VISUAL STRUCTURES FOR GENERATIVE DESIGN SEARCH SPACES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜNSU MERİN ABBAS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF ARCHITECTURE
IN
ARCHITECTURE

MAY 2014

Approval of the thesis:

VISUAL STRUCTURES FOR GENERATIVE DESIGN SEARCH SPACES

submitted by **GÜNSU MERİN ABBAS** in partial fulfilment of the requirements for the degree of **Master of Architecture in Architecture Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Güven Arif Sargın _____________________
Head of Department, **Architecture**

Assist. Prof. Dr. İpek Gürsel Dino _____________________
Supervisor, **Department of Architecture, METU**

Prof. Dr. Ahmet Can Baykan _____________________
Co-Supervisor, **Department of Architecture, METU**

Examining Committee Members:

Assoc. Prof. Dr. Arzu Gönenç Sorguç _____________________
Department of Architecture, METU

Assist. Prof. Dr. İpek Gürsel Dino _____________________
Department of Architecture, METU

Prof. Dr. Ahmet Can Baykan _____________________
Department of Architecture, METU

Instr. Dr. Dalsu Özgen _____________________
Department of Industrial Design, METU

Assit. Prof. Dr. Başak Uçar _____________________
Department of Architecture, TEDU

Date: May 30, 2014

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GÜNSU MERİN ABBAS

Signature:

ABSTRACT

VISUAL STRUCTURES FOR GENERATIVE DESIGN SEARCH SPACES

Abbas, Günsu Merin
M.Arch., Department of Architecture
Supervisor: Assist. Prof. Dr. İpek Gürsel Dino
Co-Supervisor: Prof. Dr. Ahmet Can Baykan

May 2014, 99 pages

With the adoption of computational strategies in design, the act of design, the process of problem solving, and the interaction, understanding and the representation of design artefacts has changed. With generative design methods, the understanding of design as artefact gives way to design as process. Generative methods entail multiple design solutions, which enlarge the design search space due to the large number of possible design solutions. Regarding the automated design generation process, the interaction of the designer through the design process has been decreased. Hereby, there appears a need for the amplification of designerly actions during the design process to assist designers in the exploration of complex design search spaces. This thesis investigates the use of generative methods in design, and proposes visual structures as a visual complexity management tool for complex design search spaces. Such visual structures have the potential to amplify the interaction between the design search spaces and the designers.

Keywords: generative design systems, design search spaces, visual structures, genetic algorithms, design variations

ÖZ

ÜRETKEN TASARIM ÇÖZÜM ALANLARINDA GÖRSEL YAPILANDIRMA

Abbas, Günsu Merin
Yüksek Lisans, Mimarlık Bölümü
Tez Yöneticisi: Assist. Prof. Dr. İpek Gürsel Dino
Yardımcı Tez Yöneticisi: Prof. Dr. Ahmet Can Baykan
Mayıs 2014, 99 sayfa

Tasarımda hesaplamalı tasarım yöntemlerinin uygulanmasıyla birlikte, tasarım edimi, problem çözümleme süreci tasarımcının süreç ve tasarım nesnesiyle olan ilişkisi, ona dair algısı ve onu temsil etme yöntemleri değişmektedir. Üretken tasarım stratejileri ile tasarım nesnesi tasarım sürecine dönüşmektedir. Üretken tasarım metodları çoklu tasarım çözümlerine yol açmakta ve bu dönüşüm tasarım çözüm alanını çözüm olasılıklarını artırarak genişletmiştir. Hesaplamalı ve üretken tasarım üretme yöntemleri dahilinde otomatikleştirilmiş tasarım üretme süreci göz önüne alındığında, tasarım süreciyle tasarımcı ilişkisi ve etkileşiminin azalmıştır. Böylelikle otomatikleştirilmiş tasarım üretme sürecinde tasarımcı edimlerinin ve tasarımcının süreç ve çözüm alanıyla görsel bağlantısının artırılması gerekmektedir. Bu aşamada, bu tez üretken tasarım sistemlerini incelerken, tasarımcının karmaşık çözüm alanlarını görsel olarak değerlendirmesi, çözüm alanlarının yapılandırması ve süreçle iletişimini artırmak üzere, tasarımcı odaklı bir perspektifte görsel strüktürleri yöntem olarak önermektedir.

