 2017: p.56).

Clearly, in the cases studied in this chapter, the clear objectives set by the office leaders directed the teams' course throughout the design explorations. Yet, as also suggested by Sawyer, one must make sure that there is room for improvisation as such that the designers could opportunistically benefit from emerging ideas, which requires an openness while staying on a track. "The key question facing groups that have to innovate is finding just the right amount of structure to support improvisation, but not so much structure that it smothers creativity" (Sawyer, 2017: p. 68).

CHAPTER 6

COLLABORATIVE PRACTICE

This chapter provides ‘thick descriptions’ of the events that surrounded the design process of the design projects which are the municipality building project, and the tower project, during the design development processes (Figure 6. 1 and 6. 2).

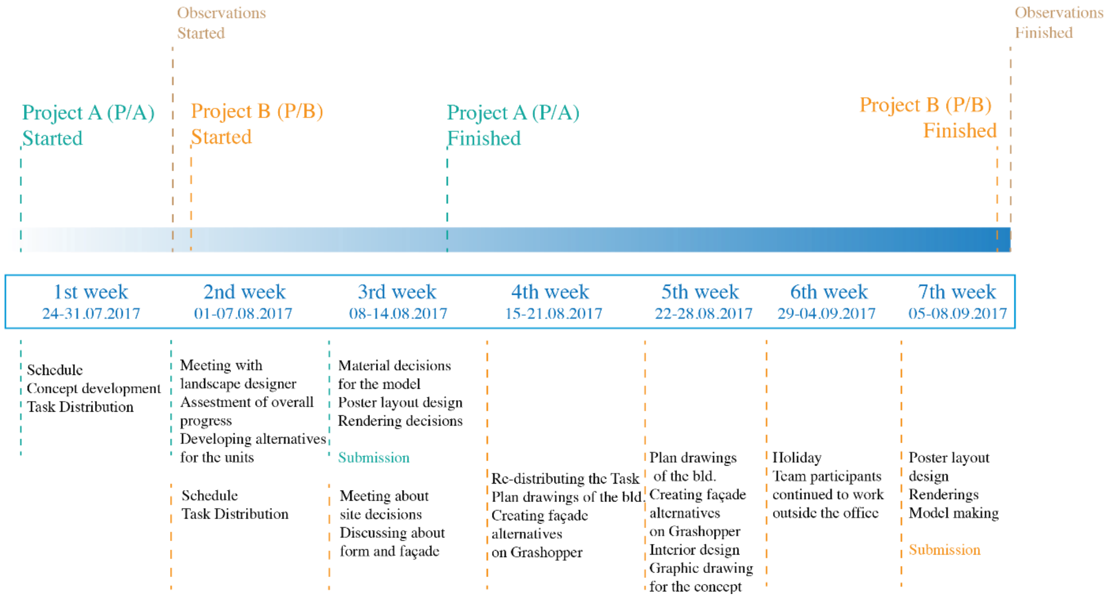


Figure 6. 1. Timeline for the youth center design project and the municipilaty building design project of the Office A.

(Source: The diagram was drawn by the author)

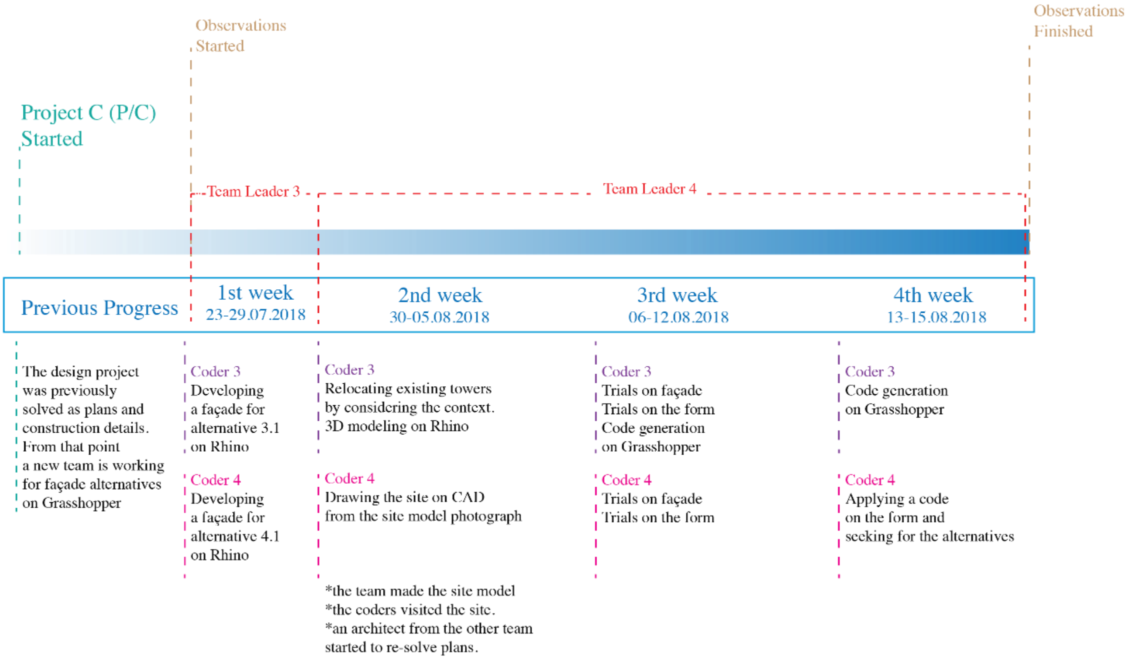


Figure 6. 2. Timeline for the towers design project by the two different team leaders in the Office B.

(Source: The diagram was drawn by the author)

Moreover, the library project is presented which is retrospective research. The main body of the data set used in this study includes documents from the design idea search and creation phases, which span over a period for a month for each office. The data set was extended to include earlier background and the archives of the offices to be able to describe events from a broader perspective. The sub-sections of this chapter follow a chronological order to account for each projects' events as they emerged during the design processes. The intentions and ideas of the design projects will be described. These collaborative environments, where different disciplines were brought together in design research, were a design process in which multiple participants contributed and their knowledge and experience overlapped. In the teams, there were office leaders as the final decision-makers, team leaders who controlled the operation of the teams and were involved in the project at the same time, and coders who did both the design and scripting work (Table 6. 1).

Table 6. 1 Key participants in the teams.

(Source: The table is drawn by the author)

Office A	Project A (P/A)	O_aL O_aC_1 O_aIA_1	Office Leader Team Leader/ Coder Intern Architect
	Project B (P/B)	O_aL O_aTL_2 O_bC_3	Office Leader Team Leader Intern Architect/ Coder
Office B	Project C (P/C)	O_bL O_bTL_1 O_bTL_2 O_bC_2 O_bC_3	Office Leader Team Leader Team Leader Architect/ Coder Architect/ Coder
	Project D (P/D)	O_bL O_bTL_2 O_bC_1 Consultants	Office Leader Team Leader Architect/ Coder Architects, Statics Engineers

In this chapter, the research focuses on how does the distributed cognitive system support for multidisciplinary collaboration in design teams and how team participants cope with different languages of different disciplines. The research is also focused on how different knowledge domains are propagated among team participants. In this research, the teams were constituted by architecture major but also different background of disciplines which is computation. Each of the teams have participants who navigate computational tools in design problem solving.

Distributed cognition is a theoretical concept that allows for the investigation of collaborative work systems and provides a way to explore the context in which they operate. The most comprehensive model of distributed cognition focuses on how cognitive tasks are organized in intricate environments, particularly exemplified in the navigation of sizable ships (Hutchins, 1995a). When utilizing the distributed cognition framework, we can draw comparisons between design processes and navigation, creating a metaphor where designers 'navigate' through a design space using various tools. Collaborative behavior is facilitated through socially established channels of interaction within a predefined, yet flexible, organizational structure.

This chapter describes some of the types of collaboration that created among the different disciplinary participants in the study. In the collaborative works, two main collaboration types were observed: (i) designers in both roles: *as a coder and a designer*; (ii) designer/coders as appliers: *coders as solution applier* (a-Sketch to Grasshopper, b-Mathematica to Rhino). Specific episodes are presented and discussed in each category to describe the work environments of the observed teams. These episodes highlight issues such as thick descriptions of multidisciplinary work environment of architecture professions in offices, knowledge propagation between different disciplines, interaction among multidisciplinary team participants, and coordination of different hierarchical systems.

The following sections provide segments from the dialogues of participants from different teams working collaboratively. The presented dialogues represent the design processes of team participants who are shared the same process with having common knowledge areas but are also fed by different disciplines. The next section provide a thick description about a multi-disciplinary work environment of the design process in the observed teams.

6.1. Propagation of knowledge in design process

In this section two different types pf the collaboration between coders and architects are presented. In the first section, "*designerscripters*" (Burry, 2011) section is present Office B's the towers' façade design. The section highlights how the main decisions made by the team leader during the design process, the problems that arise while implementing them, and the way coders are handling these problems. The problems encountered in the process and how the team was in communication on the way to the goal were presented in this section with the dialogues of the team during the design process.

In the second section, "designers as scripters" section is present two projects: first part is Office A's Municipality Building design and second part is Office B's the Library project. The section presents the situations where the designers take part in the team to implement the design idea are presented.

6.1.1. 'designerscripters'

In this section, the observed teams have their collaborative work and sharing environments where each participant was an architect, but some participants also performed other professions such as coding, graphic designing, etc. In the distribution of roles in the teams, some participants had to fulfill more than one discipline. In the teams, the newly graduated architects and/or internship architects who knew coding were also responsible to find the ideal design solution while they were responsible coding/scripting for the exploration of possible design alternatives. In this section, in the field observations, the roles and collaboration mechanisms of the team participants who take more than one role/responsibility in the design process in the teams are presented. Following part presents a segment from the collaborative work process of the office B's team.

6.1.1.1. Towers Design Project

In the observations of the office B, the team leaders (O_bTL_1 and O_bTL_2) were not capable the scripting languages but have adopted computational design approach into their design approaches. In the following dialogues, the moment of work between two newly graduated architect-coders (O_bC_2 and O_bC_3) and the team leader (O_bTL_1) is transcribed. These dialogues are typical of a communication among the team participants that is frequently encountered in the research in the observations made. The following transcripts are taken from a meeting for the tower project design process. The working environment of the team was located on the mezzanine floor of the office, accessible to other work teams in terms of visibility and auditory, and in a position where all of the teams were aware of the presence of the office leader downstairs at any moment. The team was working around a shared table so the participants were communicating at any moment easily (Figure 6. 3).

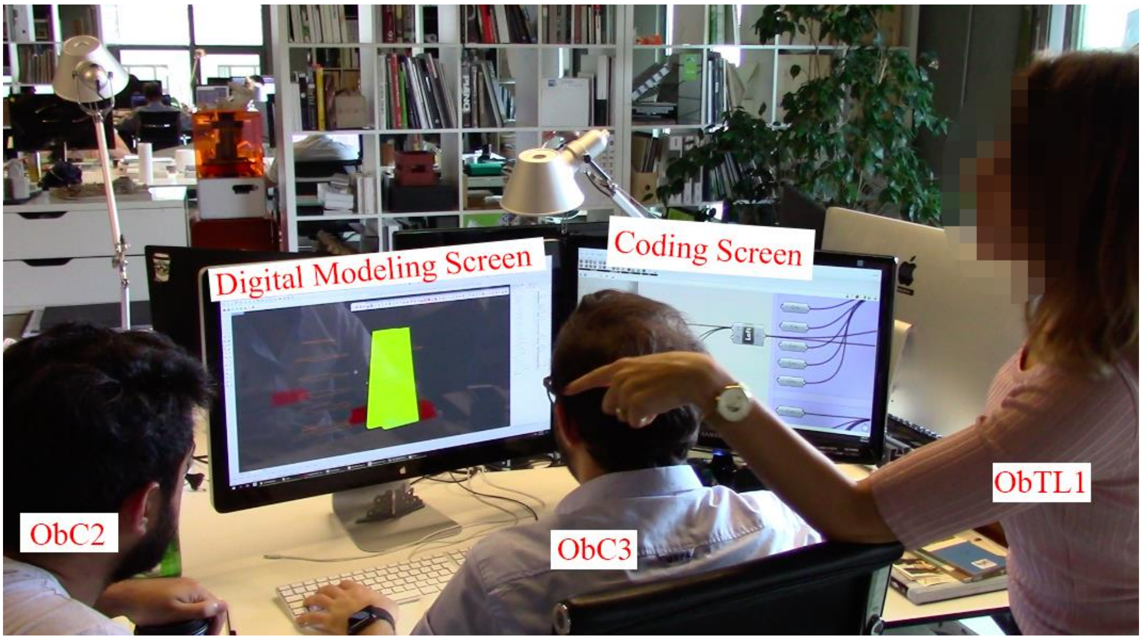


Figure 6. 3. O_bTL_1 is using gestures and body language to the coders while working collaboratively on the design project.
(Source: The photograph was taken by the author)

In the first day of the design process, the team leader (O_bTL_1) assigned different alternatives to each of the coders. O_bTL_1 showed different precedents, developed different sketches, and discussed each alternative separately to each of the coders. O_bC_3 had to develop a form that linked with the ground by lowering a smooth surface vertically from above to the ground of the tower. O_bC_2, on the other hand, after dividing the plan schema into two and combining it with a core, aimed to create a smooth surface by considering the cracks formed in the façade.

In the design process, both of the coders were working separately but, time to time the team was discussing about each one's work and evaluating together. In the following dialogs, one of the random meetings were transcribed. Before the recording was started, the coders were already discussing about the software and sharing their experiences. The team gathered around O_bC_2's computer's screens (two different monitors: Grasshopper and Rhino) to evaluate his design alternative. The coders started discussions about technical way of generating the targeted shape in Grasshopper + Rhino. O_bC_2 was trying to produce a 'network surface' on the façade to have more smooth surfaces in Rhino.

00:02 O_bC_2: “I want a network surface in the network (3D modeling), polyline or line?...”
00:19 O_bC_3: “Polyline. Exactly, take as a reference completely one of them. Only the finishing is matters. Where it ends, for example, in a way that retains that line…”
00:26 O_bC_2: “a line? there is already a line.”
00:29 O_bC_3: “Look, now, click there, press F10. it's curvilinear there.”

In this segment of the design process, the coders were dealing with Rhino. These dialogs have not relation with architecture discipline. It is impossible for someone who is not familiar with digital design tools to understand these dialogues. Until O_bTL_1 attends the meeting, the coders share among themselves the actions they have trouble performing while working with the software and jointly try to find solutions. While O_bTL_1 was conducting the tower design process, O_bC_2 and O_bC_3 were sharing knowledge with each other, but this sharing was more like competing to prove who had how much design computation knowledge. As a daily morning routine, both of the coders were exploring new plug-ins in online data bases, they were trying to enlarge their plug-in library in Grasshopper. Although the coders were not in a competition with each other, it was possible to observe as in the dialogues that there was an atmosphere of proving that ‘knowing coding better' between the coders.

(*O_bTL_1 participates the meeting*)

00:58 O_bTL_1: “I think so... Actually, I don't think there's a need for a second, third move, ObC2.”
01:07 O_bC_2: “Second, third?”
01:08 O_bTL_1: “Look at this movement, not like the example I showed.”
01:10 O_bC_3: “We're trying to fix it right now.”
01:12 O_bTL_1: “well, ok.”
01:13 O_bC_3: “Because it’s doing a camber right now...”
01:16 O_bTL1: “We need to get rid of the camber on both sides.”

O_bTL_1, while criticizing the unnecessary ‘camber’ in the form (Figure 6. 5), referred to the precedents she had previously given to the coders. Therefore, communication within the team was continued through visual elements such as 3D models, 3D digital visualizations and sketches. O_bTL_1 does not know what caused the ‘camber’ problem and asked the reason of the camber form and discussed with the coders. O_bTL_1 handled the camber problem as a design problem and the solution was again handled as design. But the problem was purely a technical one. O_bC_2 had a problem with the software while working on the form, he could not intervene enough. At this stage, the O_bC_2 explained it:

01:19 O_bC_2: "Do you know why it's happening?"
01:20 O_bTL_1: "Why?"
01:21 O_bC_2: "The inclination wants to come up" (Figure 6. 4).
01:23 O_bTL_1: "Hmm."
01:24 O_bC_2: "To make it go like this, here the belt is giving in, so that it can come out and come in."

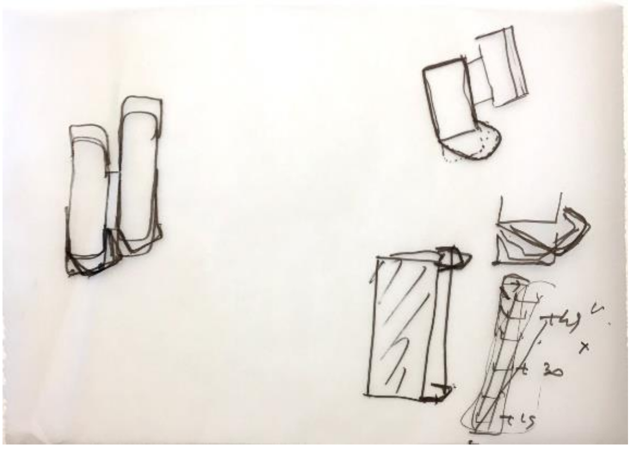


Figure 6. 4. Sketches on the inclination calculations and experiments of the search for the form of the towers, by ObC2.

(Source: Office B)

Figure 6. 5. The team is discussing about the form of 'camber'.

(Source: The photograph was taken by the author)

In the segment presented above, among the geometric forms formed as a plan in the horizontal plane, the coders are trying to create a surface towards the vertical. In this way the façade would be created. The interesting part on this process is, while the coders (O_bC_2, O_bC_3) obtained a surface by combining with a third element vertically of two basic geometric forms created in the software, as much as same flexibly when they creating design solutions by hand drawings. In the dialogues presented previously, the coders started to express that they thought the control was in the software, with phrases like "it wants to come up" and similar expressions in the design process.

During the design process of the tower project, the coders tried to gain mastery over the software used while trying to focus on the design problem. While the coders were dealing with the task of the design problem, they were creating some temporary interventional elements. For example, they may actually need to produce some basic elements that will not remain at the end of the work on 3D. However, the team leader interferes by thinking that these elements will be permanent. There have always been such disagreements in the collaborative design processes in architecture, but the disagreement in the dialogues above is not about the design method or design approach, it is much about the expertise that requires to know how to use different tools such as computational design tools. Meanwhile, the two coders try to eliminate the problem by transferring their knowledge to each other. At the beginning of this dialogs, while the coders were communicating on the computer screen through both visual (Rhino) and computational (Grasshopper) representations, O_bTL_1 was only discussing the situation through visual representations (Rhino). O_bTL_1 described the problem on the visual and asked for it to be edited. However, since they could not reach a solution, she began to voice only her demands:

01:31 O_bTL_1: "Ok. I think you don't do it at all. I assume you offset 3 meters and never touched what's inside. you upgraded here 3, you got 3 out of here. When you tie it up with a straight line like that, I'm not actually giving you two meters here, two fifty, two twenty, anyway.... I'm not giving you two meters."

01:47 O_bC_2: "Is that with this only?"

01:49 O_bTL_1: "Yes, exactly. With this one at the bottom and the one above. So, you made one a slight offset you left it.... the form that comes out when you connect it with this. I think it's true that it's such an example that this is another design. It can be done, too. But that's where we set out for the first time."

When O_bTL_1 was stating her demands for the design problem, she gave some presumptive numeric values on the visual model and said these values for the interventions to be made on the digital model. However, the coders could make these modifications in two ways: visual or computational. As a method for design problem solving, the coder adjusted some values through the computational tool and could easily change these values at any time. For example, floor highness or floor area square meter values were added as parameters in Grasshopper so, any size changes can easily be updated via software. But, the form of the towers was not adjusted in the computational at that level of the design process. Therefore, according to the situation to be negotiated, the coders were involving in the process through visual (Rhino) or computational (Grasshopper) tools. The dialogs were followed:

02:04 O_bC_2: "can you show it?"
02:06 O_bTL_1: "You just did it, I was okay. Why we went back?"
02:12 O_bC_2: "Because I hadn't done it this far, I've done it this far." (C4 shows middle parts of the tower, by controlling Grasshopper- Since the floor is getting narrower at the top of the tower, the façade is making a camber)
02:16 O_bTL_1: "Alright then…"
02:17 O_bC_2: "I move it… that's what happens."