Anahtar Sözcükler: üretken tasarım sistemleri, tasarım çözüm alanı, görsel strüktürler, genetik algoritmalar, tasarım alternatifleri

Death is a state of morphosis.

To the memory of my grandparents

Fazile Arslan, Suna Cıngıllıoğlu and Güngör Cıngıllıoğlu

ACKNOWLEDGEMENTS

First and the foremost, I would like to express my sincere gratitude to my thesis supervisor Assist. Prof. Dr. İpek Gürsel Dino for her patience, understanding, support and motivation; and my co-supervisor Prof. Dr. Ahmet Can Baykan for his useful comments, remarks and questions that compels me to think further. This thesis would not have been possible without their guidance.

I would like to present my appreciation to my jury committee, Assoc. Prof. Dr. Arzu Gönenç Sorguç, Asst. Prof. Dr. Başak Uçar and Dr. Dalsu Özgen, for their comments and discussions throughout this research process.

I would like to present my deepest gratitude to my parents Renan Cıngıllıoğlu Abbas and Bedir Abbas for their emotional support, compassion, love and guidance that I feel in every step of my life and my family members Siren Cıngıllıoğlu Kahraman and Gönül Cıngıllıoğlu for their motivation and support. Without their limitless understanding and effort, nothing would have been possible.

Last but not least, my sincere thanks to my friends who motivated and supported me through this research process. Without fun, nothing would have been worth doing it.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

GA	Genetic Algorithms
DSE	Design Space Exploration
GDS	Generative Design Systems

CHAPTER 1

INTRODUCTION

This research focuses on design space exploration, the visualization of design space in generative design systems and the management of complex design search spaces from a designerly point of view.

1.1. Research Motivation and Problem Definition

Design is an act that has been changing and evolving throughout the decades due to technological developments. Particularly during the last ten years, with the developments in computation technologies, the design practice changed dramatically.[1] Such advances in technology proliferated the digital design practice. Digital design is the exploitation of the flexibility and efficiency of computers as tools for the implementation of computation methods as well as the drafting and visualising the design object.

There are two terms that may illustrate the change in contemporary design practice as; (1) computerization and (2) computation. The utilization of computer as a tool for drafting, modelling and manufacturing (CAD/CAM software) is the computerization of design processes. Here, the design process is end-result oriented and based on the data organization and representation methods.[2] Computerization facilitated the modelling and visualization of design artefact(s) with increased efficiency and speed. Computation, on the other hand, had a more profound effect on the design practice.

[1] Spiller, N. (2008). Digital Architecture Now: A Global Survey of Emerging Talent. London, UK: Thames & Hudson

[2] Menges, A., & Ahlquist, S. (2011). Introduction. A. Menges, & S. Ahlquist, Computational Design Thinking (s. 10-29). Chichester, UK: John Wiley & Sons.

Computation in design is the articulation of a generative logic that designated the codes of design generation mechanism instead of articulating a form.[3] The convenience in implementation of computation methods with developing computer technologies results in the utilization of such methods as design exploration and generation tools. With the proliferation of computational methods in design, design has grown into a process of dynamic exploration, transformation and a continuous search. With this shift in the design act, design thinking has also been transformed into what is called computational design thinking. Computational design thinking requires the multifaceted, interdisciplinary, generative and holistic understanding of a system, which considers design as a whole process.[4]