The stage that O_bTL_1 saw the 3D visualization screen of the O_bC_2's design alternative was not yet completely applied all the façade. In the dialogs above, O_bTL_1 and O_bC_2 discussing about the version of the façade that $O_bTL_1$was not expecting. In the beginning, O_bTL_1 previsioned differently what came out at the end. To explain the reason for the unexpected façade design, O_bC_2 tried to show it by changing the parameters on the Grasshopper plug-in. So, when O_bC_2 tried to implement all of the front, he made the team look at the situation that arose. In this way, all of the team participants could involve the computational process by observing the tool.

02:20 O_bTL_1: "Okay then look, I'll tell you again. you know, C3 also said either 1,2,3,4. I don't need two connections from this point I draw a line from beginning to end and connect it."
02:29 O_bC_2: "yes yes, there are no such things, these are only two."
02:32 O_bTL_1: "But when you draw it right now to the right spot…"
02:34 O_bC_2: "But that's what happens when we add that to that."
02:37 O_bC_3: "No, you could not understand each other…"

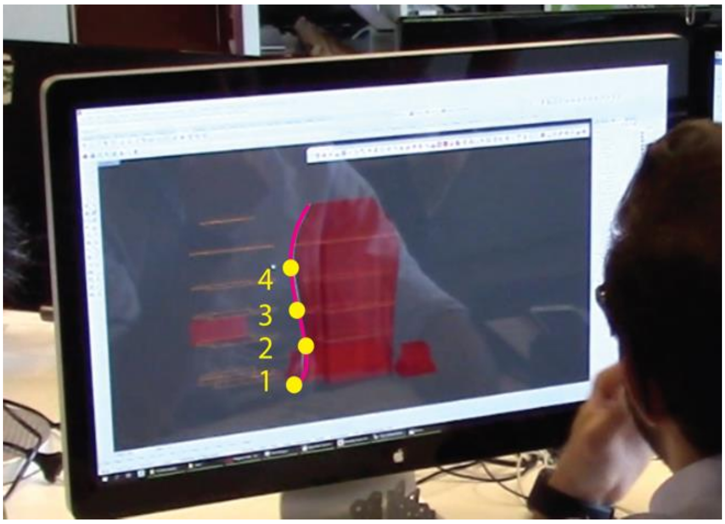


Figure 6. 6. The unexpected curve on the façade for O_bTL_1 (Yellow dots and numbers added by the author.)
(Source: The photograph was taken by the author)

O_bC_2 set a reference line based on the expansions and contractions in the floor area of the floors in Grasshopper. Since the specified reference line moves in as S shape, the surface also fluctuates relative to the S line. On the other hand, O_bTL_1 aims the façade as connected surface from the ground and the top floor without connecting multiple points on the tower (Figure 6. 6). After this phase, O_bTL_1 starts to step in the way O_bC_2's working method on the computer. O_bTL_1 began to describe on the screen which Rhino is opened, showing which points need to connected with a line and how to bring out a surface for the façade. First, O_bTL_1 wants O_bC_2 to draw a line as a reference from bottom to top of the tower. But it should be a linear line that goes directly to the top (Figure 6. 7). Previously, O_bC_2 had adapted a reference line to the enlargements and narrowing of the floors and caused a waved surface on the façade (Figure 6. 6).

02:42 O_bTL_1: “Okay, but I'm telling you that...”
02:43 O_bC_2: “I say the difference, the difference.”
02:44 O_bTL_1: “But I don't want to touch these points, I say here and there.”
02:47 O_bC_2: “Actually, they don't exist.”
02:48 O_bTL_1: “ok.”
02:49 O_bC_2: “I'm lifting this.”
02:50 O_bTL_1: “Okay, okay. why did it happen like this? it shouldn't be.”
03:06 O_bC_2: “Because it is hiding in here.”
03:10 O_bC_3: “Because of the way it surfacing. I mean, something that originated from the software.”

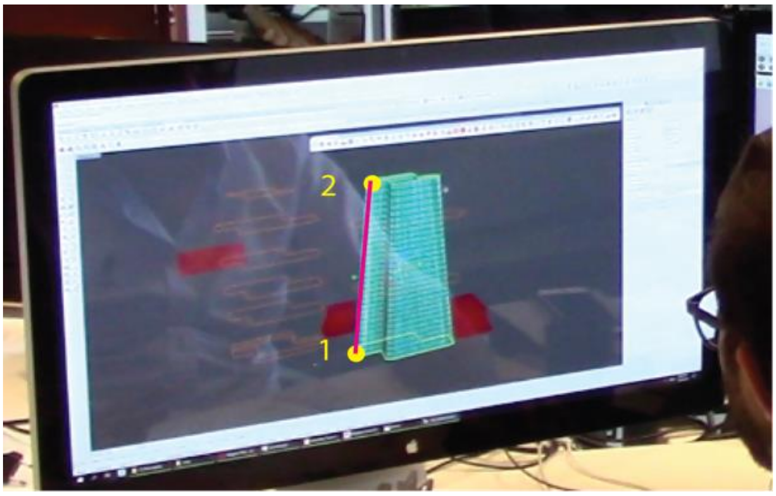


Figure 6. 7. O_bTL_1's suggestion for the façade crated by creating a line that connects only two points. (Yellow dots and numbers added by the author.)
(Source: The photograph was taken by the author)

03:13 O_bC2: "To have a smooth surface, it has to go like this."[*gesturing with hands*]

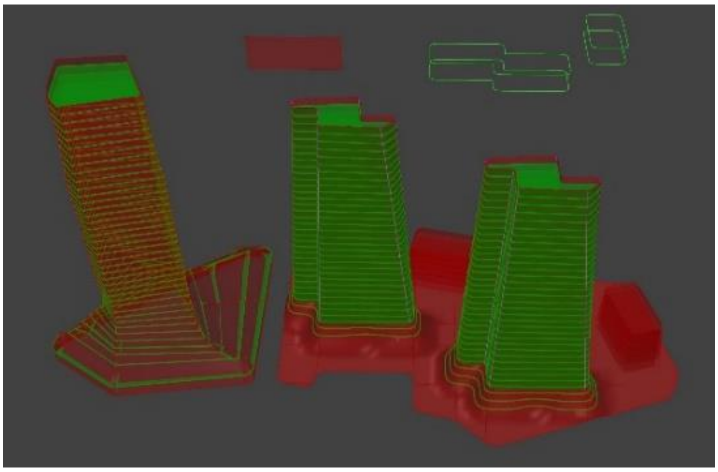


Figure 6. 8. Trials on the façade at Rhino. Work on the plans of the towers before implementing the façade, by O_bC_3.
(Source: Office B)

Until this phase, O_bC_2 had been partially executing the design with the parameters it had determined through Grasshopper. However, after the intervention of O_bTL_1, O_bC_2 continued with digital modeling on Rhino.

At this phase of the collaborative design process, the disagreement between O_bTL_1 and O_bC_2 was due precisely to interdisciplinary differences. O_bC_2 is a novice architect and expert in coding. However, the curvilinear surfaces that are too often paired with the 'parametric' design seem to have impressed O_bC_2's design approach. There was an effort to obtain a curvilinear surface on the façade of the towers, an attempt to obtain an ‘iconic’ design. However, in a residential structure, according to O_bTL_1, 'formal' movements of the towers should follow the functions of the building which is a residential.

The dialogues in the above are a situation in which two architects who know how to design with different methods and fail to agree, although the knowledge of architecture is common. O_bTL_1 and O_bC_2 assess the digital design tools on different screens so, they design in different methods. O_bTL_1 knows how digital modeling works, but she doesn't know how computational tool controls the design process. Hence, O_bTL_1 manually interfering the digital modeling part on Rhino screen, she disables the computational tool and asks for changes as manually. On the other hand, the changes that O_bTL_1 asked to done, O_bC_2 cannot make through the computational tool, he controlled through digital modeling (Rhino). So, O_bC_2 controls the multiple digital tools in a hybrid way, using digital modeling and computational design. O_bC_3 and O_bC_2 not only worked on the digital tools as hybrid, but also worked in a hybrid way between the disciplines of architecture and scripting. The coders’ roles were transformation of the knowledge between architecture and scripting disciplines. The communication among the team was sustained through digital modeling screen which is visual representation.

After a discussion period on the design idea, O_bTL_1 explained her hesitation on curvilinear façade because of ineffective interior parts of residential building. O_bTL_1 stated as:

05:52 O_bTL_1: “Now, I think something is misleading us. You know... Okay, we have masses in our hands, we don't necessarily try to take them immediately and turn them over and make an extra form. We're taking the design a little bit further to another dimension. Okay, we made a move in the hotel, here will be a little more iconic building, I don't know but, we are currently building a residence. The extra move we're going to make to that residence is what we're building on top of the current situation... Of course, our goal can be the view. Again, we have a budget issue. Orientation may be needed maybe a little terrace balcony garden in front of all of them anyway... we can do something, that's our goal. Otherwise, we don't make statues everywhere.”

In the next phase of the design, the team worked on to find the way how to get a façade through plan. O_bTL_1 and the coders aimed to provide a rotating movement between the bottom and the top by making a twist (Figures 6.9 and 6.10). O_bTL_1 aimed to provide positioning to the scenery view with this rotating movement on the tower's façade.

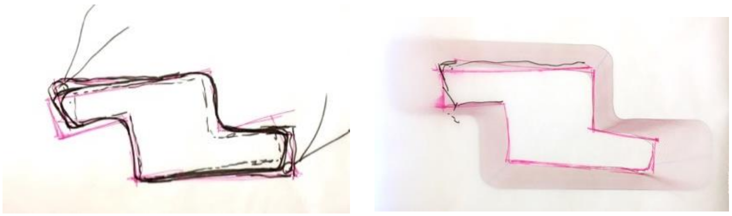


Figure 6. 9 (top) Twisting the towers via rotating the floors. Pink and black ink drawings are representing the top and the bottom floor area. A sketch by O_bTL_1 (pink ink) and O_bC_3 (black ink), produced as collaboratively. (bottom) avoiding the twist and continuing with creating a surface from the ground to top.
(Source: Office B)

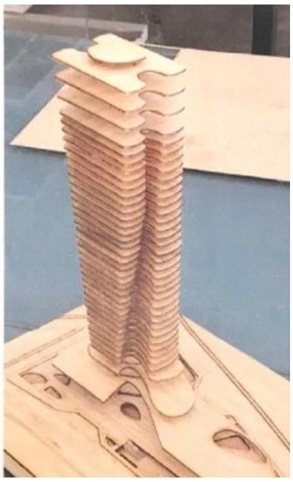


Figure 6. 10. A precedent for the towers, referred by O_bTL_1. The precedent is a tower project which has been done previously by the same office leader but different team participants.
(Source: Office B)

As a routine practice in the office, Team B's office leader O_bL checks in with the team participants in the afternoons, and evaluates what is on the computer screen. In the segment presented here, O_bL was only checking the end of the day outcomes so, he was not involved in each and every step of the coding and visualizing. O_bL has some understanding of how coding comes into play in architectural design but has no experience in practical means of creating codes and scripts. As a consequence, O_bL had to develop alternative methods to follow and guide the design phase by way of assessing the design schemes on the computer screen or on print-outs. At one of these segments, O_bL, frustrated with the design alternatives presented to him, expressed that he was unable to follow the design rationale behind these schemes and how they evolved into their final forms. O_bL stated as follows:

0:00:52 O_bL: “show the evolution of it! you should say “here it is!”, then you jump to this, then you jump to that, then you jump to that. There is no such thing here, so the cause-effect relationship is broken!”

At the end of the design process, the alternative produced by O_bC_2 was not approved by O_bL. Finding the towers' relationship to the base on the ground to be “forced”, O_bL decided to continue with a new team leader.

6.1.2. Summary

The team described above was in search of a new façade and form to the towers project. An office leader, a team leader, and two coders (designer-scripters) worked together as a team. The office leader (O_bL) did not follow the progress of the project moment by moment but visited the team at regularly such as every evening. The team leader (O_bTL_1) assigned the coders with two different alternatives and followed them more frequently than O_bL. $O_bTL_1$has situated at the same table and able to intervene while the coders were working on the project. Also, because the coders worked at the same table, they could ask each other questions at any time. Even though other participants of the office were not included in the team C, the coders could easily communicate with other office employees around the table, and the coders, who were junior designers, were able to easily access the experience and experience of other participants of the office. The accessibility of the previous experiences were not limited with employees' experiences, the team has also access to visual elements from the previous projects of the office.

The team communicated through on digital modeling tools and sketching which are shared knowledge domains. Although the digital modeling tool is commonly known, disagreements have arisen due to differences in the way participants use it.

Table 6. 2 Phases of the towers project, in collaboration of the team.
(Source: The table is drawn by the author)

Towers Project	Planning Day 1	Co-creation Day 1	Co-creation Day 2	Co-creation Day 3	Co-creation Day 4
Level 1 (Initial design phase)	O_bTL_1 assigned the roles for O_bC_2 and O_bC_3 (2 short meetings)	O_bC_2 and O_bC_3 made researches to find the way to code the form (instant 3 meetings while working)	-	O_bC_3 continues his researches for codes	-
Level 2 (Idea Generation Phase)	-	-	O_bC_2 and O_bC_3 working with codes to develop initial design idea (instant 2 meetings while working)	-	-
Level 3 (Idea and Knowledge Sharing Phases)	-	-	O_bC_2 and O_bC_3 started to share coding knowledge when they struggled. (instant communications)	-	-
Level 4 (Discussions)	-	-	-	O_bTL_1 and O_bL evaluated the current situation. (one meeting)	O_bL discuss with the team and decide to change the team leader.

6.2. Designers as Scripters

In this section, the observed teams were constituted by a coder/designer and an office leader where working close collaboration. The role distribution was defined by the office leaders. In the teams, the newly graduated architects and/or internship architects who knew coding were responsible to apply the solutions that developed by the office leaders. The coders, who are also architects, had role mainly about coding and software part, rather than designing. However, the coders were dealing with two different situations: one is clearly defined geometric forms and formulas by the office leaders, second is a form that imagined previously by the office leaders and finding the way of application the form in a software.

In this section, in the field observations, the roles and collaboration mechanisms of the team participants are presented. Following part presents a segment from the collaborative work process of the office A’s the Municipality building’s façade design process.

6.2.1. The Municipality Building Project

In this section, the collaboration of the Office A's participants is presented. The presented episode is from the dialogs of the municipality building project’s meetings between the office leader (O_aL) and the coder (O_aC_2). In addition to having one-on-one spontaneous meetings with the O_aL and O_aC_2 in the observations, they also held meetings involving the entire team. O_aL and O_aC_2 worked in constant interaction over the course of the observed episodes in the office. As a vivid example of those instances, here, a detailed account of the efforts of the team at generating the façade of the municipality service building is given. O_bTL_2 and other participants of the team took part in the form and program of the building, while O_aC_2 was only responsible for the pattern planned to be created on the façade. Although O_aL distributed roles within the team, O_bTL_2 seemed to be in charge of part of the team. For example, O_aC_2 communicates directly with O_aL, while O_aIA_4, O_aIA_5, and O_aIA_7 first get O_aTL_2's approval and then relay developments to

O_aL in joint meetings. In the Figure 6. 11, O_aL, O_aC_2, and O_bTL_2 are discussing at one of the regular meetings in the office (Figure 6. 11). The team was working and discussing collaboratively, supported by sketch papers, in front of a computer.

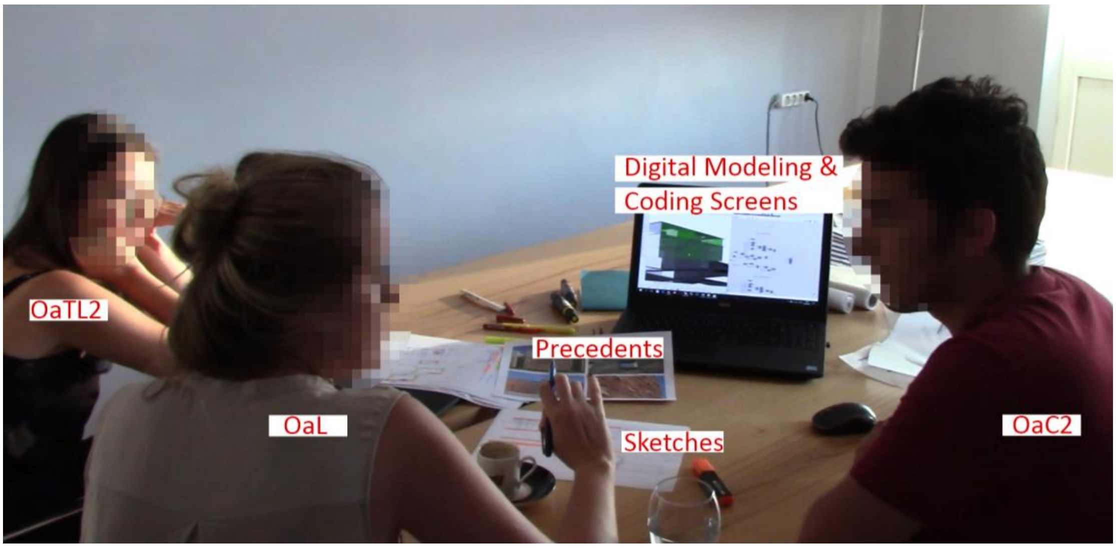


Figure 6. 11. O_aL and O_aC_2 are discussing the façade design problem on Rhino and Grasshopper software and by doing sketches.
(Source: The photograph was taken by the author)

The meeting was mainly about designing the façade and deciding the size and orientation of the façade elements. In the following dialogs, O_aL, O_aTL_2, and O_aC_2 were looking at the same computer screen and discussing the façade solution. O_aC_2 was already generated a solution that described superficially. The first solution was a trial that helped the team to see and make comments on the alternative that they were trying to describe to O_aC_2. In the following dialogs, O_aTL_2 asked to change the arches' radiuses:

00:00:00 O_aTL_2: "can't you change those arches?"
00:00:05 O_aC_2: "yes, with those archs with the pattern..."

O_aL interrupts O_aC_2 and orient them to the main idea.

00:00:09 O_aL: "Don't focus on the pattern now, the pattern is the next job. Let's find the thing about it first, then we'll look at the pattern. Because the pattern will also be related to the thing, we will relate to them with the alignments on the floors. Before the pattern, I think let's try to find out the thing... Actually, the pattern that your hand can remain for now (by showing the screen). But not because I chose it, just because something would show there. But let's say it will come from above and carry it at the level of the floor first. Let's say work on 4 meters alignments. One of them will be with two elements, this element 1 [Figure 6. 12#1], element 2 [Figure 6. 12 #2] is like this. Okay? Let them both come straight. Let's say... these thicknesses are too thick for me 40, I don't know."

O_aTL_2 and O_aC_2 searched the façade pattern rather than parametric logic behind the visual of the façade. O_aTL_2 asks the possibility of the change the arches' sizes, radiuses, and diameters. When O_aC_2 tried to express his thoughts about the arches and the pattern relation, O_aL stepped into the conversation and stopped the way that they were approaching the design. The office leader (O_aL) directs not only the design process, but also the way of the coder (O_aC_2) and the team leader (O_aTL_2) design thinking and problem solving. O_aL wants to develop a rule before creating a 'pattern' that indicate visual of the façade. O_aL gives attention to rules and design approaches rather than the shape and the view of the façade. The rule was described as sketches and conversations among the team participants. In the sketches below (Figure 6. 12), O_aL described to O_aC_2 the rule of arranging the elements to be lined up on the façade. O_aL described the elements to be lined up so that one invert was flat, and how many meters would be the intervals. Also, O_aL gave the height and thickness of each façade element.

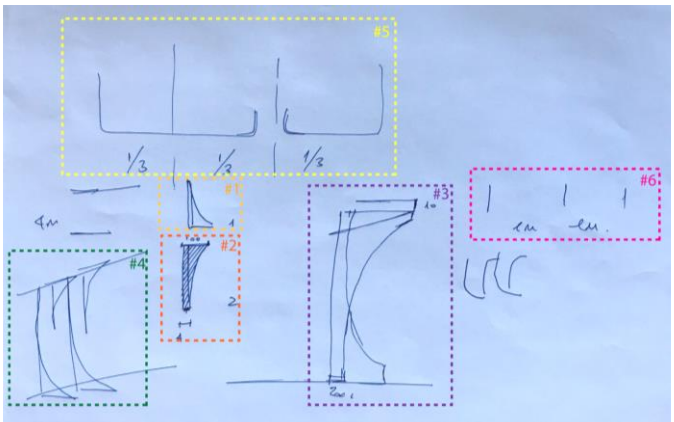


Figure 6. 12. O_aL sketches for the façade elements. #1 an element section for the façade. #2 an upside down element for the façade. #3 a side view sketch for the both options of the elements. #4 an isometric perspective of the façade elements with both positions. #5 proportions of the distribution of the façade elements on the façade #6 spaces between each elements on the façade.

(Source: Office A)

O_aL is capable of using computational design tools and is an expert in computation. Her knowledge base and skills enable her to follow and to guide the coding phase by drawing on-paper sketches annotated with numbers, partial formulas, and various geometries to define the steps of coding for the façade design. Her initial input broadly draws the boundaries of the coding algorithm to be used in the subsequent phases of design. So, when O_aL was transferring information to O_aC_2, she shared information by describing the steps that a coder also would consider.

00:02:35 O_aL: "here is one at the bottom of the top... open to me a scale version of it. Save it and scale. [*waiting while the coder is applying*]. I think it could be like a reverse, a straight. Or those coming from one direction top... Try it, I can't say anything unless there are a few alternatives, the first straight, the second reverse, try it (*sketching*) and let them come like this (*makes a cross with her arms and hands*). And it could be like this. The one that touches the top will arch at the top (*alternative 1*), and the one that touches the bottom will do the arch at the bottom (*alternative 2*). Like this (*sketching*), there is such underfloor. Then this will be an arch at the bottom and this will do it at the top (*alternative 3*). That's the rule. Make alternatives to this. I think we need to construct what we call a pattern according to what we give. At the moment it can look like very decorative. I think it makes sense to associate it with this gap."

O_aL asked from O_aC_2 produce multiple alternatives, and also O_aL made a list of rules for alternative alignments of façade elements. Although O_aL described the rules for alternatives, she didn't describe how O_aC_2 would work with which plug-in while scripting at Grasshopper. O_aC_2 had to find its own way in this regard. The general rules of the algorithm were transferred from O_aL but how to create the algorithm was left to the knowledge and skills of O_aC_2. While in the meeting, O_aC_2 started to ask about more specific questions such as the radius of the arches of the façade elements. O_aC_2 expressed his question about the arches as sketches (Figure 6. 12):

00:06:17 O_aC_2: [*sketching*] "now we have parts that are coming down from above, should both of the arches start in alignment? Or should this one start from a distance? Should they all end up aligned or these."

O_aC_2 was acting almost a producer of the façade elements but actually he was matching the both of the knowledge of architectural measurements and necessary information to apply into the codes in Grasshopper. Afterwards, O_aC_2 had a role much more an applier in the software rather than a designer. The conversation in the meeting followed:

00:06:34 O_aL: “I think the ending alignments should be the same in the first stage. But if you say so, give a rule. What alignment are you talking about when you say alignment?”
00:06:45 O_aC_2: “where that arch starts ends.”
00:06:47 O_aL: “That may also change. If there is an alternative, it would be better if you could create a situation where we could see that it was equal. Then we can compare.”

Even though $O_aC_2$was focused the practical part of the design, O_aL was giving a space to him for designing by his ideas. The software, Grasshopper, was an idea generator for O_aL and she was trying to direct the design process as using the tool in this way; she was asking for the alternatives that produced by the tool in the frame of rules defined by her. The discussion was continuing on the computer screen –Grasshopper and Rhino opened- and $O_aC_2$was showing the alternative that he created previously. However, O_aC_2 draw a sketch about the façade elements lengths and their change and asked (Figure 6. 13, red rectangle):

00:07:00 O_aC_2: “it starts here as it gets taller and longer…”
00:07:02 O_aL: “I think it's nice that it is. But I say it would not be bad if we saw that there is an equal. If you can put it in the script... it is something controlled [controlling the façade elements by the tool]. If the rule goes down after the equal…”

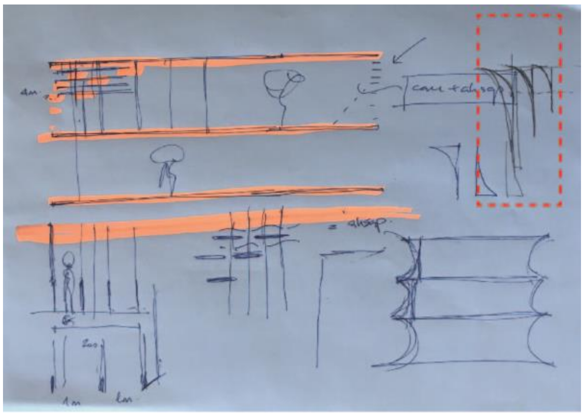


Figure 6. 13. O_aL sketches in the meeting. O_aC_2 sketches on the same paper with O_aL about arches’ alignments on the façade with arch’s radiuses are gradually increased (red rectangle, added by the author).
(Source: Office A)

O_aL and O_aC_2 simultaneously were sketching on the same paper to express their thoughts. $O_aC_2$began to handle the design problem in more and more specific aspects, he began to think about the details, and think about each façade element sizes, and also their positions on the façade. In order to develop the solution of the façade problem in a Grasshopper, $O_aC_2$had to elaborate the parameters that he had to construct in the beginning of the design process. In traditional design processes, when a designer is sketching or drawing in a CAD environment on a computer, the designer might proceed step by step instead of thinking about the entire façade at once. However, in the computational design process, $O_aC_2$preferred to design the entire façade by going through the whole and the detail at the same time.

During the meeting, although the team leader (O_aTL_2) was not involved in the coding part but, she was reminding about other issues in the decisions made about the façade; for instance, using endless numerous different façade elements would increase the cost of the construction. O_aTL_2's concern was about practical issues of construction rather than the geometric articulation of the pattern.

00:07:22 O_aTL_2: "Or for instance you will have 5 kinds of modules maximum. If they are all different, this time the method of producing the front will be difficult. Cost... why we are doing such a thing... you have a straight front, in fact it can be a bit contradictory. Because it has a modular state ... Why are you doing that?"

O_aL streamlined decisions of the design process O_aC_2's efforts in exploring design solutions by providing clear instructions via her sketches with numeric expressions and visuals, which were implicitly suggesting a rule-based exploration. O_aL did not create scripts herself, but she explained the governing logic, hence, the coder used visual images and numeric values to represent the solutions in Grasshopper as scripts or codes. The collaboration between O_aL and O_aC_2 sustained through the knowledge from architecture profession and coding. While communicating through both professions, the team preferred use conventional tools such as sketching. Besides the sketching, O_aC_2 used simultaneously updating the codes on Grasshopper in the meeting.

O_aL eventually approved the codes created by O_aC_2 as they together inspected the features of the façade on the computer screen displaying both the codes and the 3D models of the design solution:

00:07:51 O_aL: " you did it at one angle… three of…it will be nice if you think about it too, but if you don't, it is my acceptance. We make it up." [laughs]

00:08:14 O_aC_2: "I'll see if I can do it with codes. Because 'playing' is needed a little bit. And then there has to be that constant, the same round every time (?). Outstretching arms…"

00:08:26 O_aL: "If you can't reduce it to five types, don't worry. We look at it. As long as we see it... we'll do things accordingly..."

O_aL expects O_aC_2 to produce alternatives according to the criteria it has determined. OL-A is the one who sets these criteria, and implies that if a code is not developed for any of them, the problem can still be solved. It is expected the elements to cover the façade not to be infinite in number, and the façade elements should be produced of five different elements' arrays on the façade.

In the dialogues above, O_aL has presented O_aC_2 with the rules that may be related to their arrangement after describing each facade element exactly. O_aC_2 has a role as a applicator here. After producing on Grasshopper and Rhino, they focused on the alternative that they could use in the project.

6.2.1.1. Summary

The team described above was designing a façade for a competition project. An office leader, a team leader, and a coder (intern architect-scripter), and three intern architects worked together as a team. The office leader (O_aL) lead and participated the design process of the project and arranged meetings with the team regularly every evening. The team leader (O_aTL_2) assigned by O_aL as leading the all participants in the design process but the coder was directly lead by O_aL. All the team participants –except O_aL - worked in the same room and sit next to the each other. Because the team participants worked at the same table, they could ask each other questions at any time. Even though other participants of the office were not included in the team B, the team participants could easily communicate with other office employees around the table, who were intern architect, were able to easily access the experience and experience of other participants of the office. O_aL and O_aC_2 communicated through on sketching and Grasshopper which are shared knowledge domains. The information flowed as: O_aL produces as sketch, and then O_aC_2 to sketch, at the end Grasshopper and 3D visuals.

Table 6. 3 Phases of the municipality building project, in collaboration of the team. (Source: The table is drawn by the author)

Municipality Building	Planning Week 1	Co-creation Week 3	Co-creation Week 4	Production Week 5	Production Week 6&7
Level 1 (Initial design phase)	O_aL assigned the roles for O_aC_2. (in the meetings with all the team)		-	-	-
Level 2 (Idea Generation Phase)	-	O_aL, O_aTL_2, and O_aC_2 had a meeting specifically about the façade. O_aL gave the instructions for the façade elements.	O_aC_2 continued to create alternatives	-	-
Level 3 (Idea and Knowledge Sharing Phases)	-	O_aC_2 re-sketched the O_aL instructions from her sketches.		O_aC_2 shared the alternatives, O_aL decided to the façade.	
Level 4 (Discussions)	-	-	-	-	O_aC_2 finished the façade drawings before send to the renderings.

6.2.2. Library Project

The library project which is a retrospective research in the study is presented in this section. The participants in the design process of the library project are the office leader (O_bL) and the coder (O_bC_1), who collaborate as a close coupling in the interactions of them. O_bL and O_bC_1 were working together in front of the same computer screen from the initial design phase as much as possible. Afterwards, when the core team (O_bL and

O_bC_1) explored a form to apply to the structure, an architect specialized in structural systems (O_bTL_2), an architectural consultant (AC), and a draftsman (AD) participated the design process. Moreover, the project D's visualizations (renderings and Photoshop editing) were done by a visualization team in the Office B. From the outside of the office, the employer was visiting the office every month regularly and observing the progress and situation of the project. Also the team has comprehensive meetings two times with the employer and the consultants for the project.

The search of the form in the software Wolfram Mathematica were began years ago at the office B. The team and O_bL were concentrated and fascinated to find a perfect variation of the Hyperbolic Paraboloid form in the software by changing the parameters of the formula created by the software company. The collaboration was described by O_bC_1 as synchronous in the search phase of the design solution. The design process were held on two sided; one was searching for a shell that will cover the library and the other one was searching for programmatic design for the library. All of the solutions were also be evaluated with the structural system of the library. While O_bC_1 was changing the parameters in the Mathematica, the software produces a 3D images of the formula. At the same time, O_bL and O_bC_1 observed the 3D visuals and made assumptions which form could be applied to the design. Interestingly, unlike the usual design process, the team straightly worked on the software, rather than starting with sketches. O_bC_1 expressed that the team has not produced any sketches individually or as collaborative in the explorative stages of the project:

00:04:22 R: "hmm… well. was any sketch produced in the beginning?"
00:04:25 O_bC_1: "no, this is a culture of the office. This is a form that we researched, and we aimed to use it that way."

Although not very clearly stated, not producing sketches had become a culture of the office, or the form in which they were engaged, their culture. However, the team did not submit any early sketches of the design process to the research. Rather, the team presented sketches of the phases related to the structure, which involved O_bTL_2 after the form was created.

The following figure (Figure 6. 14) presents the screen of the software while O_bL and O_bC_1 searching possible alternatives for the shell of the library. The numeric values were adapted and changed by O_bC_1 and the 3D image of the formula was observed by the team. The Mathematica screen provides a two different information for the designers, one is formula lines and the other is 3D image that connected to the formula's values. Thus, two different information is propagated to the observers of the software outcomes. The propagated knowledge inform the designer in two different situation but, the designer can only transform the outcome through the formula, not 3D images.

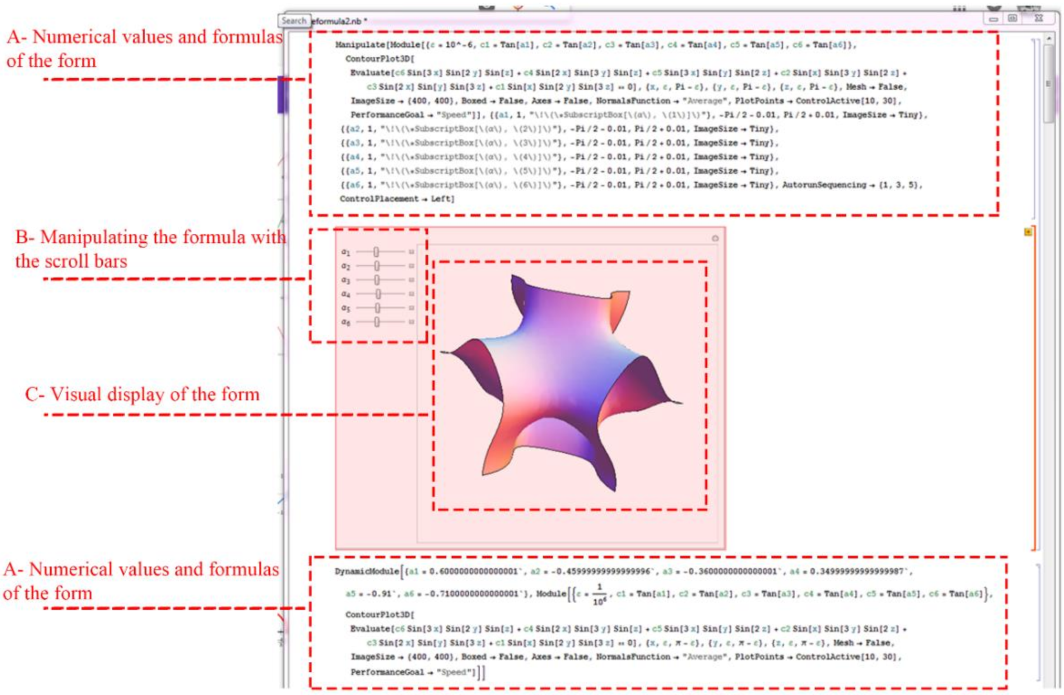


Figure 6. 14. A screen shot of Wolfram Mathematica software interface from the library project design phases.

(Source: Office B)

The formula creates curvilinear continuous lines (Figure 6. 14) and when O_bC_1 changes the parameters within the code, the lines re-shape and the system generates and adapts the surfaces between the lines. The team was trying to obtain possible alternatives for the shape of the library to create spaces for the library. The assessment involved the opportunities for daylight penetration, generation of an architectural space, and affordances for a structural element to form an arcade system. The team was progressing the design process in two directions; on the other hand, they were deciding the program and square meters of each spaces in the library (Figure 6.15) and examining the case projects. As shown in the square meters distribution of the program (Figure 6.15), the team needed to design large volumes for some spaces. Also, the team aimed to re-interpret "library" and reading rooms. For this reason, while producing some spaces, it was important that the form they obtained could carry itself.

00:07:35 O_bC_1: "We have had different inputs for it. In a library, you need to protect the library from UV. So we wanted to create a fringe and an arc. We wanted to create an arcade, we pulled the building back. Two; we didn't want to see columns inside the building, so we had the building moved from the top. That's why we need a special production shell."

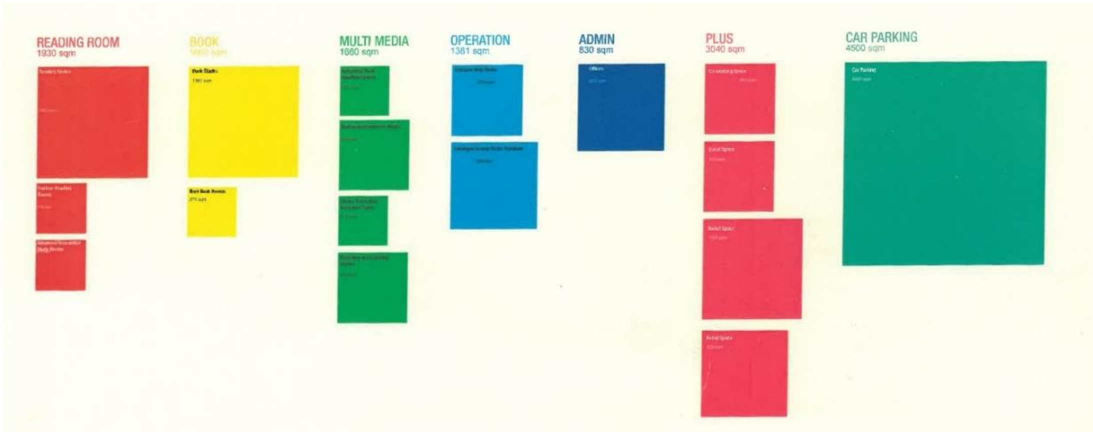


Figure 6. 15. The diagram produced for the square meter distribution of the library project, by the team.

(Source: Office B)

As it indicated in the interview, O_bC_1 express their design idea that a shell as a structural system. In the following figure (Figure 6. 16), the yellow dots represents when O_bC_1 change the parameters, the dots changes, too. Thus, O_bC_1 and O_bL while looking at the screen of the following image, they also envisioned the structural system, architectural spaces, and even daylight accessibilities. However, it is inabile to intervene on the mathematica screen on the 3D model by manipulating through mouse, it is only possible to change the form by the sliders and/or changing the formula's values.

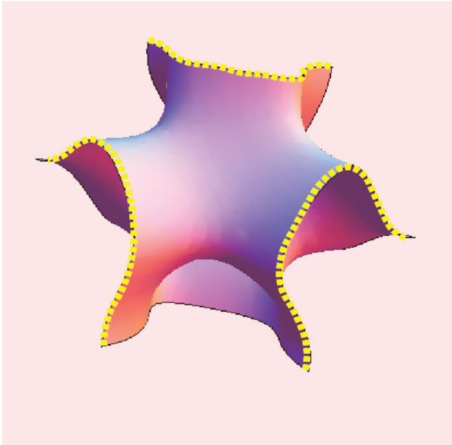


Figure 6. 16. Yellow dots show the lines that are created by the formula changes. (Source: Office B)

In the research, the library project was a retrospective study, but during the interviews, O_bC_1 opened the Mathematica software and transferred the process. In the following dialogues, O_bC_1 explains how they decided on the form:

00:01:04 O_bC_1: “this is Mathematica. Here we get such forms. We take these forms, but not ordinary forms, we see, for example, this does not work for me, for example, but (Figure 6. 17, showing the screen) that comes here, I mirror it, I deform it, and I can create something from it.”

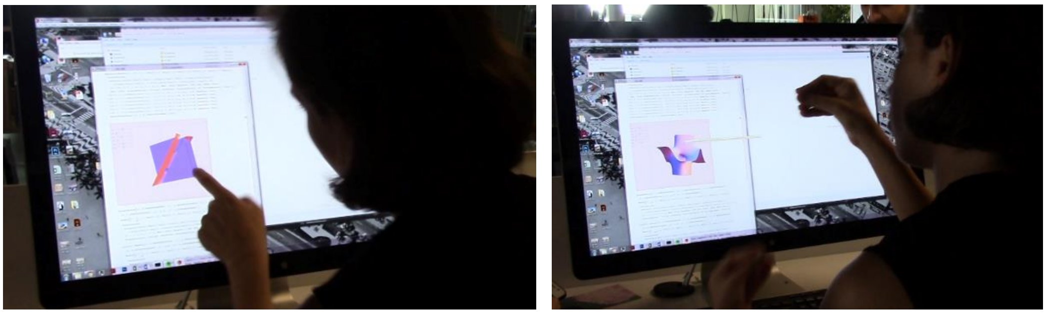


Figure 6. 17. O_bC_1 explaining how she use Mathematica to create an architectural space. (Source: The photographs were taken by the author)

[*Continues*] “We export them and import them into Rhino. In Rhino, according to Architectural inputs... what is it... this is height, slope, fullness, space...that’s it. We are developing these forms by saying I don't want to see that many facades.”

When O_bC_1 mention the expressions "mirrored, deformed, created", she actually describes the stages of collaborative working while working on the computer. The interventions to the produced form are related to an expert's (O_bL) view and at the same time aiming to reveal what is in mind. When O_bC_1 saw a 3D form in Mathematica, she calculates and assumes how the form could be transformed in Rhino. According to O_bC_1's and O_bL's assumptions, that they observed possible alternatives in Mathematica, they saved these alternatives (Figure 6. 18) for the next step, which included formal manipulations of the digital model in Rhino Software.