With computational design thinking, the understanding of a design problem has also radically changed. Traditionally, design problems are considered as ill-defined (or wicked), due to their changeable, ambiguous, multi-objective and subjective nature, the. [5,6] Ill-defined problems are ill-formulated and characterized by conflicting values, confusing information and ramifications. [7] Accordingly, ill-defined problems are qualified as indeterminate problems as they do not have any definitive problem formulation, initial states, a stopping rule or clear goals, problem constraints or conditions and methods. [8,9] Therefore, ill-defined problems have large, unspecified and changing solution domains.[10] However, in computational design, although the problems still are ill-defined, they need to be clearly and unambiguously defined together with the design goal (or objective evaluation criteria) and a design generation

---

[3] Kolarevic, B. (2000). Digital Morphogenesis and Computational Architectures. *Proceedings of the 4th Confernece of Congreso Iberoamericano de Grafica Digital*, (pp. 98-103). Rio da Janeiro.

[4] *Op. Cit.* (Menges & Ahlquist, 2011)

[5] Simon, H. A. (1973). The Structure of Ill-structured Problems. *Artificial Intelligence* (4), 181-201.

[6] Dorst, K. (2003). *Understanding Design.* Amsterdam: BIS Publishers.

[7] Rittel, H., & Webber, M. (1973). Dilemmas in General Theory of Planning. *Policy Sciences, Volume 4* (Issue 2), 155-169.

[8] *ibid.*

[9] Akın, Ö. (2001). Variants in design cognition. C. Eastman, M. Mike, & N. Wendy (eds.), *Design Knowing and Learning: Cognition in Design Education: Cognition in Design Education* (pp. 105-125). Netherlands: Elsevier.

[10] *ibid.*

method. The objective evaluation criteria need to be well-defined and quantifiable (such as performance/ fitness criteria) so to be able to computationally guide the design process. Here, computation can only support design if a description for problem formulation and problem constraints that defined the solution domain are clearly defined.

In approaches that make use of such understanding **the notion of a design problem, the act of design, the design process, the design artefact and the involvement of the designer** have changed. The design artefact becomes dynamic; it can be generated and modified algorithmically. Such design approaches and methods are termed as generative design systems.

Generative design systems enable the simulation, exploration and the generation of complex geometries and evolutionary processes of nature,[11] such as swarm behaviour, growth of plants and cities, and social orders and networks. As complex and evolutionary processes of nature provide a mathematical model and a scientific explanation for the generation of complex structures and interactions, they also provide a model for design generation.[12,13] In generative design, such complex systems are modelled by means of a schema (or a procedure or an algorithm) that encodes and regulates the design generation process.[14,15] By articulating a design process by means of a schema, generative design systems replace the single end product with a set of design alternatives and the idea of a process.[16] The design search

[11] *Op. Cit.* (Kolarevic, 2000)

[12] Hensel, M., Menges, A., & Weinstock, M. (2004, May). Emergence: Morphogenetic Design Strategies. *Architectural Design, 74*(3), 6-9.

[13] DeLanda, M. (2002). Deleuze and the Use of the Genetic Algorithm in Architecture. In Neil Leach (ed.), *Designing for a Digital World.* New York: Wiley.

[14] Soddu, C. (2006). Generative Design. A swimmer in a natural sea frame. *Generative Art International Conference.* Milan.

[15] Kalay, Y. E. (2004). Arhitecture's New Media: Principles, Theories, and Methods of Computer-Aided Design. Cambridge, Massachusetts: The MIT Press.

[16] *ibid.*

space that consists of a set of possible solutions for a specified design problem[17] is diversified and broadened with the generation of high numbers of design instances in an automated fashion. Accordingly, for the act of searching by generating and navigating in a design search space,[18] which is termed as design space exploration, the supporting tools are developed as a computational aid for designers to navigate through the design space with the representation of design states by symbols.[19] As such, design space exploration research aims to amplify design activities by the development of design space organization and representation to incorporate the designer into the computation process.[20]

In non-computational design exploration processes, the design space is formed manually by the designer, and therefore is narrower in comparison with the design spaces of generative design systems.[21] Here, the designer may need to explore and evaluate several design alternatives to find the most suitable design solution and prevent too-early design decisions.[22] Cross (2011) suggests two approaches for problem-solving activity as; (1) depth-first approach and (2) breadth-first approach.[23] Depth-first problem-solving is the identification of a problem and the exploration of its solution up until the solution reaches a final evaluation. If the explored solution

---

[17] Cagan, J., Campbell, M. I., Finger, S., & Tomiyama, T. (2005, September). A Framework for Computational Design Synthesis: Model and Applications. *Journal of Computing and Information Science in Engineering* (Vol.5), 171-181.