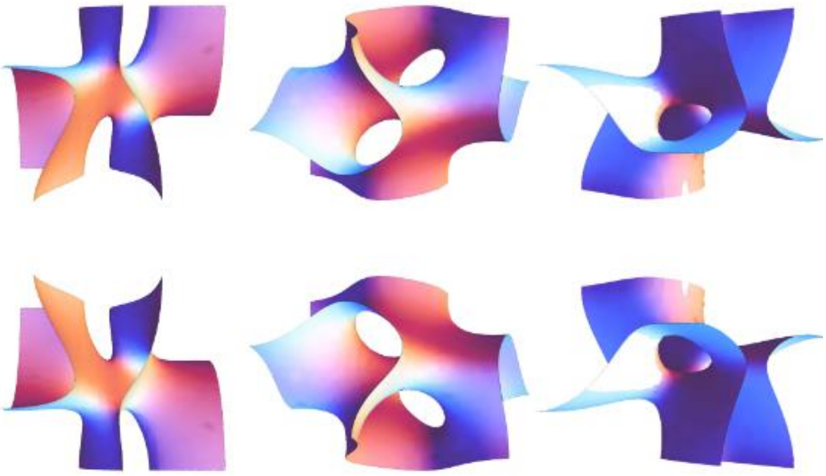


Figure 6. 18. Alternative forms that are created in Mathematica, by ObC1.
(Source: Office B)

In the Rhino, O_bC_1 makes to the interfering such as mirroring, transforming, etc. and applying the form in a structural system for the library (Figure 6. 19).

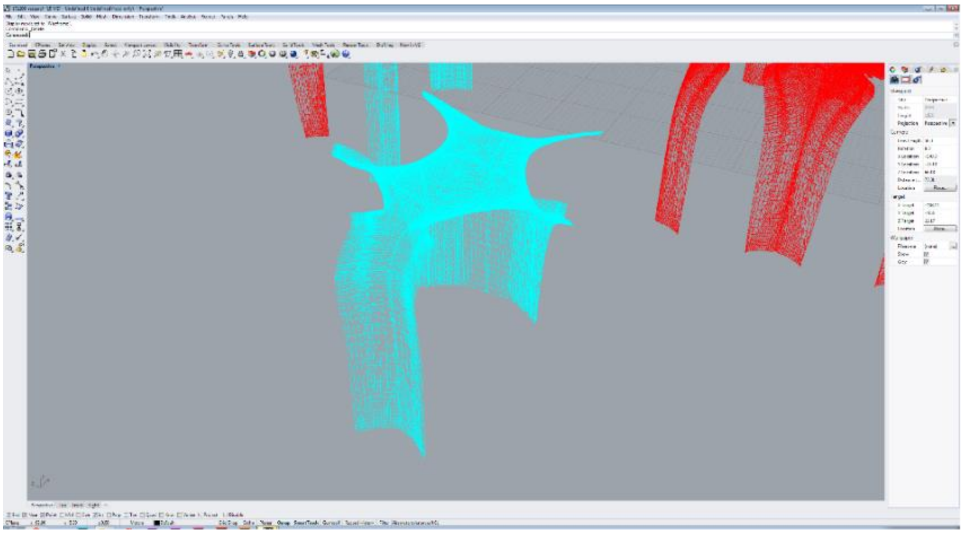


Figure 6. 19. An alternative form transformed into a structural system in Rhino, by O_bC_1.

(Source: Office B)

The purpose of transferring the created 3D form from Mathematica to Rhino is to create an architectural space. In their efforts to create architectural space of the library, the aim was to implement the space organization they wanted to propose, in line with the researches made from the very beginning. O_bC_1 expressed their purpose as:

00:03:05 O_bC_1: “We suggested other than them. We have proposed a library model where the books are collected as a core in the middle, there are reading rooms around the facade, and non-standard, that is, social things are added as a plug to the front. I can say that we have proposed a new typology.”

In this stage, O_bTL_2 [expert, architect] has joined the team. O_bTL_2 contributed to the team on the loadbearing systems of the form and other structural solutions while creating the architectural space. The program requested by the employer in the library could fit on three floors. Therefore, the form that created in Mathematica, which was tried to be applied in a 3-storey structure, caused some presumed constructive problems:

00:14:42 O_bTL_2: “when we put those three floors, that protruding portico that we wanted to achieve, as the floor was not connected, consoles of 15-20 meters began to appear...”

O_bTL_2 and O_bL had to make some decisions in order to solve the structural problems O_bTL_2 mentioned above. The team would either give up on solving the building with reinforced concrete and turn to the steel system, which is an expensive system, or they would give up on the form that created in Mathematica. Building in a standard form and getting the same look with the gypsum boards was not in their option. After long discussions with two different static engineers, the team decided to solve the shell of the building as a carrier, as they did not want any column in the center of the library.

After discussions and new decisions, O_bC_1 continued to solve the structural system of the library in Rhino. First, O_bC_1 designed a unit that including a column and a shell, then O_bC_1 multiplied and modified the units and combined them as unified with the structural system and the shell (Figure 6. 20).

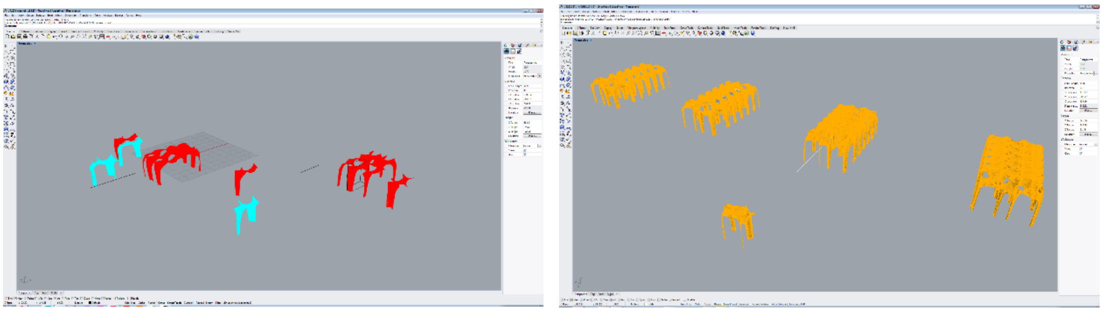


Figure 6. 20. Designing the shell with its structural system in Rhino, by O_bC_1. (Source: Office B)

At the stage when the team established the relationship between the form and the structure on Rhino, they started to produce sketches (Figures 6. 21 and 6. 22). The team participants worked with sketches on the issues they prioritized for the library, such as the entrance of the building, the areas where it will receive daylight, and the light shelves. In the following sketches, the design team decided firstly where to have entrance of the library in the site.

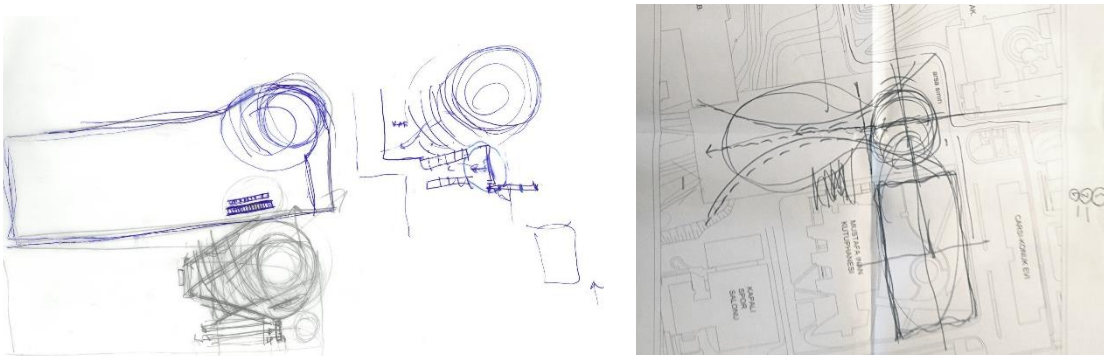


Figure 6. 21. Discussing with sketches about the main entrance of the building. (left) circular sketch expressing the entrance area (right) arrows shows the possible access to the library on the site.

(Source: Office B)

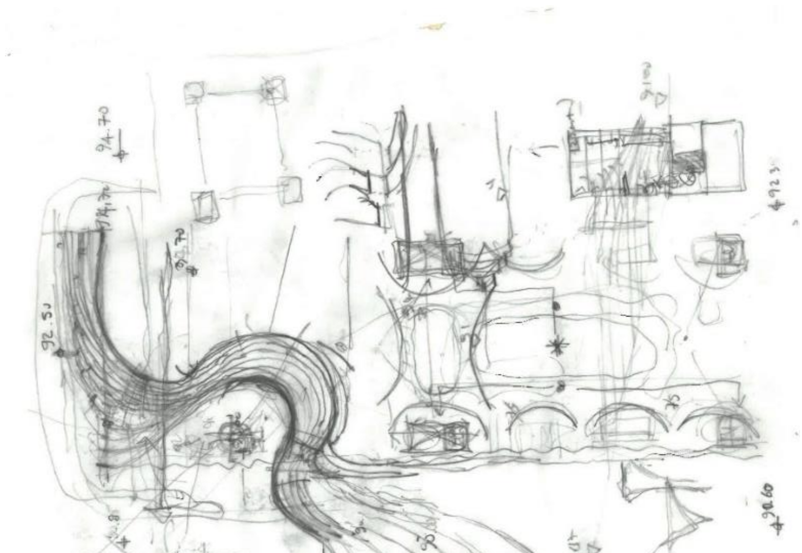


Figure 6. 22. Tracing the Rhino drawing with the previous sketches and exploring possible light shelf and the entrance with the structural system.

(Source: Office B)

While deciding on the entrance of the building, they also took decisions about where the circulation would be inside. In this stage, the structure of the building and the decisions regarding the skylight spaces to be opened in the roof shell were proceeding in consultation with O_bTL_2 and the static engineers (Figure 6. 23).

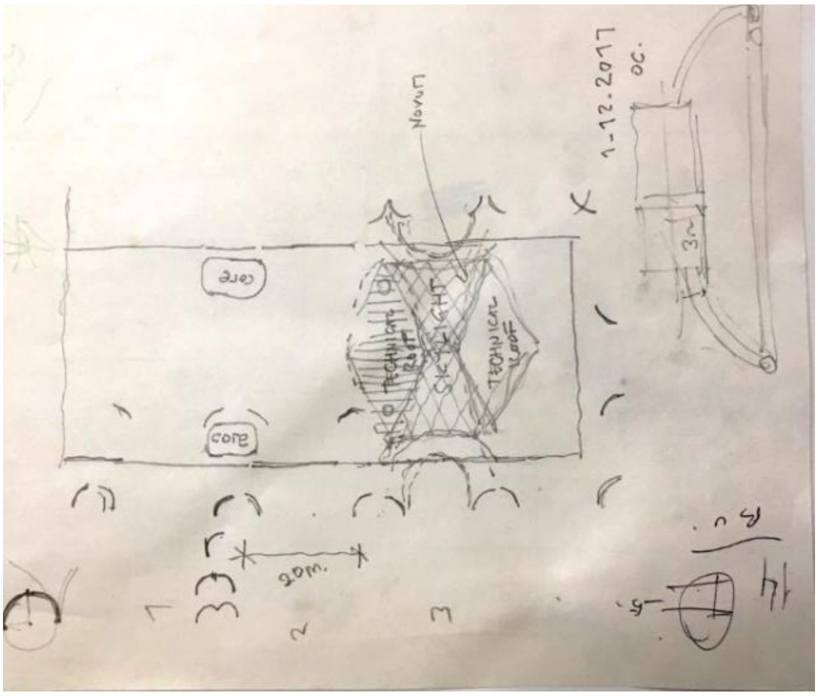


Figure 6. 23. Tracing the Rhino drawing with the previous sketches and exploring possible light shelf and the entrance with the structural system.
(Source: Office B)

Computational design tool, Mathematica, guided the exploration for the design problem with the 3D visuals it derived while O_bC_1 and O_bL were working on the same computer screen. The software also interactively guided the process when it produced expected or unexpected results, or what it discovered while searching for the expected one. Therefore, O_bC_1 and O_bL tried to apply in a structure every potential form in Rhino and then objectified it as 3D print (Figure 6. 24). Each possible form in Mathematica, followed implementation in Rhino. The form, which was specialized and adapted to a structure in Rhino, was produced as a model with 3D printing. Thus, the team continued the design process not only in the computer environment, but also on a physical production. The production of alternatives as 3D prints within the office is aimed at storing, showcasing, and archiving the information as a record of the office knowledge.

Figure 6. 24. 3D print models in the order of production from the top left.
(Source: Office B)

6.2.2.1. Summary

This section presented one of the dimensions of collaboration that is team leader and coder collaboration. The coder sits at the computer and tries to obtain a predetermined form in a pre-planned manner by the office leader. The presented segment is about a collaborative production with the one who know computation. The coder and the office leader worked synchronously in the initial phases of the design. The coder worked asynchronously between two software, first in Mathematica and then in Rhino. Between two software, the coder tried to apply the office leader's idea and also applied architectural necessities such as structural system that guided by the team leader.

Table 6. 4 Phases of the library project, in collaboration of the team.
(Source: The table is drawn by the author)

Library Project	Planning	Co-creation	Production
Level 1 (Initial design phase)	O_bL and O_bC_1 had meetings with the employer.	O_bL and O_bC_1 worked for form explorations together.	
Level 2 (Idea Generation Phase)		O_bL and O_bC_1 worked for form explorations together.	The form was achieved in Mathematica.
Level 3 (Idea and Knowledge Sharing Phases)		O_bTL_2 participated the team. O_bC_1 started to apply the form in a structural system.	The form applied in a structural system in Rhino.
Level 4 (Discussions)		O_bTL_2 and the consultants had meetings about the structural problems.	O_bC_1 finished the form creation and draw in a structural system. The visualization team produced renderings.

6.3. Discussions

In this chapter, episodes from two different offices with three different teams employing computational design tools in a multi-disciplinary environment. The first one (the tower project) is about who are designer and coder and their roles are intertwined, and the second (the municipality building and the library projects) is the participants who are designers and have role as coders in the team. In the second type, the coders have design knowledge, although they inevitably transfer design knowledge in the process, the office leaders gave more responsibility for the coding part of the process even if the office leader know coding or not.

In all the cases, the office leaders lead the design process. However, in the code writing part of the design process, the office leaders are either partially involved or left entirely to the coders.

Table 6. 5 Comparision of two different roles of the coders in teams.
(Source: The table is drawn by the author)

	“designerscripters”	**Designers as Scripters**	
	Project C	**Project B**	**Project D**
Participants	O_bC_2, O_bC_3, TL3, O_bL	O_aC_2, O_aL, O_aTL_2	O_bC_1, O_bL, O_bTL_2
Code	writing	writing	transforming
Roles			
Coder	Design and scripting	Scripting	Scripting
Office Leader	Product Evaluation	Designing and Evaluation	Designing and Evaluation
Team Leader	Designing	Advising	Problem Solving
Design	Form designing Explorative (partially)	Code-related design Explorative (form has not got priority)	Code-related design Presumed-expected form
Tool	Rhino-Grasshopper Sketch	Rhino-Grasshopper Sketch	Mathematica-Rhino Sketch

In the *tower design case*, O_bC_2 and O_bC_3 were both separately instructed to translate these initial ideas into schemes through adaptation of available scripts together with their visualizations. While O_bC_2 and O_bC_3 were experienced in both architecture and coding, O_bTL_1’s practical knowledge in generating and manipulating codes was limited. Therefore, when gathered around the computer screen the team’s progress mostly relied on O_bTL_1’s assessment of the formal qualities of the tower design which were shaped by the intentions verbally introduced and sketched by her. Accordingly, the coders' task involved a translation of design decisions that taken by O_bTL_1 into a code.

Although both of the coders (O_bC_2, O_bC_3) made different designs, they were often in dialogue with each other and exchanging ideas. Even, the other employees in the office could also be involved in the knowledge propagation, and daily dialogues continued, sometimes except coding and design issues. Sometimes an art object or daily object (coffee mug, glass, etc.) in their hands could be an inspiration for the design. In such cases, drawing an object in Grasshopper was the issue rather than the specialization of the idea taken from the object. Similarly, images or visuals or grasshopper plug-ins that the coders found on the online web pages, they were elaborating as a possible design solution and trying to adapt it to their design problem. This state of exploration led to a lot of information sharing. It might be positive but, it might also have lengthened the way to a solution.

O_bTL_1 directed the team in two different design solutions to show O_bL two different alternatives. O_bC_2 and O_bC_3 often exchanged information with the O_bTL_1 when developing these solutions. When O_bTL_1 saw a problem, she would see and solve the problem on the Rhino in 3D screen. But the coders were always trying to see and solve the situation from the Grasshopper. When there was a problem, the coders were not inclined to fix it by modeling through Rhino. Whether the stylistic situation bothered the coders or not, it was always their priority to satisfy them with the situation they developed in Grasshopper.

In the *municipality façade design process*, the office leader (O_aL) guided the design process from the beginning, and she expressed her thoughts and expected from the team also. O_aC_2 worked in the team as a coder even he was an intern architect. O_aTL_2 participated the dialogs between O_aC_2 and O_aL to convey her thoughts. O_aL as a designer has computational design approach and she was giving priority to computation rather than formal expressions. Hence, in the design process of the façade elements, the pattern of the façade elements did not have priority. O_aL transformed to a logic of the order of the façade elements then she expected alternatives according to the rules. O_aC_2 kept the role of applier of the decided rules by the team.

The façade design process were sustained by producing sketches and then computation in grasshopper. But, the sketches was not only 3D images of the thoughts, the numerical values also were included. In the presented segment in this chapter, O_aC_2

and O_aL sketches synchronously in the meeting but, in the extended process of the design, O_aC_2 always re-produced the sketches as his own sketches before transfer to the software.

In the *library design process*, the design team constituted by a leader (O_bL) and a coder (O_bC_1) in the beginning. The core team was working in a close collaboration in the early stages of the design. It is expressed that O_bL was participating the coding part as an observer and commenting the results that produced in Mathematica. As a leader, O_bL was trying to achieve a formal expression that he had in his mind. His formal explorations was sustained by coding in Mathematica. O_bL had belief that there was not possibility to imagine the design solution that they created at the end.

Mathematica software, an agent that plays a role in the designer's thinking and transformation. It can provide a two-way flow of information, between the parameters the designer enters while trying to achieve what it’s in the mind, and the software that gives the designer an idea by producing visuals while changing and transforming these parameters. Moreover, O_bC_1 tried to foresight the produced form as transformed in Rhino. O_bC_1 visualized the transition between two different software, Mathematica and Rhino. The knowledge gathered in Mathematica had transformation and it was propagated to Rhino by O_bC_1. Then, presented to O_bTL_2 whom responsible for the structural problems.

In software such as Mathematica, while executing the design process in digital tools, there is no possibility to trace it with the previous design stage as in sketching with tracing paper. While the team often 3D prints, they tried to trace the process by lining up the 3D print models side by side, just like the tracing paper supplied on sketching.

This research presented first, how different design reasoning mechanisms are interlocked with different tools and methods of representation, starting with sketching and coming up to visualization (the tower design and the municipality building design), and converted to inseparable in collaborative design process. Second, in the traditional design process (hierarchical), the novice develops and matures while working with his/her master. However, with the contribution of newly developed digital tools such as computational practices, the hierarchy disappears in the collaborative work. Just as the tools were intertwined, the design approach also intertwined and hybridized. While there is computational design, one can progress with a formal search (the tower design and the municipality building design). However, the hybridization in the teams can cause the disagreements in the teams. When O_bL could not participated the design process (the tower design), needed to renew the team leader with new one.

Multi-disciplinarity in the teams in the cases was supported by team participants rather than incorporating the consultants. Architects in the teams undertook a double role as scriptwriters/coders and designers, whom Burry (2011) calls 'designerscripters'. As D'souza (2020) argues contemporary designer needs to be a 'multi-skilled designer', stating that only being an design expert may not be enough. A designer needs to be 'skilled' in many issues including in collaborating or in adapting remote disciplines into design. In our cases, designer-coders in the teams needed to have an understanding of their team leaders' and their office leaders' design approaches and needed to find a way to translate and sustain their design approaches via computational design tools. Multidisciplinary collaboration has been increased and architecture discipline is in partnership with many different disciplines. In addition, working with individuals from different disciplines is now getting stronger with the multi- disciplined participants in architectural collaboration. An architect can also be a computer scientist. Thus, individuals can enrich the solutions by adapting the different areas of knowledge they have acquired to the work they are dealing with. Nowadays, we often come across architects who specialize in the field of "scripting" languages in architectural collaborative works. 'Architect' is no longer just architect but also architect + coder, architect + graphic designer, architect + renderer, etc. In the teams observed as part of the research, the team leaders were often not experienced in scripting domain. Newly graduated architects are mostly preferring computational design tools to extend their knowledge domains. However, there may be some disagreements between team leaders and scripter-architects because of non-overlapping parts of their knowledge domains.

The multiplicity of interfaces on computer screens –to reflect both codes and the architectural qualities – and on paper in different sketches, and in different modalities allowed participants to negotiate and progress. Often these different versions of representations denoted the same content ensuring again redundancy (Hutchins, 1995) in the system.

The research offers instances in which design teams leaders coordinated and developed the architectural design proposals in tandem with individuals with extended skills in computing. While the developing common language helps achieving the consensus and progress in design, the redundancy of overlapped knowledge domains (Hutchins, 1995) was observed to be critical in creating a robust collaborative practice to create multiple alternatives.

The tower design and the municipality building design teams' participants initially used 'paper-based sketches' to propagate the knowledge among the actors in the team. In such collaborative environments, drawings with definite details are considered as explicit modes of representations that partially eliminate interpretation compared to other forms of coordination (Nomura & Hutchins, 2006). To achieve coordination, in the Team B, O_aL preferred to produce relatively more explicit detailed drawings to guide the work of coders. In the observations the teams B and C mainly supported their imagination with computer visualizations on screens, sketching on paper, talking, referring to precedents, and gestures. Thus, the knowledge was propagated in the distributed cognitive system via multiple modes that ensured robustness through redundancy.

Having coding skills, O_aL was able to communicate her intentions about design directly, which helped the team to arrive at a solution quickly. Concerning coding, the overlapping knowledge domains possessed by both O_aL and O_aC_2 ensured robustness in the system (Fig.26a). In the Project C, the knowledge overlapping situation was different (Fig.26 b, c). O_bL, who did not have coding skills, used other ways of conveying his ideas, which created more ambiguity in pursing acceptable design solutions. Thus, Team C produced more alternatives and their search space was enlarged unnecessarily. In comparison, the first case, where the robustness in the distributed system was ensured, the solution was reached quickly, but it was not possible to explore possible alternatives because the search space remained limited.

Talking about a hierarchy among undiscovered design solutions, Woodbury & Burrow (2003) state that 'effort' and 'connectivity' are important in terms of accessibility to design solutions. In our cases, O_aL streamlined O_aC_2's efforts in exploring design solutions by providing clear instructions via her sketches with numeric expressions and visuals, which were already implicitly suggesting a rule-based exploration. O_aL did not create scripts herself, but she explained the governing logic hence, O_aC_2 used visual images and numeric values to represent the solutions in digital media. According to Burry (2011), this points to an inevitable handing over some responsibilities to juniors who are more up to date with current advancements in computational tools and techniques.

This chapter has discussed situations where hierarchy disappears, knowledge begins to be distribute, and where multiplicity exists, and the discipline of architecture's own knowledge domain's sub-plan is multiplied - or there can be multiplicity from discipline- among the mentioned computational practices above. Therefore, the hierarchy may be overlaid by a different mechanism, and which means for collaboration might be a situation where different collaborations can develop in each design process. Different from the traditional collaboration, the team leaders and junior participants may discover again and again new ways of collaboration. In the municipality building design project, the office leader (O_aL) participated the design process as actively in every moment and with her feedbacks. Similarly, O_bL participated the design process actively and gave feedbacks in any moment.

In the presented cases, the feedbacks of the office leaders, such as the example of O_aL, where they have the computation knowledge and can actively participate in the process, and when the office leaders do not have the computation knowledge, they can stay outside in the coding part, manage the process from outside, and have a non-hierarchical majority, where there are participants who have the computation knowledge, are discussed. It has been disclosed how the traditional and hierarchical interlocking diagram (Figure 6. 25) begins to diverge, both in terms of collaboration and that the nature of the design can change (Figure 6. 26).

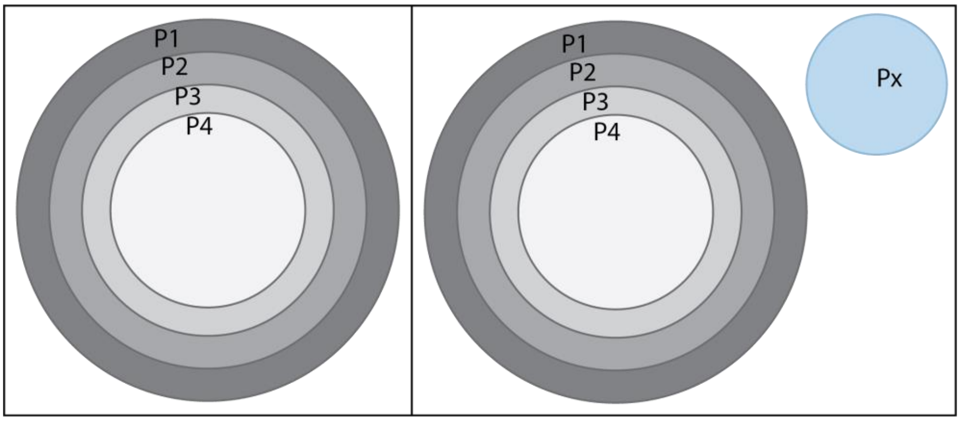


Figure 6. 25. (left) Hutchins' overlapped distribution of knowledge, (right) A non-overlapped distribution of knowledge in a design team. P: Person
(Source: The diagrams were drawn by the author.)

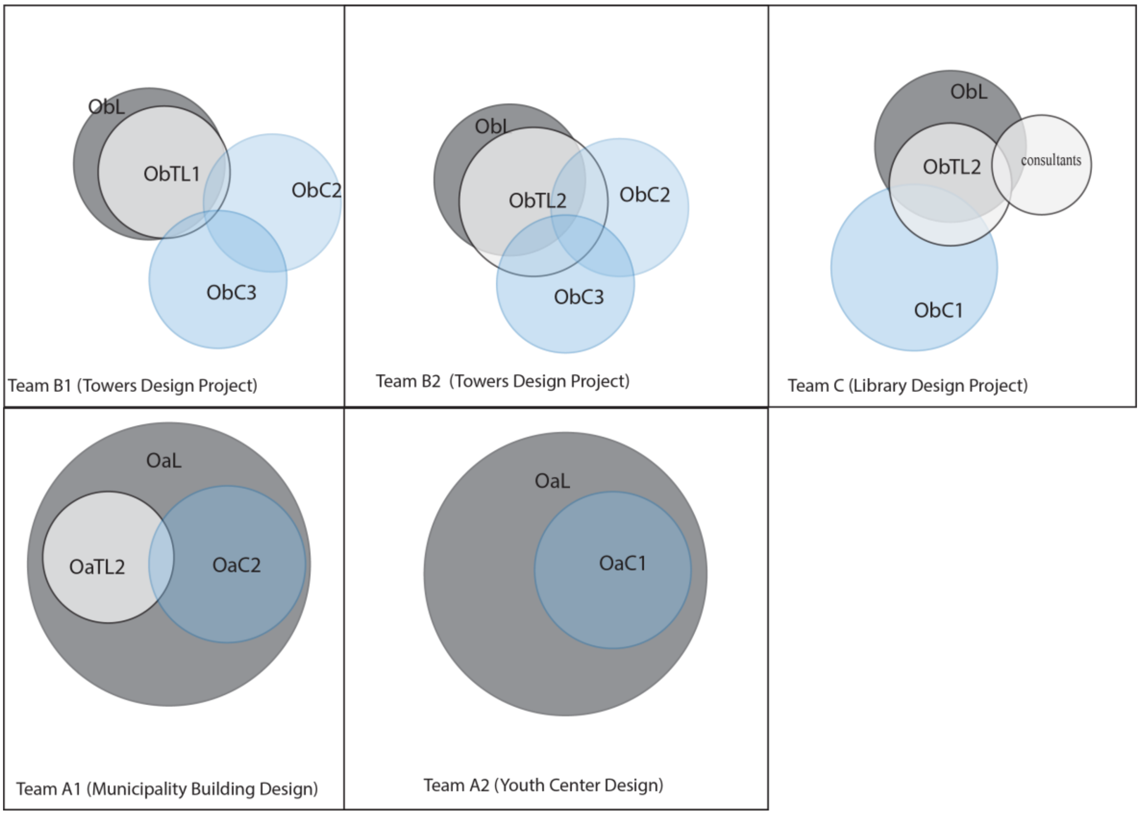


Figure 6. 26. Overlapped and nonoverlapped knowledge areas of the teams. (Sizes of the circles represent the area of the knowledge about computational design tools). (Source: The diagrams were drawn by the author.)

In the office A, the office leader and the coder have wider knowledge area because they are expert both areas of knowledge they can communicate more among themselves. And they don't need intermediary solutions or tools to support their collaboration. However, in the office B, the team participants need to have alternative supportive procuders facilities for collaboration; (1) computer screen, (2) gestures, (3) language, (4) sketching (5) 2D printing, (6) 3D printing. Because of not knowing computational design tools, the team participants need for additional supportive communication tools. Even the users know both same languages, the team participants still require additional communication tools just like in the cockpit (Hutchins, 1995b) where the information is re-represented to make more redundant.

Since the computational knowledge areas among the participants of Office B do not overlap, they need to resort to repeated representations in order to provide redundancy. In this way, they were able to represent the information multiple times within the collaborative system. The production of all alternatives as 3D prints may have been done to ensure repeated representation and redundancy.

McComb & Jablokow (2022) state that the collaboration of different disciplines generate a new disciplinary understanding, and could be consolidated with a lingua franca. The cases suggest that not only the terminology, but the skills and capacities activated within collaborative computational practices are necessary to facilitate a sphere to translate design intentions into representations of design. In the cases presented, the coders who were also junior designers acted as '*mediators*' between the office and team leaders and computational design tools. As office structures extend to interdisciplinary fields such as evolving computational tools, they undergo changes and become dynamic. Rather than having a fixed team of participants, individuals from various disciplines involved in the design problem participate in design process. Instead of a hierarchical system, a structure emerges where production begins at the intersections of knowledge areas.

Computational design tools and especially parametric modeling tools such as Rhino and Grasshopper facilitated cooperation by way of representing the design information on two different screens; one representing codes and the other 3D view screen. The simultaneous use of two screens made the collaboration between the coders and the team leaders simpler, by creating a medium to merge knowledge from different domains.

Design activity cannot be without a representation. Ideas must be represented to share with others and oneself (Porter et al., 2001). Designing is defined as "the production of a design representation" (Galle, 1999, p. 63). Design representations play role on design process and product (Galle, 1999). Design representation is identified with two fundamental roles: 'communication' and 'exploration' (Eck, 2015). Computational design tools are an alternative exploratory tools and have the potential to extend human imagination. van Berkel (2013) expressed that computation has potential of possibility and flexibility of communication among multiple disciplines. In teams, the design might progress in multiple directions, with multiple participants navigating in slightly different directions. While each navigating actor might navigate individually in an ambiguous space, there is a path to follow which is defined by legible and explicit design instructions. A multi-participant process, if it is going to be successful, requires a direction with a shared purpose and a calculated aim as suggested by the distributed cognition theory. Creativity in design cannot be sustained only through explicitly defined and shared instructions; it also requires explorations of a serendipitous solutions hidden within ambiguity, where each participant could want to act individually to explore. As suggested by Coates (2010), it is possible to introduce serendipity in a computational environment such as through generative algorithms facilitating emergence of unexpected outcomes, which may support a creative system. An alternative may lie within a coupled system of human designers and computational tools, in which human agents explore opportunistically visual results of numerical inputs computed in the software.

CHAPTER 7

GENERAL DISCUSSIONS

This chapter discusses the overall findings of the thesis within the framework of two main topics. The first topic examines how the concepts of serendipity and legibility emerge in the design process and how they shape the process. The second section discusses how knowledge is distributed within interdisciplinary environments in design teams.

7.1. Serendipity and Legibility in Computational Design

According to field observations, it is found that designers use computational design tools to explore multiple alternatives and approaches, while narrowing down the options based on selected design criteria. Design process involves a sequential construction of an explorative space where designers navigate and recognize clues of possible solutions (Terzidis, 2006). Ambiguity or uncertainty is considered as an advantage for creative thinking (Runco & Pritzker, 2013) in the way of allowing serendipitous explorations of design ideas which would not otherwise have emerged in the design. However, the downside of ambiguity is the possibility of not converging on an acceptable design solution. Therefore, while expanding the exploration space of design problem, designers also try to provide legibility so that the design rationale can be understood by third parties and by themselves. While designers try to keep the exploration space wide by providing maximum ambiguity, they also aim to create an understandable and legible design idea. Computational tools are flexible and adaptive tools that provide a wide exploration space, and also capture the sequence of how design ideas emerge.

Computation in design is used both to explain a design style via computing visually (Knight & Stiny, 2001) and to explore a multitude of alternatives in a formalistic way (Terzidis, 2006). Computational design is considered as a revolution from predicted to unpredicted (Kolarevic, 2013). Therefore, starting with rules or predetermined criteria in the design process revolves around an exploratory process that evolves through computational design tools, leading to unexpected outcomes. Designers aim to discover and obtain the unknown, and they perceive these unconventional solutions as creative. The coders observed in this study, although not as proficient in computational tools as computer engineers, have actually expanded the realm of exploration by using the tools in a more ambiguous manner. Their limited expertise in the computational tools has made them more open to coincidences, but it has also provided them with a greater range of possibilities. Rather than strictly inputting precise numerical values to achieve a specific outcome, they have adopted a more intuitive approach, utilizing the tool's manually adjustable features to their advantage. As an illustration, in the presented cases, the architects and coders have greatly facilitated the production of diverse forms by engaging in activities such as manipulating sliders within software interfaces, serendipitously exploring components through search engines, and actively integrating traditional methods with digital tools. These practices have enabled them to explore a wide range of possibilities and achieve unique outcomes. However, on the other hand, there were factors and evaluations that guided the design, such as design criteria, the validity and comprehensibility of the design idea, and its feasibility. As a result, designers were able to find their direction and navigate through an exploration space. Designers, in collaboration with individuals and tools, have worked to navigate the process between ambiguity and legibility. Tolerance for ambiguity enhances the creativity of the design process (Guilford, 1950), but it can also delay designers from reaching a final outcome. By setting rules and establishing boundaries, designers have accelerated the design process, but in doing so, they have limited the number of alternatives by narrowing the scope of exploration space.

By clearly visualizing all the rules and the steps, the studied team participants provide a clear trajectory of tasks and operations. As emphasized by Knight and Stiny (2001) visual computations could be improvisational, perceptual, and action-oriented as much as algorithmic. In the observations of the youth center design, O_aC1 expands the exploration space by improvising visually and computationally through both sketching and modelling. In the visual computations of the youth center, O_aC1 moves her idea from the beginning to the end of the design process through detailing, which is called as a vertical transformation (Goel, 1992). Interestingly, the coder’s preliminary design phase is relatively well-structured rather than being ill-structured as it is mostly assumed in the literature (Goel, 1992).

Referring to work of Gün (2012), George Stiny states that designer thinks visually and seeing is the most interesting part of the design process. The formulation offered by Knight and Stiny (2001), is a form of “reflective practice”; an interactional process in design which follows “seeing-drawing-seeing” as defined by Schön (1991). Designer reads the situation and interprets again while drawing in the design process. However, in the observations the municipality building design and the youth center design followed a visual prescription in the initial phase, namely sketching which is supported by numerical explanations. Then, following explorations with computational tools progressed through both visual and formal representations and formulizations. The obvious thing is visual representation facilitates and ensures explicitness. However, in the observations, the designers tried to intentionally create complexity as well. For this reason, they extended their exploration space and wandered in ambiguous exploration spaces by adding number of alternatives. In the youth center project, although OaC_1 employs rules to govern the geometric order of the units' forms in the first stage, a level of ambiguity is added to the process through the use of computational methods in the way these units are multiplied on the site. In the unit design process, the first stage is well structured with setting the rules, and then in the second, O_aC_1 follows an indeterministic path to set a route by duplicating the units. In the observations, the coders in the youth center and the municipality building designs followed a clearly calculated initial phase then applied randomness to obtain ambiguity which could support serendipitous explorations.

Sketching is a valuable tool for representation and exploration (Schön, 1991). ‘Reflective interaction’ (Schön, 1991) and ‘seeing and doing’ (Stiny, 2006) relate to the notion of ‘emergence’ in shape grammars (Knight & Stiny, 2001). Others have also emphasized emergence of ‘unexpectedness’ through ‘ambiguity’ (Coates, 2010; Knight & Stiny, 2001). As in the studied cases, the designers define a space to explore alternative possiblities. But, outside of the predetermined exploration space of computation there is an ambiguous exploration space acheived through ensuring ambiguity in the system. Designers decide a route to follow that shapes the design process through shared decisions in the team in the form of legible computational descriptions. However, while aiming a legible description, there is always an aim to have a creative solution that triggers designers to further explorations in an ambiguous search space.

When the process start with defining certain rules to guide the development of design, it is evident that there will limited room for ambiguity. However, as the coders progress in the design process within the computer environment and generate alternatives, they step into a conceptual space of design exploration. Therefore, while computational design advances with pre-defined rules without the computer, ambiguity is only supported within the exploratory space with the aid of digital tools. Computational design tools are an alternative exploratory tools and have the potential to extend human imagination (Terzidis, 2003).

In collaborative teams, the design might progress in multiple directions, with multiple participants navigating in slightly or radically different directions. While each navigating actor might navigate individually in an ambiguous space, there is a path to follow which is defined by legible and explicit design instructions. A multi-participant process, if it is going to be successful, requires a direction with a shared purpose and a calculated aim as suggested by the distributed cognition theory (Hutchins, 1995). Creativity in design cannot be sustained only through explicitly defined and shared instructions; it also requires explorations of a serendipitous solutions hidden within ambiguity, where each participant could want to act individually to explore (Heylighen & Martin, 2004). As suggested by Coates (2010), it is possible to introduce serendipity in a computational environment such as through generative algorithms facilitating emergence of unexpected outcomes, which may support a creative system. An alternative may lie within a coupled system of human designers and computational tools, in which human agents explore opportunistically visual results of numerical inputs computed in the software.

The outcome of an algorithm cannot be known without running it (Coates, 2010), therefore, in the presented episodes, designers attempt to progress in a more predictable manner through visual means. In the case of the library design, the slider displayed on the Mathematica screen is manipulated freely by the coder, along with altering the numeric values, and decisions are made based on the visuals on the screen. The computational tool supported the designers' exploration processes by performing calculations interactively and immediately in the background. Developing coding in Mathematica using a text-based language provides designers with an open-ended exploration workspace that allows them to reach multiple alternatives. However, due to the coder's lack of sufficient mastery of the tool and the need to progress through visuals while working with the office leader, the coder has required visual representations that originated within the exploration environment.

Serendipity has been attributed to factors such as time constraints and social relationships in the context of discoveries (Copeland, 2019). Additionally, the field observations and interviews indicate that inadequate knowledge sometimes leads to unexpected explorations. Instead of writing algorithms in Mathematica, the coder has manipulated a ready-made formula using a slider, thus uncovered the unpredictable in the design solutions. Emphasizing that reaching a form unthinkable with "human mind" requires computational design, the office leader (O_bL) highlights that expanding the realm of exploration relies on the computational tools and interdisciplinary teams.

In the following graph (Figure 7. 1), the observed cases are placed according to the density of serendipity and legibility. For example, in the library project, the computational tool was used intensively from the very beginning of the design and the exploration areas were kept wide. There was not any predetermined format for the design process and the product.

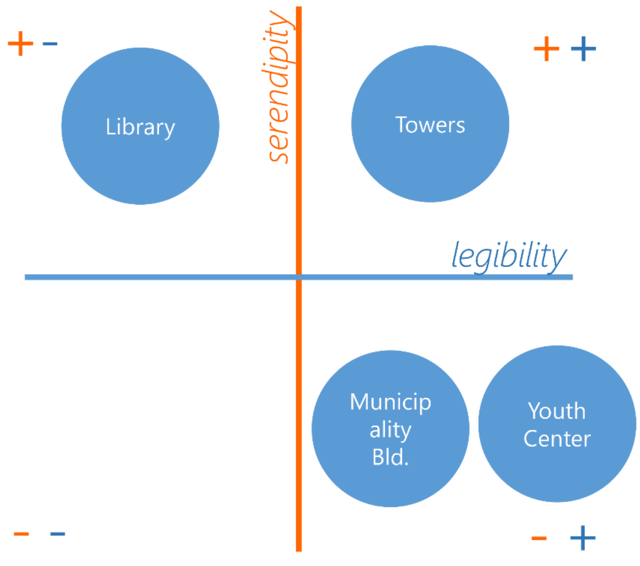


Figure 7. 1 The observed cases are placed according to the density of serendipity and legibility.

(Source: The diagrams were drawn by the author.)

On the other hand, in the Municipality building and the Youth Center projects where the computational tool was used for a short period and the problem-solving approach was predetermined, the exploration space was defined and limited. There was a high emphasis on the legibility of the project concept. In the Towers project, however, the coders were given more autonomy in design solutions, so they often kept the exploration space wide. However, legibility was a desired aspect requested by the office leader.