[18] Gero, J. S. (1993). Towards a Model of Exploration in Computer-Aided Design. J. S. Gero, & F. Sudweeks (eds.) in *Formal Design Methods for CAD (pre-prints)* (pp. 271-291). IFIP, University of Sydney.

[19] Woodbury, R., Datta, S., & Burrow, A. (2000). Erasure in Design space Exploration. (J. S. Gero, eds.) *Artificial Intelligence in Design*, 521-543.

[20] Stouffs, R. (2006). Design spaces: The explicit representation of spaces of alternatives. *AIE EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 20*(2), 61-62.

[21] Cross, N. (2001). Design Cognition: Results From Protocol and Other Empirical Studies of Design Activity. C. Eastman, M. McCracken, & W. Newstetter, *Design Knowing and Learning: Cognition in Design Education* (pp. 79-105). Netherlands: Elsevier.

[22] *ibid.*

[23] Cross, N. (2011). Design Thinking: Understanding How Designers Think and Work. New York: Berg Publishers.

fails, the designer starts again the exploration process from the beginning for the next possible solution. This associates with the behaviour of novice designers.[24] Breadth-first problem-solving, on the other hand, aims to broaden design exploration space towards the sub-solutions. This associates with the behaviour of expert designers.[25] Novice designers concentrate on less number of design variations with trial-and-error, while expert designers evaluate multiple design solutions in parallel to the problem exploration process.[26] Accordingly, depth-first problem-solving behaviour results in narrower solution space by the exploration of less design alternatives; and breadth-first problem-solving behaviour results in larger solution space with the exploration of alternative solutions. Here, premature design decisions associate with the depth-first problem-solving behaviour that results in premature design decisions and requires new trials for design solutions. (Fig. 1). To overcome early design decisions, the designer might need to generate and explore a larger design space and more design alternatives. Generative design systems can broaden design spaces by algorithmically generating many design instances at once.

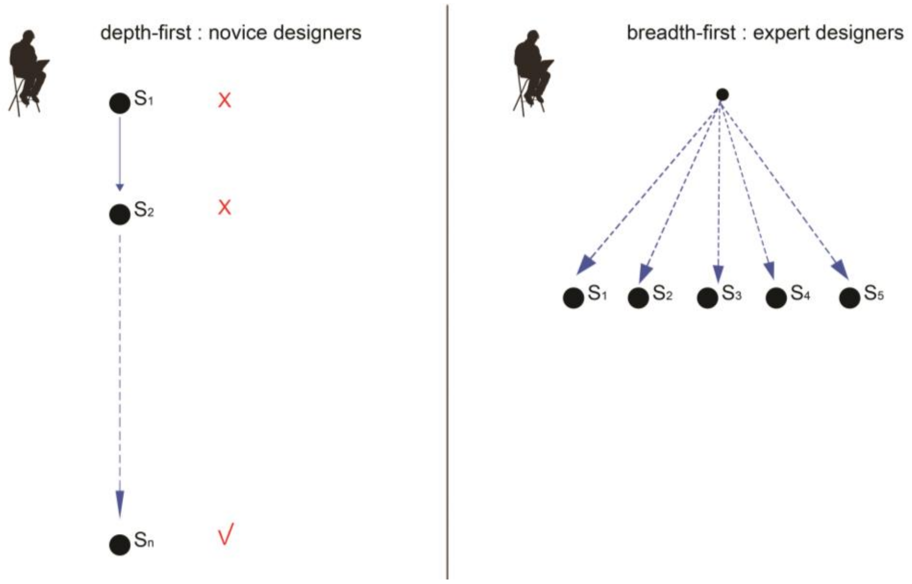

Figure 1 Depth-first and Breadth-first Problem-Solving Behaviours

[24] *ibid.*

[25] *ibid.*

[26] *ibid.*

However, in generative design systems, the high number of design alternatives might pose difficulties in the management and organization of the design search space.[27] Similarly, design space exploration might be challenged.