7.2. Distribution of Knowledge and Collaboration

Currently the accessibility of architectural knowledge and the ability of emerging technologies to produce many alternatives under the name of architecture are reshaping the architectural profession and its relationship with other professions. Architects are no longer sole individuals with expertise in space making; they also need to be knowledgeable in various other fields. In doing so, they can design and manipulate their own design tools through evolving technologies, similar to how many architectural design offices expand the use of certain representational modes like diagrams and physical models (Yaneva, 2009) or develop their own tools like Gehry technologies (Kolarevic, 2003; Smith, 2017).

At this stage, designers are increasingly inclined towards individual work during the design stage rather than relying on a master-apprenticeship model. However, they still benefit from a collaborative system to adapt new tools into the design process. Despite not being highly proficient in all the available computational design softwares, designers can integrate these tools into the design process using numerous open-source computational design tools. The manipulability of these tools provides convenience to designers, allowing them to access more alternatives and make intuitive progress. The traditional hierarchical system in architectural design teams is transforming into a different order through the addition of new fields of knowledge, creating a collaborative system that is reshaped with each new project. Consequently, even in their individual work processes, architects now participate in a collaborative process through digital means and open sources. If the areas of expertise of experienced architects within teams are limited to the field of architecture, less experienced architects who are knowledgeable about emerging technologies possess expertise in knowledge areas beyond the hierarchy within the collaborative system.

During the process from designing a structure from scratch to its construction, multiple individuals work collaboratively and/or individually. If it is considered that this entire process is divided into the design process and the construction process, a system is created by architects in which collaboration occurs among themselves in the design process. This collaboration is also divided into sub-processes where individuals work on their own. In these individual sub-processes, designers progress in collaboration with tools within the distributed cognitive system. In the design process, including the individual work processes of the coders, communication can be established through visual representations with all team participants, whether they have coding knowledge or not. The coders follow the process through both representations which are 3D visualization and coding. Therefore, visual representations were the fundamental means of knowledge sharing in the distributed cognitive system.

Design problem solving is a process of exploration, and larger exploration space provides larger number of alternatives. Collaborating with different disciplines to enlarge the exploration space enriches the process of the design problem solving. Although the teams assume that computational design creates a large search space, they are also aware that computation tools can cause the design to deviate from its intended aim. On one hand, the teams may have initially had a loosely defined design objective. On the other hand, their intention was to explore all potential alternatives. In this contradictory situation, when teams work together with various disciplines, different knowledge domains and experiences are brought together, creating a multidisciplinary environment. While ambiguity may arise in team composition, where knowledge domains do not overlap, it can limit the variety of ideas. Conversely, legibility, where robustness is ensured due to overlapping knowledge domains, is more common in teams was observed. In teams where knowledge domains do not overlap, the legibility of the design idea remains more ambiguous.

In the observed teams, when participants with different knowledge areas could not use communication with the tools in which they were experts, they resorted to representational systems that they were accustomed to, such as sketching. Despite the integration of digitalization into the design process, various representation systems continue to be utilized, such as sketches, models, etc. In addition to traditional representation systems, computational design tools are also used together. Especially when the knowledge bases of designers within a team do not overlap, the representation systems used become critical, and both traditional representations and computational design tools become instrumental in completing tasks. The observed teams were not non-overlapping; instead, the team participants had overlapping knowledge in the architecture discipline (Figure 7. 2).

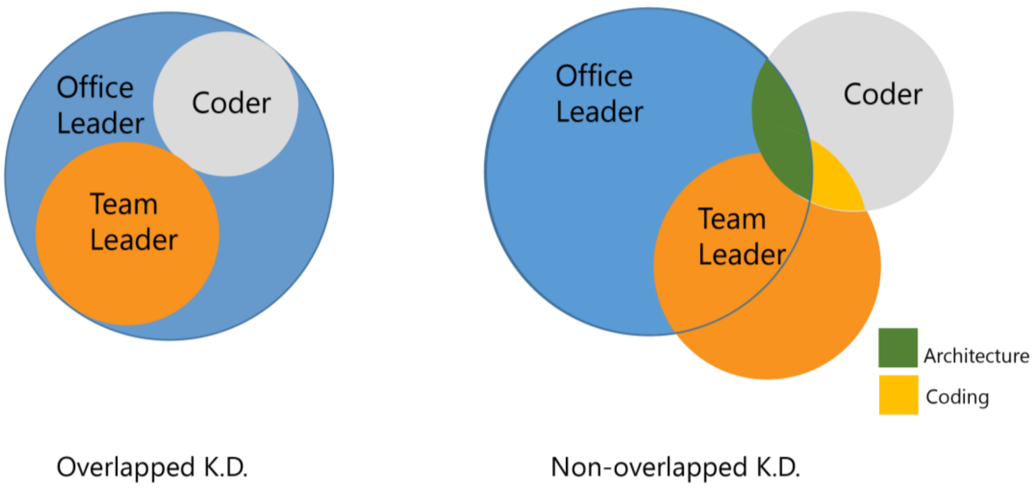


Figure 7. 2 Overlapped and non-overlapped knowledge areas of the teams. [Sizes of the circles represent the area of the knowledge requied to execute the task].
(Source: The diagrams were drawn by the author.)

When knowledge domains overlap, these systems are perceived as being more robust. If a comparison is made between the concept of robustness and the concepts of serendipity, legibility, and knowledge domains, a greater overlap of knowledge areas results in collaborative systems becoming more robust, accompanied by an increased intention to make their design ideas legible. On the other hand, when knowledge domains do not overlap, the system's ability to control itself diminishes, leading to reduced robustness. This circumstance gives rise to additional demands for legibility, such as the documentation of the design process, and an increase in the occurrence of serendipitous exploration within these systems.

Robustness is primarily related to a state of distributed system in which the system continues to function regardless of failures encountered during the course of any task execution. Robustness is ensured by redundant distribution of knowledge across team components that allows dynamic reactions to unexpected events. Hutchins states “We can think of the team as a sort of flexible organic tissue that keeps the information moving across the tools of the task. When one part of this tissue is unable to move the required information, another part is recruited to do it.” (Hutchins, 1995a: p. 223).

According to Hutchins (1995a) robustness is a matter flexibility within the system, ensuring the functionity of the system in unexpected circumstances. In creative domains, what matters most is ensuring the generation of potentially acceptable novel ideas (serendipity) which is proposed to be a matter of serendipity; being open to benefit from unexpected opportunities (ideas). In creative domains, serendipity (to foster novel idea generation, one needs to be open and ready to emerging alternatives) (Sawyer, 2003). Legibility is understanding the otherside. According to Sawyer (2003), serendipity in ensured through one is deep listening, second is openness to new ideas. (3) Following Sawyer (2003), this research proposes that in a collaborative design task serendipity is ensured through either (i) overlapped knowledge domains (Hutchins, 1995a), (ii) translatibity and transferribilty of ideas through a multitude of design representations.

Within Sawyer's (2003) perspective, the participants are all theater artists, demonstrating a strong inclination to listen to and understand each other. With the increase in legibility, there is a corresponding increase in robustness when knowledge areas overlap. The crux here lies not solely in the accurate transfer of information, but also in the capability to introduce new expansions while conveying this information. While ensuring the accuracy of information transmission remains crucial, maximizing the potential for new expansions also emerges as a priority. Hence, the presence of overlapping knowledge domains, wherein partial design knowledge is possessed by the coder and partial coding knowledge is exhibited by the designer, facilitates improvisational communication. This forms the basis for effective collaboration among proficient designers and other experts, such as coders.

According to the field observations, as the overlapped knowledge domains increased within teams, serendipity decreases, and the system becomes more robust. The team, being distant from serendipitiy, progresses through more specific steps and reaches the outcome faster. However, this process eliminates many alternative possibilities that could have been explored. In cases where knowledge domains do not overlap, serendipity increases, leading to a broader exploratory space and, the system becomes less robust. The team encounters numerous alternatives within serendipity and may obtain more creative options as a result. However, the design problem-solving process may require a longer time frame to accommodate this broader exploration. Establishing design rules from the beginning to achieve a legible design accelerates the design process but narrows down the exploration space. This does not imply that a design product emerging from an ambiguous design process in cases where knowledge domains do not overlap would be inherently illegible. On the contrary, in the absence of predefined rules describing the design product from the outset, the design process itself becomes the means of expressing the design idea.

In both situations, the teams had worked in different collaborative systems. In the first type, communication between the leader and the coder followed a hierarchical progression. Consequently, the coder's involvement in the design process from the architecture knowledge domain was limited. In the second case, the common language between the leader and the coder was design, and implementing the leader's requirements through computational tools became the sole responsibility of the coder. The coder was able to integrate his/her design knowledge into the process. Similar to what Burry (2011) mentioned, in both offices observed in this research , the design progressed within a hierarchical system, while computation was applied in a non-hierarchical manner in the second office.

While discussing the scripting culture, Burry (2011) notes that this new process lacks a fixed formula in office settings; instead, it exhibits variability. In the offices observed, team formation was consistently tailored for each project, adapting and evolving based on the specific project requirements and the participants' skill sets.

In areas where creativity takes the forefront and risks are less prominent, it appears that, as important as system robustness, sharing ideas and generating new ones are emphasized even more. Within this context, the use of different forms of representation and the transfer of ideas among these forms are seen as trigger for creativity.

The office leader of Office B, who does not know coding languages, consistently seeks confirmation of the produced version and its progress. This approach ensures that knowledge is shared effectively by monitoring the coders' outputs through representation systems other than coding. With each transfer, information is repeatedly represented, minimizing the risk of loss between the initial and final stages of information representation, which is the production process. These transfers also involve a form of translation, as the information is represented in different languages, yet it is crucial to maintain content accuracy as much as possible while one always seeks for novel interpretations. In the translation, what matters is continual search for novel ideas, emerging from previous shared ideas rather than simplistic effort to maintain accuracy of knowledge across different representational systems. This situation offers complementary view of robustness as it was formulated by Hutchins (1995a).

The organizational structure within teams can contribute to an increase in overlapped knowledge among team participants when supported by the diversity, versatility, and transferability of the representation systems they use in the design process. It is recognized that a deeper analysis of the organizational structure would offer a richer understanding of knowledge representation and dissemination. Yet within the scope of this thesis, the emphasis was not on the organizational structure of design teams.

Sawyer (2017) suggests that in a good collaborative system, participants engage in improvisation by deeply listening to others, adding their contributions, generating something new, and presenting it to others. This process creates an additive dynamic. On the other hand, in non-overlapping situations as presented by Hutchins (Hutchins, 1995a), experts have different knowledge domains and lack overlap, thus requiring a common language for deep listening and understanding. In cases where knowledge does not overlap, improvisation may occur through the intersections of other areas. In the cases presented in this dissertation, leaders without computational knowledge were able to engage in improvisation through transitions and diversity between novices with coding knowledge and the use of tools. In the context of distributed cognition, when considering overlapping and non-overlapping situations, it is suggested that systems can be either robust or not. In systems where non-overlapping is considered insubstantial, it is possible that there is less communication between individuals, which may lead to more improvisation. This, in turn, can foster the exloration of new paths to enhance creativity. By expanding the exploration space and seeking a common language, individuals aim to increase communication channels and enhance collaboration.

CHAPTER 8

CONCLUSION

"a single man cannot build a house"
Yinka Ilori

This dissertation presented an ethnographic study that was conducted to understand how tasks related to different expertise domains, including architectural design and computation, are distributed and how experts dynamically collaborate in a design process. To address this inquiry, this dissertation first presents an overview of distributed cognition (Hutchins, 1995) and design collaboration mechanisms within interdisciplinary teams that utilize computational design tools. The descriptions are based on a series of ethnographic observations spanning a total of two months with a deep focus on the design competition process in two architectural offices. Through field observations and interviews the study looked closely at two issues which were formulated as research questions: (1) what is the nature of collaboration in architectural design teams, and (2) how different expertise domains are distributed and how experts dynamically collaborate across design phases.

In the three core episodes, namely chapters 4, 5 and 6, summarizing and discussing the ethnographic observations, the first focused on the role of serendipitiy in the design exploration, the second on the explicitness/legibility of design ideas, and the third on the collaborative interdisciplinary design process in architectural design teams. The dissertation analyzed the situated interactions among team participants through interdisciplinary processes and representations, particularly incorporating the disciplines of architecture and computational design technology.

Chapter 4 presents how design ideas are explored and how computational tools are used to facilitate the exploratory process in teams using the computational design tool. The cases presented in the chapter establishes the positions of design exploration by computational tools in the sense of knowledge propagation among team participants with a particular focus on how serendipity is achieved through computational tools. The chapter presented how architects and coders have expanded their exploration processes and domains by utilizing tools and methods that enable them to broaden their scope. Their limited proficiency in these tools has made them more receptive to serendipitous discoveries while also providing them with a multitude of alternatives. The designers used the flexibility of tools to increase the space for exploration.

Chapter 5 presents how teams using computational tools and approaches make their design ideas explicit and legible to support collaborative communication. The cases presented in the Chapter 5 establish the significance of representational tools in reflecting design approaches and design solutions, where designers utilize computational tools. In this chapter, the transparency of the design concept within the team and/or outside the team, as well as the efforts of the teams to achieve it, are presented. Through legibility, teams tolerate the clarity of the design concept, improving the improvisation of internal team dynamics and processes.

Chapter 6 presents episodes of collaboration among team participants, focusing on the use of computational tools, the distribution of knowledge, and the multidisciplinarity of the team participants. The chapter considers the system as distributed and discusses the collaboration mechanisms in the observed teams in reference to the distributed cognitive systems theory of Hutchins. The chapter presents examples where leaders of design teams collaborated and worked alongside individuals with advanced computing skills to develop architectural design proposals. The establishment of a shared language facilitated consensus-building and design progress. However, it was noted that the presence of redundant knowledge domains (Hutchins, 1995) was crucial in fostering a resilient collaborative practice that could generate multiple alternatives.

Today, technological tools are used more and more in architectural design. Digital tools have taken a big place in many professions. These technological developments have also increased the number of multidisciplinary teams. This research shows that these multidisciplinary teams integrate computational design tools into their design processes together with a multitude of architectural representations.

Understanding how multidisciplinary teams can use these design tools provides an opportunity for more effective and robust collaborations between different disciplines. These findings can play an important role in the transformation and evolution of architectural design. In particular, it may be possible to achieve more effective and innovative results by increasing collaboration between design teams.

Architectural discipline, with its constantly changing and diversifying boundaries, has always incorporated and will continue to incorporate various other professional fields and specializations. With the involvement of different expertise in the design process, we can talk about a hierarchy that evolves and adapts according to the problem rather than a frozen hierarchy. For instance, in a design problem that is intended to be solved using a computational design approach and tools, an expert architect and a team member who specializes perhaps in coding but is not as specialized in architecture will have an equal say in the project, rather than a strict hierarchy.

In architectural education, when design studios and many other courses are conducted in an interdisciplinary manner, the convergence of different disciplines can lead to the development of new tools, the exploration and enhancement of new communication channels. Designing and arranging educational spaces to encourage interdisciplinary work environments, providing spaces and environments on campus that support these encounters and intersections, can foster the spontaneous emergence of numerous interdisciplinary productions especially given the ever increasing potential of computational technologies and tools.

In the future, it is evident that office structure will involve an increasingly diverse team composition in terms of disciplines. As a result, a team arrangement can emerge where hierarchy is reshaped for each task, and team participants contribute to the process at different stages and hierarchies. The hierarchical system, being redefined with each new project/task, thus leads to witnessing productions where the execution of work transitions from being attributed to a single individual to conveying a sense of collective production.

The field of architecture, with its ever-changing boundaries, has always incorporated and will continue to incorporate various other professional fields and specializations. With the involvement of different expertise in the design process, we can talk about a hierarchy that evolves and adapts according to the problem rather than a stable hierarchy. For instance, in a design problem that is intended to be solved using a computational design approach and tools, an expert architect and a team member who specializes perhaps in coding but is not as specialized in architecture will have an equal say in the project, rather than a fixed hierarchy.

8.1. Recommendations and Future Work

In this dissertation, the design processes of two architectural offices with expertise in the field of computational design in Turkey, were observed for a period of time in their authentic office environments. All team participants were architects with or without the skills and knowledge in coding. The research examines how knowledge domains from different disciplines are integrated into the design process and how experts of different knowledge domains collaborated. As a future research direction, teams involving a greater variety of knowledge domains could be investigated.

Being present in the offices as an observer during the research process may have hindered the team participants from feeling completely at ease, especially in the early stages. Taking on the role of a participant observer can provide an opportunity for participants to feel more comfortable and act more naturally, while also allowing for a deeper exploration of the process. Within the scope of the research, the algorithms developed in computational design tools have been partially shared by the offices. In an observational research study, having multiple researchers follow the teams instead of a single researcher can provide an opportunity to delve deeper into concurrently unfolding events. This approach allows for a more comprehensive examination of the process, as different researchers can focus on different aspects and provide a richer understanding of the observed phenomena.

As a researcher, not having coding knowledge has allowed for more open-ended and in-depth questioning. This lack of expertise in coding has led to a curiosity-driven exploration of the subject matter, enabling the researcher to ask broader and more probing questions. This approach may result in a deeper understanding of the research by approaching it from different perspectives and encouraging a more comprehensive investigation.The researcher has been granted access to the offices to observe the competition processes. However, observing long-term design projects with an implementation phase within the offices can provide a much more in-depth direction for the research. By observing such projects, the researcher can delve deeper into the intricacies of the design process, uncovering valuable insights and generating a more comprehensive understanding of the subject matter.

REFERENCES

Akin, Ö., & Akin, C. (1996). Frames of reference in architectural design: analysing the hyperacclamation (A-h-a-!). *Design Studies*, *17*(4), 341–361.

Andel, P. Van. (1992). Serendipity: "Expect also the Unexpected". *Creativity and Innovation Management*, *1*(1), 20–32.

Ball, L. J., & Ormerod, T. C. (2000). Putting ethnography to work: the case for a cognitive ethnography of design. *Int. J. Human-Computer Studies*, *53*, 147–168.

Besserud, K., Katz, N., & Beghini, A. (2013). Structural Emergence: Architectural and Structural Design Collaboration at SOM. *Architectural Design*, *83*(2), 48–55.

Brewer, J. (2003). Ethnography. In R. L. Miller & J. D. Brewer (Eds.), *The A-Z of Social Research: A Dictionary of Key Social Science Research Concepts* (pp. 100–103). SAGE Publications, Ltd.

Bucciarelli, L. L. (1988). An Ethnographic perspective on Engineering Design. *Design Studies*, *9*(3), 159–168.

Bucciarelli, L. L. (1996). *Designing Engineers*. The MIT Press.

Burry, M. (2003). Between Intuition and Process: Parametric Design and Rapid Prototyping. In B. Kolarevic (Ed.), *Architecture in The Digital Age: Design and Manufacturing* (pp. 210–230). Span Press, Taylor & Francis Group.

Burry, M. (2011). Cultural Defence. In *Scripting Cultures: Architectural Design and Programming* (pp. 27–71). Wiley Academy.

Callon, M. (1987). Society in the Making: The Study of Technology as a Tool for Sociological Analysis. In T. J. Bi.Jker, W. E., Hughes, T. P., & Pinch (Ed.), *The Social Construction of Technological Systems*.

Cheng, N. Y. W., & Kvan, T. (2000). Design Collaboration Strategies. *The Fifth International Conference on Design and Decision Support Systems in Architecture*, 62–73.

Clark, A., & Chalmers, D. (1998). The Extended Mind. *Analysis*, *58*(1), 7–19.

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

Copeland, S. (2019). On serendipity in science: discovery at the intersection of chance and wisdom. *Synthese*, *196*(6), 2385–2406. https://doi.org/10.1007/s11229-017-1544-3

Creswell, J. (2003). *Research design: qualitative, quantitative, and mixed method approaches*. Sage Publications.

Creswell, J. (2007). *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. Sage.

Cross, N. (2006). *Designerly Ways of Knowing*. Springer.

Cuff, D. (1992). *Architecture: The Story of Practice*. MIT Press.

D'souza, N. (2020). *The Multi-Skilled DesignerA Cognitive Foundation for Inclusive Architectural Thinking*. Routledge.

Denzin, N. K., Lincoln, Y. S., Denzin, N. K., & Lincoln, Y. S. (2005). *The SAGE Handbook of Qualitative Research* (3rd ed.). Sage Publications.

Doctors, S. I. (2015). Digitizing the Collaborative Divide Historical Problems with Non-Architect "Others." *The International Journal of Architectonic, Spatial, and Environmental Design*, *9*(3).

Donchin, M. (2013). Kahn, Komendant, and The Kimbell Art Museum: Cooperation, Competition, and Conflict. In G. Herbert & M. Donchin (Eds.), *The collaborators : Interactions in the architectural design process* (pp. 205–234). Ashgate.