With generative design systems, the digital tools become more than representation and realization tools; and computation becomes the instrument of the design generation. Due to the complexity of the generative design processes, the representation as a tool for design synthesis and designer's dialogue with the process is no more in use for designers. However, design is a reflective practice that requires a reflective conversation with the designer and the states/ conditions of a design/ design problem.[28] Design, therefore, requires the sensory perception/input (by drawing, modelling, sketching, etc.) to construct and re-construct the design problem, design solution and the conditions of a design space.[29] Traditionally, visual representations play a crucial role during the design process which the designer is in a dialogue with a design problem. The designer uses representations as a problem-solving tool[30] and visual representations mediate between the designer's cognitive activities and the design artefact.[31] Visual representations such as drawings, sketches and physical/ digital models, according to Akin[32], are prominent during the design activity due to their direct correspondence with reality, and their accuracy based on design form. As Visser states; "[t]he possibilities provided by sketches and other types of drawings compared to those offered by purely alphanumeric representations, for example, with respect to the ease of visualisation and manipulation and their corollaries may facilitate

---

[27] *Op. Cit.* (Cross, 2001)

[28] Schön, D. A. (1983). The Reflective Practitioner: How Professionals Think in Action. New York: Basic Books Inc.

[29] Schön, D. A. (1992). Designing as Reflective Conversation with the Materials of a Design Situation. *Research in Engineering Design, Vol. 3*(Issue 3), 131-147.

[30] Visser, W. (2009). Design: One, But in Different Forms. *Design Studies, 30*(3), 187-223., Elsevier Publishing

[31] Oxman, R. (1997, October). Design by Re-representation: a Model of Visual Reasoning in Design. *Design Studies, Vol.18* (Issue 4), 329-347.

[32] *Op. Cit.* (Akın, 2001)

simulation and other forms of evaluation of what are going to become physical artefacts."[33]

Visual representations have a decisive role in the construction of the design search spaces, since design search spaces are formed by the representations of design instances. As a consequence, *visual representations* of a design space are a crucial part of design activity.

A particular example for such case from generative design systems is genetic algorithms which is a search for the optimum solution. During the design generation processes of genetic algorithms, it is the abstract genetic representations (algorithms, genomes, rules, constraints, genetic operators etc.) that steers the design synthesis process rather than visual representations. The genetic representation of design alternatives cannot form a visual dialogue with the design artefact. Here, the role of visual design representations is diminished, eliminating the involvement of the designer from the generation process. Furthermore, in GA processes, up until the generation of a design artefact, the designer is not in contact with the synthesis process. Therefore the designer's visual reflective dialogue with the design process and design artefact(s) is disrupted.

In genetic algorithms, with the automated design generation, visualization is not in use and the designer's visual involvement in decisions is eliminated from the generation phase. However, at the end of the design synthesis, the designer may need the display of the complete design space that is the visual representations of design individuals because the obtained solution may not be the most suitable solution in reference to the subjective criteria of a designer; such as aesthetical and formal anticipations. Within a set of design solutions there may be a design instance that satisfies both designer's subjective criteria and objective evaluation criteria. In other words, the subjectivity of the designer should be equally valued as the objective evaluation of the generative mechanism.

[33] *Op.Cit.* (Visser, 2009)

This thesis is motivated by the challenges regarding design exploration, design search space representation and complexity in design search spaces. Accordingly, this thesis tackles two problems that characterize the generative design synthesis of genetic algorithms: (1) designer-process interaction/ involvement of a designerly evaluation and (2) the management of design alternatives. (Fig. 2) In this case, there arises the need to support the designer's navigation in a design space[34] and help manage the complexity of design spaces that also amplifies the visual dialogue of the designer with the design generation process. As a solution for these problems, several methods are proposed and discussed in following chapters.