Dreyfus, H. L. (1999). *What Computers Still Can't Do: A Critique of Artificial Reason*. The MIT Press.

Dyer, L., Power, J., Steen, A., & Wallis, L. (2021). Uncertainty and disciplinary difference: Mapping attitudes towards uncertainty across discipline boundaries. *Design Studies*, *77*(C), 1–24. https://doi.org/10.1016/j.destud.2021.101055

Eastman, C. (1969). Cognitive processes and ill-defined problems: A case study from design. *Proceedings of the International Joint Conference on Artificial Intelligence: IJCAI*, *69*, 669–690.

Eck, D. van. (2015). Dissolving the 'problem of the absent artifact': Design representations as means for counterfactual understanding and knowledge generalisation. *Design Studies*, *39*(C), 1–18.

Emerson, R. M., Fretz, R. I., & Shaw, L. L. (1995). *Writing Ethnographic Fieldnotes*. The University Of Chicago Press.

Fischer, G., & Ostwald, J. (2005). Knowledge communication in design communities. In R. Bromme, F. Hesse, & H. Spada (Eds.), *Barriers and Biases in Computer-Mediated Knowledge Communication* (pp. 213–242). Kluwer Academic Publishers.

Fiske, J. (1990). *Introduction to communication studies*. Routledge.

Foster, N., Sudjic, D., & De Grey, S. (2001). *Norman Foster and the British Museum*. Prestel.

Gabora, L. (2010). Revenge of the 'neurds': Characterizing creative thought in terms of the structure and dynamics of memory. *Creativity Research Journal*, *22*(1), 1–13.

Gabriel, G. C., & Maher, P. M. Lou. (2000). Computer Mediated Collaborative Design In Architecture: The Effects Of Communication Channels On Collaborative Design Communiation. In *Department of Architectural and Design Science Faculty of Architecture: Vol. PhD*. University of Sydney.

Galle, P. (1999). Design as intentional action: a conceptual analysis. *Design Studies*, *20*(1), 57–81.

Gigone, D., & Hastie, R. (1993). The common knowledge effect: Information sharing and group judgment. *Ournal of Personality and Social Psychology*, *65*(5), 959–974.

Goel, V. (1992). "Ill-Structured Representations" for Ill-Structured Problems. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*.

Goel, V. (1995). *Sketches of Thought*. The MIT Press.

Goldschmidt, G. (1991). The Dialectics of Sketching. *Creativity Research Journal*, *4*(2), 123–143. https://doi.org/10.1080/10400419109534381

Goldschmidt, G. (1994). On visual design thinking: the vis kids of architecture. *Design S*, *15*(2), 158–174. https://doi.org/10.1016/0142-694X(94)90022-1

Goldschmidt, G. (2014). Ubiquitous Serendipity: Potential Visual Design Stimuli are Everywhere. In J. S. Gero (Ed.), *Studying Visual and Spatial Reasoning for Design Creativity* (pp. 205–214). Springer.

Goldschmidt, G. (2015). Ubiquitous Serendipity: Potential Visual Design Stimuli are Everywhere. In J. Gero (Ed.), *Studying Visual and Spatial Reasoning for Design Creativity* (pp. 205–214). Springer. https://doi.org/https://doi.org/10.1007/978-94-017-9297-4_12

Goldschmidt, G., & Porter, W. L. (2004). *Design Representation*. Springer.

Groat, L. N., & Wang, D. (2013). *Architectural Research Methods* (Second). John Wiley & Sons, Inc.

Guilford, J. P. (1950). Creativity. *American Psychologist*, *5*, 444–454.

Guilford, J. P. (1973). *Characteristics Of Creativity*. Illinois State Office of the Superintendent of Public Ins.

Hammersley, M., & Atkinson, P. (2003). *Ethnography*. Routledge.

Heerwagen, J. H., Kampschroer, K., Powell, K. M., & Loftness, V. (2004). Collaborative knowledge work environments. *Building Research & Information*, *32*(6), 510–528.

Henderson, K. (1999). *On line and On Paper: Visual representations, visual culture, and computer graphics in Design Engineering*. MIT Press.

Heylighen, A., & Martin, G. (2004). That Elusive Concept Of Concept In Architecture A First Snapshot Of Concepts During Design. *Design Computing and Cognition*, 57–76.

Hight, C., & Perry, C. (2013). Introduction to Collective Intelligence in Design AD September–October 2006. In M. Carpo (Ed.), *The Digital Turn in Architecture 1992–2012The Digital Turn in Architecture 1992–2012* (pp. 188–207). John Wiley & Sons, Inc.

Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: A new theoretical foundation for human-computer interaction research. *ACM Transactions on Human-Computer Interaction*, *7*, 174–196.

Hutchins, E. (1991). The social organization of distributed cognition. In L. Resnick, L. B, M. John, & S. Teasley (Eds.), *Perspectives on Socially Shared Cognition* (pp. 283–307). American Psychological Association.

Hutchins, E. (1995a). Cognition in the Wild. In *MIT Press*.

Hutchins, E. (1995b). How a Cockpit Remembers Its Speeds. *Cognitive Science*, *19*, 265–288.

Hutchins, E. (2004). Distributed Cognition. In J. Smelsner (Ed.), *International Encyclopedia of the Social & Behavioral Sciences* (pp. 2068–2072). Elseiver.

Hutchins, E. (2006). The distributed cognition perspective on human interaction. In N. J. Enfield & S. C. Levinson (Eds.), *Roots of Human Sociality: Culture, Cognition and Interaction* (pp. 375–398). Bloomsbury Academic.

Hutchins, E. (2014). The cultural ecosystem of human cognition. *Philosophical Psychology*, *27*(1), 34–49.

Kahn, L. (1930). The Value and Aim in Sketching. In A. Latour (Ed.), *Louis I. Kahn, Writings, Lectures, Interviews* (pp. 10–14). Rizzoli International Publications, INC.

Kamprath, M., & Tassilo, H. (2019). Serendipity and innovation: beyond planning and experimental-driven exploration. In A. B. J. Chen, E. Viardot, & P. K. Wong (Eds.), *The Routledge Companion to Innovation Management* (pp. 343–360). Routledge.

Kasali, A., & Nersessian, N. J. (2015). Architects in interdisciplinary contexts: Representational practices in healthcare design. *Design Studies*, *41*(156), 205–223. https://doi.org/10.1016/j.destud.2015.09.001

Katz, N., Schwinn, B., & Tobias, K. (2013). Interacting with The Model. In B. Peters & T. Peters (Eds.), *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design* (pp. 80–91). Wiley.

Kirsh, D. (2008). Distributed cognition: A methodological note. In I. E. Dror & S. Harnad (Eds.), *Cognition Distributed: How cognitive technology extends our minds*. John Benjamins Publishing.

Kirsh, D. (2010). Thinking with external representations. *Ai & Society, 25*(4), 441–454.

Knight, T., & Stiny, G. (2001). Classical and non-classical computation. *Information Technology*, *5*(4), 355–372.

Kolarevic, B. (2003). Information Master Builders. In *Architecture in the digital age - Design and Manufacturing* (pp. 56–62). https://doi.org/10.1007/s00004-004-0025-4

Kolarevic, B. (2013). Parametric Evolution. In B. Peters & T. Peters (Eds.), *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design* (pp. 50–59).

Latour, B. (1996). On actor-network theory: A few clarifications. *Soziale Welt*, *47. Jahrg*, 369–38.

Latour, B. (2005). *Reassembling the Social: an Introduction to Actor-Network-Theory*. Oxford University Press.

Laurel, B. (2003). *Design Research: Methods and Perspectives*. MIT Press.

Lave, J. (1988). *Cognition in Practice: mind, mathematics and culture in everyday life*. Cambridge University Press.

Law, J. (2003). *Notes on the Theory of the Actor Network: Ordering, Strategy and Heterogeneity*. The Centre for Science Studies, Lancaster University. http://www.comp.lancs.ac.uk/sociology/papers/Law-Notes-on-ANT.pdf%0APublication

Loukissas, Y. (2009). Keepers of the Geometry. In S. Turkle (Ed.), *Simulation and Its Discontents* (pp. 153–170). The MIT Press.

Lynch, K. (1960). *The image of the city*. MIT press.

Lyon, E. (2005). Autopoiesis and Digital Design Theory: CAD Systems as Cognitive Instruments. *International Journal of Architectural Computing*, *317–334*.

Lyon, E. (2011). Emergence and Convergence of Knowledge in Building Production: Knowledge-Based Design and Digital Manufacturing. In T. Kocatürk & B. Medjdoub (Eds.), *Distributed Intelligence in Design* (pp. 71–98). Wiley-Blackwell.

Marr, D. (2010). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT press.

McComb, C., & Jablokow, K. (2022). A conceptual framework for multidisciplinary design research with example application to agent-based modeling. *Design Studies*, *78*(C).

Nardi, B. A. (1995). Studying context: a comparison of activity theory, situated action models, and distributed cognition. In A. N. Bonnie (Ed.), *Context and consciousness* (pp. 69–102). the MIT press.

Nemeth, C. J., & Nemeth-Brown, B. (2003). Better than Individuals?: The Potential Benefits of Dissent and Diversity for Group Creativity. In P. B. Paulus & B. A. Nijstad (Eds.), *Group Creativity: Innovation Through Collaboration*. Oxford University Press.

Nomura, S., & Hutchins, E. (2006). Study for Bridging between Paper and Digital Representations in the Flight Deck. *Computer Supported Cooperative Work (CSCW)*.

Norman, D. A. (1991). Cognitive artifacts. In M. C. John (Ed.), *Designing interaction* (pp. 17–38). Cambridge University Press.

Olsen, C., & Namara, S. M. (2014). *Collaborations in Architecture and Engineering*. Taylor & Francis.

Oxman, R. (2002). The thinking eye: visual re-cognition in design emergence. *Design Studies*, *23*(2), 135–164.

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

Perry, M., & Sanderson, D. (1998). Coordinating joint design work: the role of communication and artefacts. *Design Studies*, *19*(3), 273–288. https://doi.org/10.1016/S0142-694X(98)00008-8

Peters, B. (2013). Realising the Architectural Idea: Computational Design at Herzog & De Meuron. *Architectural Design*, *83*(2), 56–61.

Porter, W. L., & Goldschmidt, G. (2001). *Design representation*. Springer.

Poulsgaard, K. S., & Malafouris, L. (2017). Models, Mathematics and Materials in Digital Architecture. In S. J. Cowley & F. Vallée-Tourangeau (Eds.), *Cognition Beyond the Brain: Computation, Interactivity and Human Artifice* (Second). Springer London.

Rittel, H. (1971). Some Principles for the Design of an Educational System for Design. *Journal of Architectural Education*, *25*(1/2), 16–27.

Ritzer, G. (2004). *Encyclopedia of Social Theory I* (G. Ritzer (ed.)). SAGE Publications.

Runco, M. A., & Pritzker, S. R. (2013). Encyclopedia of CREATIVITY. In *Academic Press: Vol. VOLUME 1-A*. https://doi.org/10.1017/CBO9781107415324.004

Sabin, J. E. (2013). Matrix Architecture. In B. Peters & T. Peters (Eds.), *Inside*

Smartgeometry: Expanding the Architectural Possibilities of Computational Design (pp. 60–71). Wiley.

Sawyer, K. (2010). The interdisciplinary study of creativity in performance. *Creativity Research Journal*, *11*(1), 11–19.

Sawyer, K. (2017). *Group Genius : The Creative Power of Collaboration*. Basic Books Inc.

Sawyer, K., & DeZutter, S. (2009). Distributed Creativity : How Collective Creations Emerge From Collaboration. *Psychology of Aesthetics, Creativity, and the Arts*, *3*(2), 81–92. https://doi.org/10.1037/a0013282

Sawyer, R. K. (2003). *Group Creativity: Music, Theater, Collaboration*. Lawrence Erlbaum Associates, Inc.

Schön, D. A. (1991). *The Reflective Practitioner: How Professionals Think in Action*. Ashgate Publishing Limited.

Schön, D. A. (1992). Designing as Reflective Conversation with the Materials of a Design Situation. *Research in Engineering Design*, *3*, 131–147.

Shaw, G. B. (2010). A cognitive account of collective emergence in design. *CoDesign*, *6*(4), 225–243.

Simon, H. A. (1996). *The Sciences of the Artificial*. the MIT press.

Smith, R. S. (2017). *Fabricating the Frank Gehry Legacy: The Story of the Evolution of Digital Practice in Frank Gehry's office*. Amazon Books.

Sommer, B., & Sommer, R. (1997a). Interviews. In R. Sommer (Ed.), *A Practical Guide to Behavioral Research: Tools and Techniques* (Fourth). Oxford University Press.

Sommer, B., & Sommer, R. (1997b). Observation. In B. Sommer & R. Sommer (Eds.), *A Practical Guide to Behavioral Research Tools and Techniques* (Fourth). Oxford University Press.

Stasser, G., & Birchmeier, Z. (2003). Group Creativity and Collective Choice. In P. B. Paulus & B. A. Nijstad (Eds.), *Group Creativity: Innovation Through Collaboration* (Vol. 1, pp. 85–109). Oxford University Press.

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.

Strehlke, K. (2009). Digital Technologies, Methods, and Tools in Support of the Architectural Development at Herzog & de Meuron. *The 29th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 26–29.

Suchman, L. A. (1987). *Plans and situated actions: the problem of human-machine communication*. Cambridge University Press.

Suwa, M., Gero, J., & Purcell, T. (2000). Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. *Design Studies, 21*(6), 539–567.

Szalapaj, P. (2005). The Digital Design Process in Contemporary Architectural Practice. *The Digital Design: The Quest for New Paradigms: 23rd ECAADe Conference*, 751–759.

Terzidis, K. (2003). *Expressive Form: A conceptual approach to computational design*. Spon Press.

Terzidis, K. (2006). *Algorithmic Architecture*. Elsevier.

Turkle, S. (2009). *Simulation and Its Discontents* (J. Maeda (ed.)). The MIT Press.

Verzijl, W. I. (1997). Introduction. *ARCHIDEA*, *Autumn*(XVI), i–ii.

Walker, C. (2014). *GAD's Serra Gate on Display for Istanbul Design Week*. https://www.archdaily.com/570837/gad-s-serra-gate-on-display-for-istanbul-design-week

Wilson, R. A., & Keil, F. C. (1999). *The MIT Encyclopedia of the Cognitive Sciences* (R. A. Wilson & F. C. Keil (eds.)). The MIT Press. https://doi.org/10.1016/S0004-3702(01)00095-9

Woodbury, R., & Burrow, A. (2003). Notes on the Structure of Design space. *International Journal of Architectural Computing*, *1*(4), 517-532.

Yaneva, A. (2005). Scaling Up and Down: Extraction Trials in Architectural Design. *Social Studies of Science, 35*(6), 867–894. https://doi.org/10.1177/0306312705053053

Yaneva, A. (2009). *Made by the office for metropolitan architecture: An ethnography of design*. 010 Publishers.

Yu, R., Gero, J., & Gu, N. (2015). Architects' Cognitive Behaviour in Parametric Design. *International Journal of Architectural Computing*, *13*(1), 83–101.

APPENDIX A

IRB APPROVAL

Altuğ Kasalı, co-advisor to this dissertation, had the documents submitted for the Ethics Committee review at Izmir University of Economics.

IZMIR UNIVERSITY OF ECONOMICS

Number : B.30.2.IEÜ.0.05.05-020-055 07.11.2017

Subject : Ethics Committee Approval

Dear Dr. Altuğ Kasalı,

The Izmir University of Economics Research Ethics Committee has recently considered your application for the research protocol titled "Distributed Expertise of Computational Practices in Architectural Design Teams." The sub-committee assigned to process the proposed protocol has finalized the review on 07.11.2017. The proposed project has been approved on the basis of the information contained in the application.

If any complaints or other evidence of risk should occur, or if there is a change in plans, the Izmir University of Economics Research Ethics Committee must be notified via an amendment.

Sincerely,

Professor Dr. Filiz Başkan

Research Ethics Committee Officer

Sakarya Cad. No: 156, 35330 Balçova - IZMIR - TURKEY • Ph: +90 232 279 25 25 pbx • Fax: +90 232 279 26 26
e-mail: info@izmirekonomi.edu.tr • www.izmirekonomi.edu.tr

APPENDIX B

SAMPLE TRANSCRIPT

		Transcript_ MVI-0049 _02_08_2017
		Video Record: MVI-0049- Participants O_aC_1 and O_aL: - Location: Office A office - Duration: 00:32:42
00:14:51	O_aC_1	aslında hem zeminde akıyor hem de yukarda gidiyor o şu an için. Zeminde de gidebiliyorsunuz buradan karşıdan karşıya. En azından burada zeminden vazgeçiyor olacağız.
00:15:04	O_aL	niye? Geçebilirsin yani. Aslen bunun amacı üst katta ulaşımı sağlamak içindi değil mi? Dünkü konuşmalarımız da. Ama bunun nasıl bir etkisi oldu buraya toplasan zaten iki tane delik açıyor burada. Dünkünü açalım, burada hali hazırda zaten bir tane burada vardı. Burada belki… peyzaj bakmamız gerekecek tabi. Ona göre bir tane daha açmamız gerekecek tabi. Onu anlamıyorum buradaki mantık… burada ne değişti yani. Bambaşka değişmiş çok değişmiş…
00:16:06	O_aC_1	evet evet…
00:16:13	O_aL	şimdi kuralı olmazsa random olunca biz öylesine koymuşuz da işte dağıtmışız olur. Hiçbir mimar onu okuyamaz. okuyamadığı şeyi de yorumlayamaz. Öyle söyleyeyim size. Baktığı anda anlaması lazım. Baktığım anda ben bugün anlayamadım. Anlatabiliyor muyum? Kuralını sebebini anlayamadım. Dün o halde siz gösterdiğinizde direk ne düşündüğünüzü anlayabiliyordum daha anlatmadan. Ama burada kaybolmuş o. Çok random olmaya başlamış. Seninkinde de öyle olmuş (O_aIA_1'e söylüyor). Çok detaya girince o ana fikri biraz geri plana atmışsınız. Bunu görüyorum, şimdi onu bir kurtarmamız lazım bakalım ne yapalım.
00:17:04	O_aC_1	bunda evet yani her grup içinde şuan 3 kuralım var gibi oldu. Ama diğerinde öyle yapmamıştım bütün hepsini aynı kuralda yapmıştım.
00:17:13	O_aL	şu biraz şuna benziyor sonra mirror oluyor bir şeyler var değil mi?
00:17:15	O_aC_1	evet küçük şeyle yapıyorum aslen ama. Evet birbirinden farklı biraz büyüyorlar.
00:17:27	O_aL	bence büyüme yönleri yanlış. Sen burada büyütüyorsun ama burada hedefin courtyard yaratmak büyütürken.
00:17:38	O_aC_1	ve bir şekilde sirkülasyona bağlamaya çalışıyorum.
00:17:41	O_aL	tamam ama hep böyle courtyard yaratıyorsun.
00:17:44	O_aC_1	evet

APPENDIX C

CODING GUIDE

Coding guide for emergent categories and super ordinate categories:

Design Tools			
1	**Making the design/form open for further manipulation**	**Definition**	Instances where individuals or participants manipulate representations to make them accessible.
		Sample (O_bC_2)	Tabi ki tekrar çiziyorum ve bana gelen feedback le o değişince bütün modeli update etmiş olacağım.
2	**Interaction through digital tool**	**Definition**	Instances in which design tools facilitate interaction between participants
		Sample (O_bC_2)	Böylelikle zaman kaybetmemiş oluyorum işte kesitlerim en son ki durumda bu şekilde oluyor. Biraz da dönmüş hali sonda da atıyorum böyle ve bunu ben işte burada mesela değiştirebiliyorum yani isteğe göre hani ihtiyacımıza göre ondan sonra açılarını da buradan ayarlayabiliyorum.
Issues of Form Finding in Design Process			
3	**Form Finding through rule-based strategies**	**Definition**	Instances where designers follow rule-based approaches with or without computational design tools
		Sample (O_aC_1)	bu modüler bir accomodation birimleri yapmaya çalışıyoruz. İlk başta baya random bir yerleşim yapmıştık ilk denememizde. Şimdi biraz daha kurallı unitler haline getirip onları tekrarlıyoruz. Biraz değiştirerek tabi. Genel olarak amaçlarımızda hem böyle kendilerine özel avlular yapmak, hem toplu kullanacakları avlular yapmak, bir ikincisi de şöyle bir (eliyle aksları tarifleyerek gösteriyor) aks akışı yaratmaya çalışıyor olacağız burada yarattığımız boşluklara göre geçiş aksları. Ona göre yerleşmeye çalışıyoruz.

4	**Introducing sophistication in design**	**Definition**	Instances where participants introduce further sophistication and complexity in design
		Sample (O_bL)	E tabi herkes grid daha algılayabildikleri şeyin üzerine gidiyor. Mesela bu da hem dış hem iç mekân düzeni açısından güzel yapılmaz bir şey değil. Ama bizde dedik ki böyle bir fırsat varken biraz daha ileriye götürebilecek hemen çözümlenmeyecek hemen algılanmayacak bir şeyin üstüne gidelim.
5	**Motivation on extraordinary/complicated design form**	**Definition**	Instances where participants create complicated forms.
		Sample (O_aC_1)	rotate dahi etmeden sadece çoğaltıyorum ve yeterince karışık oluyor.
		Sample (O_bTL_2)	ama O_bL bunun bu kadar basit olmasını istemeyecektir.
6	**Experimentation in form finding**	**Definition**	Instances in which participants refer to their studies of form finding
		Sample (O_bL)	Bunun sistemi aslında bununla ilgili bir deney yaptık. Gene bu parçaların bir araya gelmesinden bunu nasıl stüktüre ederiz diye.
7	**Manual interferences on design process**	**Definition**	Instances where participants manually interfere with automatic form finding system.
		Sample (O_bC_3)	Evet belki manuel olarak aşağıları kendimiz ekleriz blokları.
8	**Capacity of tool control**	**Definition**	Instances where digital tools results with unpredictable forms
		Sample (O_bTL_2)	işte hep karşılaşılan sorunlardan bir tanesi cevaplardan bir tanesi 'bunu böyle çekince böyle oldu' tamam böyle oldu da yani onu çeken sensin öyle olacağını bilmesen bile

			öyle olduktan sonraki kısmı sorgula ne oldu bir sor…
9	**Unexpected discovery of form**	**Definition**	Instances where computational design tools present unforeseen formal outputs
		Sample (O_bTL_2)	Bence bu parametrik tasarım kısmındaki en büyük sorun o tesadüflere bırakıldığı zaman, tabi ki tesadüflerden de çok güzel şeyler çıkarabilirler ama daha önce ne olduğunu olacağını tahmin etmeden bilmeyebilirsin tabi ama tahmin etmeden ortaya çıkan şey, bence yarıdan daha fazlası başarısızlıkla sonuçlanıyor. Çünkü senin başta koyduğun ilkeler doğrultusunda gitmemeye başlıyor. Kaçıyor yoldan sapmaya başlıyor.
10	**Prioritizing structural design over form finding**	**Definition**	Instances in which participants assess structural features in design representations.
		Sample (O_bC_1)	ilk önce strüktürel doluluk boşluk strüktür formun bize getirdiği hafıza bunlar her şey çok önemliydi. Programın mesela 16 metrelik akslar core ne olacak taşıyıcılar bunların her biri kriterdi.
11	**Motivation to integrate computational methods**	**Definition**	The persistent intention to integrate computational methods in form finding
		Sample (O_bL)	İşte taşıyıcı duvar ve taşıyıcı sistemin bu şekilde evrilmesi nasıl oluşur, 3. boyutta nasıl oluşur… bu Wolfram dünyasıyla 10 yıldır uğraşıyoruz. Biz bunda çok arkadaşımız buradan ayrılan çok arkadaşımızda bu konuyla uğraştılar.
12	**Employing a formula in form finding**	**Definition**	Instances where participants use mathematical formulas in form creation
		Sample (O_bC_1)	formülün getirdiği şey. Tarama sistemi, x2 nin nasıl böyleyse x3 sıfırla bir arası şöyle bir şeyse, bu da formül onu getiriyor.

13	**Aesthetic or stylistic formal preferences**	**Definition**	Instances where participants offer their aesthetic or stylistic preferences in assessing design representations
		Sample (O_bL)	bir de bunda binalar daha iyi bunu biraz daha yumuşatırsanız o zaman.
		Sample (O_bTL_2)	şimdi bunun kenarlarını yuvarlat O_bL'nin istediği gibi ortada ki çekirdekleri iki kat daha yükselt onlara ikişer kat daha ekle bütün hepsinden daha yüksek olsun
Intuition in Computational Design			
14	**Intuitive manipulations of digital models**	**Definition**	Formal manipulation of digital models based on design intentions.
		Sample (O_bC_3)	Çok bol dönersek hani bu devam edilen süregelen şey belki buradan koparız hani bu ikisi bir grup olur bu ikisi bir grup olur.
		Sample (O_bC_3)	Böyle tam smooth değil yani.
15	**Form Finding practices**	**Definition**	Instances where participants follow local or global strategies in manipulation of the form.
		Sample (O_bTL_1)	Aynen aynen bütünden gidip yontarak gitmek belki de daha doğru olacak.
Time in Design Process			
16	**Advantages of computational tool**	**Definition**	Instances in which participants refer to the advantages of computational tools in time and workload management
		Sample (O_bC_3)	Tabi ki tekrar çiziyorum ve bana gelen feedback le o değişince bütün modeli update etmiş olacağım. Böylelikle zaman kaybetmemiş oluyorum işte kesitlerim en son ki durumda bu şekilde oluyor. biraz da dönmüş hali sonda da atıyorum böyle ve bunu ben işte burada mesela değiştirebiliyorum yani isteğe göre hani

			ihtiyacımıza göre ondan sonra açılarını da buradan ayarlayabiliyorum.
17	**Schedule constraints in design decision**	**Definition**	Instances in which participants refer to schedule constraints
		Sample (O_aL)	Sonuçta senin burada kurduğun kuraldan geliyor ama o okunuyor. Bu duvar etkisini güçlü bulmamızın nedeni zaten birimleri hep böyle kullanmışşın çoğunu yani. Sonra uzamak için bu tarafa dikeyleri kullanıyorsun çoğunlukla ve dikeyleri okuyoruz buraya giderken. Dikeyden kastım bu yönde olan... burada şöyle giderken dönüyor hep courtyard yapıyor…
18	**Adopting a particular design method**	**Definition**	Instances where participants refer to their reasoning process in adopting in particular design method.
		Sample ($OaIA_l$)	…Bence daha kurallı bir şeye dönmemiz gerekiyor. Çünkü bunlarla çok uğraşırız. Vaktimiz yetmez.
Making The Design Process Transparent			
19	**Making Design Moves Transparent**	**Definition**	Instances where participants make design moves and decisions transparent.
		Sample (O_bL)	Hangi mevcut durumun ben görmüyorum burada! bak bir kere dedim ki mesela evolution gösterin bak tak burada dersin sonra buna atlarsın sonra buna atlarsın sonra buna atlarsın burada öyle bir şey yok ki yani sebep-sonuç ilişkisi kopuk.
Representational Practices			
20	**Using multiple design representations**	**Definition**	Instances where designers feel the need to assess and develop their work through a variety of media.
		Sample (O_aL)	peki bana üçü bas yediyi bas bide beşi bas onları bambaşka şeylerle değerlendireceğiz tamam mı üç beş yedi başka var mı burada

			göstermek istediğin onları A3'e bas üzerinden eskiz yapacağız senin tamam mı
21	**Comparison of representational systems**	**Definition**	Instances where participants compare tools according to their capacities.
		Sample (O_bL)	Bu işte! bu kadar oluyor! (eskizi işaret ediyor) Bu tavan şiirsel de bir işe yaramaz bunlar çok şiirsel hani böyle hani two değil bu. Bu 3D bu da 2 değil ama 3D bunu bu hale getirdikten sonra 3D printerde bunu elde edebiliyoruz. Bunun elde edilmişlerini gördün.
		Sample (O_bTL_2)	Karalamaya başladığın zaman o kâğıdın üzerinde o çizgiler birden üçüncü boyutta şekilleniyor. Ekranda o his gelmiyor bir türlü.
Digital Collaboration Mechanisms			
22	**Having an archive at one's disposal**	**Definition**	Instances when designers use scripts from their own and collective archives to provide design solutions.
		Sample (O_bC_3)	… ben zaten genelde bir script yazdıktan sonra bir yere kaydediyorum, arşivliyorum. Sonra başka bir yerde ihtiyacım olunca geri çağırıyorum onu böylelikle…
23	**Coordination through tools**	**Definition**	The way of collaboration of the teams in digital work environment.
		Sample (O_aC_1)	O_aC_1: … Bunu 3 koldan hallettik, ben ayrı O_aIA_1 ayrı O_aIA_2 ayrı ve şu an serverımızda bir şey oluşturduk nasıl diyeyim… I: ortak bir dosya gibi mi? O_aC_1: research ortak dosyamız var.
Design Conceptualization			
24	**Having general design approach**	**Definition**	Team participants' problem-solving method in design according to teams' design understanding.

		Sample (O_bL)	… Dolayısıyla bizim binalarımız da gelecekte fonksiyon değişiklikleri olursa fonksiyon değişikliğine karşı kolay değiştirilebilecek flexible bir yapı kabuğu haline gelebilir mi. Bir yandan da bunu araştırıyoruz. Bir yandan da buna bakıyoruz. Yani o kabuk tek başına içinde bir şey olmadan da güzel durur mu yani sanki durur gibi maketleri ve diğer şeyleri
Collaborative Practices in Design Process			
25	**Coordination strategies in collaborative work**	**Definition**	Instances in which planned or unplanned meetings to achieve coordination between participants who are ether internal or external
		Sample (O_aC_1)	her akşamda 1-1.5 saat minimum TL ile büyük toplantımız oluyor.
26	**Shared Approaches and intentions within teams**	**Definition**	Instances where participants refer to their team's shared design approach
		Sample (O_bL)	…bizim ofiste yukarıda tartışmaların içinde duymuşsundur nesne tasarlamak değil bizim amacımız öncelikle bunun içinde nasıl yaşarız nasıl mest ederiz. Bize yarayacak olanları nasıl oluştururuz çekiştiririz…
27	**Interdisciplinary collaboration**	**Definition**	Collaboration of different participants with different capacities of knowledge and or skill
28	**Engaging External Parties in design process**	**Definition**	Instances where participants refer to their collaboration with external parties
		Sample (O_bL)	Yapımı için firmalar ile görüştük. Mühendis takımı ile bir organizasyon yaratmak üzere işte facade engineering için. Kimler ile ne yapacağımız belli.
29	**Client Engagements**	**Definition**	Instances in which clients influence design decision making

		Sample (O_bC_1)	…rektörün (costumer) bizden beklentisi bir landmark yaratmaktı…
Precedents			
30	**Sources of inspirations**	**Definition**	Instances where participants refer to a project or architect's style in order to describe an idea
		Sample (O_bTL_2)	Yani sen Zaha Hadid'in projesinde baktığın zaman nasıl o akışkanlık hiç kaybolmuyor nerede olursan ol o sürekli şeyi yukarıdan doğru çıkan şeyi nasıl dönüyor musun inceliyor musun ne yapıyorsan yap o fludity i bozma.
31	**Reference to precedents**	**Definition**	Instances where participants refer to a precedent
		Sample (O_bL)	…. Mesela Taichung'da şeydekinde Toyo Ito'nun projesinde o da bu sistemle çalışıyor…
32	**Contextualizing design ideas**	**Definition**	Instances where participants relate their design work to precedents or existing approaches in architecture.
		Sample (O_bL)	…cam cepheyi diyelim biraz içeri çekip bir revak sistemi gibi çözdük aslında bütün bunlar klasik mimari de olan şeyler…
Budget Issues			
33	**Prioritizing design idea over budget constraints**	**Definition**	Instances where participants prioritize a design idea over the cost of construction
		Sample (O_bTL_2)	O_bTL_2: işte bazı şeylerin bütçesi olmuyor.Yani… I: bunu bütçeyle eşleştirmemek lazım. O_bTL_2: eşleştirmemek lazım. Çünkü her bakış açısı bütçeyle kısıtlandığı zaman e o zaman hakikaten kutu kutu pense. Başka bir şey yok. En ucuzunu yapacaksak o. Onun bile yapılabilirliği sırasında bir sürü şey deneyimleyebilirsin. O da ayrı mesele. Onu bile yapmıyoruz.

34	**Material aspects of design**	**Definition**	Instances where participants refer to materials to be consider in design.
		Sample (O_aIA_1)	bence yerel malzeme odaklı… malum taş çıkıyor, işçisi de varmış. Hem taş işçisi kazansın… hem şu bimslerden ötürü düşünelim. Oradan da bims kazansın.
Linking Parameters with Each Other			
35	**Introducing programmatic concerns in design**	**Definition**	Instances in which participants refer to program related activities within developing design
		Sample (O_bL)	Çatı kısmında kitabı alıp çıkabileceğin bir bahçe terası var. Günümüzün işte organik küçük ölçekli yetiştirilebilecek her şey için kimisi güneş gören kimisi görmeyen göğüs üstünde kabuklu bir sistem olan experimental bir alan.
Design Approach			
36	**Introducing Legibility**	**Definition**	Instances where participants refer to their efforts to achieve legible schema or process
		Sample (O_aL)	şimdi kuralı olmazsa random olunca biz öylesine koymuşuz da işte dağıtmışız olur. Hiçbir mimar onu okuyamaz. okuyamadığı şeyi de yorumlayamaz. Öyle söyleyeyim size. Baktığı anda anlaması lazım. Baktığım anda ben bugün anlayamadım. Anlatabiliyor muyum? Kuralını sebebini anlayamadım. Dün o halde siz gösterdiğinizde direk ne düşündüğünüzü anlayabiliyordum daha anlatmadan. Ama burada kaybolmuş o. Çok random olmaya başlamış.
Research in Design			
37	**Conducting Research**	**Definition**	Instances where participant refer to their research efforts in relation to developing design work

		Sample (O_aC_1)	…Ve research aşaması da sürekli devam ediyor. Artık bitmiş bir şey değil, hala daha var.
		Sample (O_aC_1)	burada çok şey var karakteristik olarak çardak gibi yapılar var çok fazla piknik yapılıyormuş. Bank ve üstü kapalı şeyler oluyor ya. Karakteristik olarak bir bir TL'nin gözlemlediği bir şey bu site'da. Dolayısıyla bunu bir patlamış exploded bir şey olarak bunu site'a dağıtmak gibi bir fikrimiz var…
Design Explorations			
38	**Structured and unstructured explorations in design**	**Definition**	The way of design of the teams through the phases of design process as expected or unexpected explorations.
		Sample (O_aC_1)	şöyle birimlerimi oluşturdum. 2lik 4lük 6lık yaptım. 8lik yapamadım. Şey gelmedi hem büyük geldi hem kullanım açısından hep eğreti durdu ikincisinin yanına gelirken. 6 çok idealdi hep. Neden bilmiyorum. Bu yönde gridleri yavaş yavaş yerleştirmeye çalıştım şu an hiç rotate dahi etmedim bir birimimi. Olduğu gibi duruyorlar. Ama bunlar bile gayet karmaşık duruyorlar. Şeyde de araya bir kaç 4lük 2lik serpeceğim, yerleşime göre. Sonra akslara göre biraz ayıracağım v e eğitmenler yerleşecek falan filan
		Sample (O_bC_3)	aaa! birazcık olmaya başladı ha ne dersiniz! bak buraya kadar gelmiştik bak baştan alıyorum sana çok hızlı divide etmesini yaptık mesela belirli okey sonra bunu düzelttim o fix yöntemiyle sonra böyle panellere bölüyorum sonra eksiltiyorum sonra da birbirinin içine geçirtiyorum ama bunu mesela sadece horizontal olarak scale

			yap diyeceğim olacak aslında şimdi her yönüne yapıyorum scale 1d var mı 2d mi artık ney…volume var curve istiyor bu ya da bunların kesişen yerlerini trim et diyeceğim ama o da istediğimiz bir şey değil sanki
		Sample (O_bC_1)	I: Bu bir buçuk yıldır denediğiniz aradığınız şeylere dair söyleyebileceğin şeyler var mı? Yani neydi, ne deniyordunuz? O_bC_1: mesela şuradaki şeyle [maketi gösteriyor] Berlin'de yaptığımız enstalasyonla o strüktür o ortaya kitaplık ve strüktür çok farklı şeyler değil aynı. Ya da yıllardır form finding yaptığımız o matematiksel formüllerle (anlaşılmıyor) en iyi çalışan sistemin.. o yıllarda kullandığımız şeyi orada kabuk olarak kullandık, oraya evirttik. Hepsi uzun bir araştırmanın orada kullanılmasından ibaret.

APPENDIX D

INTERRATER RELIABILITY PROCESS INSTRUCTIONS

INSTRUCTIONS TO REVIEWERS

1. Introduction
2. Research questions
3. Coding instructions
4. Coding guide
5. Sample transcript

1. INTRODUCTION

This study focuses on how information is produced and distributed among experts who collaborate in architectural design processes. The research also involves in how knowledge is transferred through designers, tools, and representations, describes the role and tasks of each team participant, and the nature of the interactions within interdisciplinary teams. The research uses qualitative methods in analyzing the field data collected in-situ. The study inquires design teams' communications, knowledge transferring approaches, and representation techniques in the design process. The sample provided below belongs to the data set that was gathered through field methods including observations and interviews. The interrater reliability process involves the analysis of a selected interview excerpt following the coding guide presented below. In order to validate coding and analysis processes, the rater is being asked to participate in a series of sessions involving coding (individual), discussion and evaluation (collective).

2. RESEARCH QUESTIONS

The main research question and sub-questions that are pursued in this research are;

"Throughout the different stages of the architectural design process, how different experts are distributed and dynamically collaborate in a design process and how the knowledge distributed among them?"

1. What is the nature of collaboration in architectural design teams?

2. How different experts are distributed and dynamically collaborate across design phases?
3. What are the representation systems in architectural design teams? How do design team participants employ different representation systems in executing particular tasks?
4. How do interdisciplinary teams generate and coordinate representations collectively in the context of architectural design?

3. CODING INSTRUCTIONS

The attached document provides a guide to the set of categories –made up of codes used by researchers- in analyzing notes and transcripts from interviews and meetings. The guide includes the descriptions emergent categories and their higher-level categories created in the study. The rater is initially asked to read the guide to get familiar with the nature of categories generated. The section following the coding guide includes a sample transcript of a design meeting with several participants (architects, coder, and landscape architect) to clarify, discuss and resolve the issues presented. The meeting was recorded during the design development phase of a competition project. The reviewer is asked to read the transcript carefully and use the provided codes and any other additional codes that he/she sees appropriate. The reviewer is asked to carefully mark the segments of transcriptions and indicate the associated code. 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.

CURRICULUM VITAE

Name Surname: Livanur ERBİL ALTINTAŞ

EDUCATION:

PhD in Architecture: Izmir Institute of Technology (2023) Thesis: Distributed Expertise of Computational Practices in Architectural DesignTeams.

M.Sc. in Arch: Izmir Institute of Technology (2013) Thesis: Collaboration in Architectural Design Competitions.

B. Arch: Girne American University (2009)

PROFESSIONAL EXPERIENCE:

Lecturer (Part-time), Faculty of Fine Arts and Design, Department of Architecture, Izmir University of Economics.(Feb. 2020 - current)

Lecturer (Part-time), Manisa Celal Bayar University Faculty of Fine Arts, Design and Architecture, Department of Architecture (Feb. 2021 - Jan. 2022)

Research Assistant, Faculty of Architecture, Department of Architecture, İzmir Institute of Technology. (Feb. 2013 - July 2019)

PUBLICATIONS:

Dogan, F., Taneri, B., & **Erbil, L**. (2018). Use of analogies, metaphors, and similes by students and reviewers at an undergraduate architectural design review. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 1-16.

Erbil, L., & Dogan, F. (2012). Collaboration within design teams participating in Architectural design competitions. *Design and Technology Education: an International Journal, 17*(3), 70-77.

Erbil, L., Kasali, A., Dogan, F. (2022). *Multi-disciplinarity and Collaboration in Computational Design Teams*. Paper is presented at 40th Education and Research in Computer Aided Architectural Design in Europe (eCAADe) Conference, 14 September 2022, Ghent, Belgium

Erbil, L., Kasali, A., Dogan, F. (2021). *Legibility of Design in Collaborative Computational Practices*. Paper is presented as online at XXV Sigradi Designing Possibilities Ubiquitous Conference 2021, Sao Paulo, Brasil.

Erbil, L., Kasali, A., Dogan, F. (2020). *Serendipitous Explorations in Distributed Work in Parametric Design*. Paper is presented as online at Design Computation and Cognition Conference (DCC'20) 2020, Georgia, USA.

Erbil, L., Kasali, A., Dogan, F. (2019). *Computational Design in Distributed Work: Using Digital and non-Digital Tools in Architectural Design Competitions*. Paper presented at the eCAADe + SIGraDi 2019 Conference 2019, Porto, Portugal.