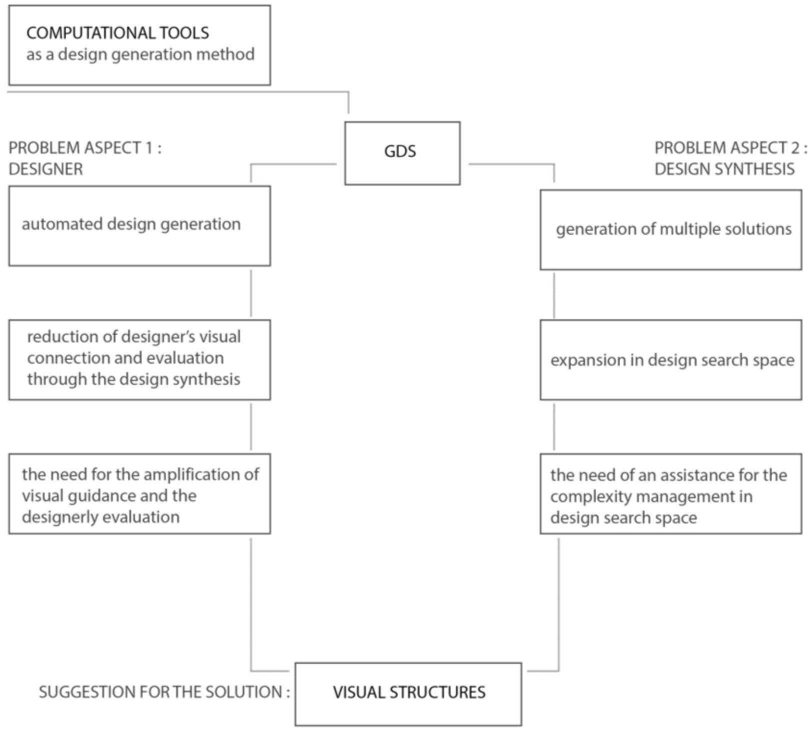

Figure 2 Flowchart

1.2. Research Questions

There are two main research questions that direct this research:

[34] *Op. Cit.* (Stouffs, 2006)

- How can designerly actions be amplified during generative design processes?
- What are the different ways of organizing a design search space to manage multiple design solutions?

1.3. Research Approach, Methodology and Outputs

This thesis approaches the design as an interdisciplinary field of research by various disciplines from nature to computer science. The methods and terms of various disciplines make creative design extensive and a rich research area which are favourable to interdisciplinary approaches. Within this context, the theoretical basis of this research is motivated by the influence of various disciplines such as evolutionary biology, systems science and computational design.

The research methodology is based on three phases: (1) a critical overview of the literature on generative design systems, genetic algorithms and the concepts of emergence and population thinking, (2) the construction of a framework for design search space structuring, and (3) several visual structuring approaches that are suggested for generative design systems.

1.4. Chapter Outlines

Chapter II: Generative Design Systems

Chapter II is a literature review on theoretical background on generative design systems. Several concepts that characterize generative design systems are explained, genetic algorithms is introduced as a generative design system.

Chapter III: Visuality and Generative Design Systems

Chapter III discusses the role of visuality in GDS, and how it has changed as the understanding of the act of design and the role of the designer.

Chapter IV: Structuring the Design Search Space

Chapter IV presents, exemplifies and discusses the role of structuring the design search space in design processes based on genetic algorithms as a method to manage large solution spaces. In this chapter three methods for visual structuring of the design search spaces (perception-based, retrieval-based, optimality-based visual structures) are discussed.

Chapter V: Conclusion

Chapter V is the conclusion of this thesis which summarises the research process and findings and suggestions for the future study.

There are three phases that this thesis follows as; (Fig. 3)

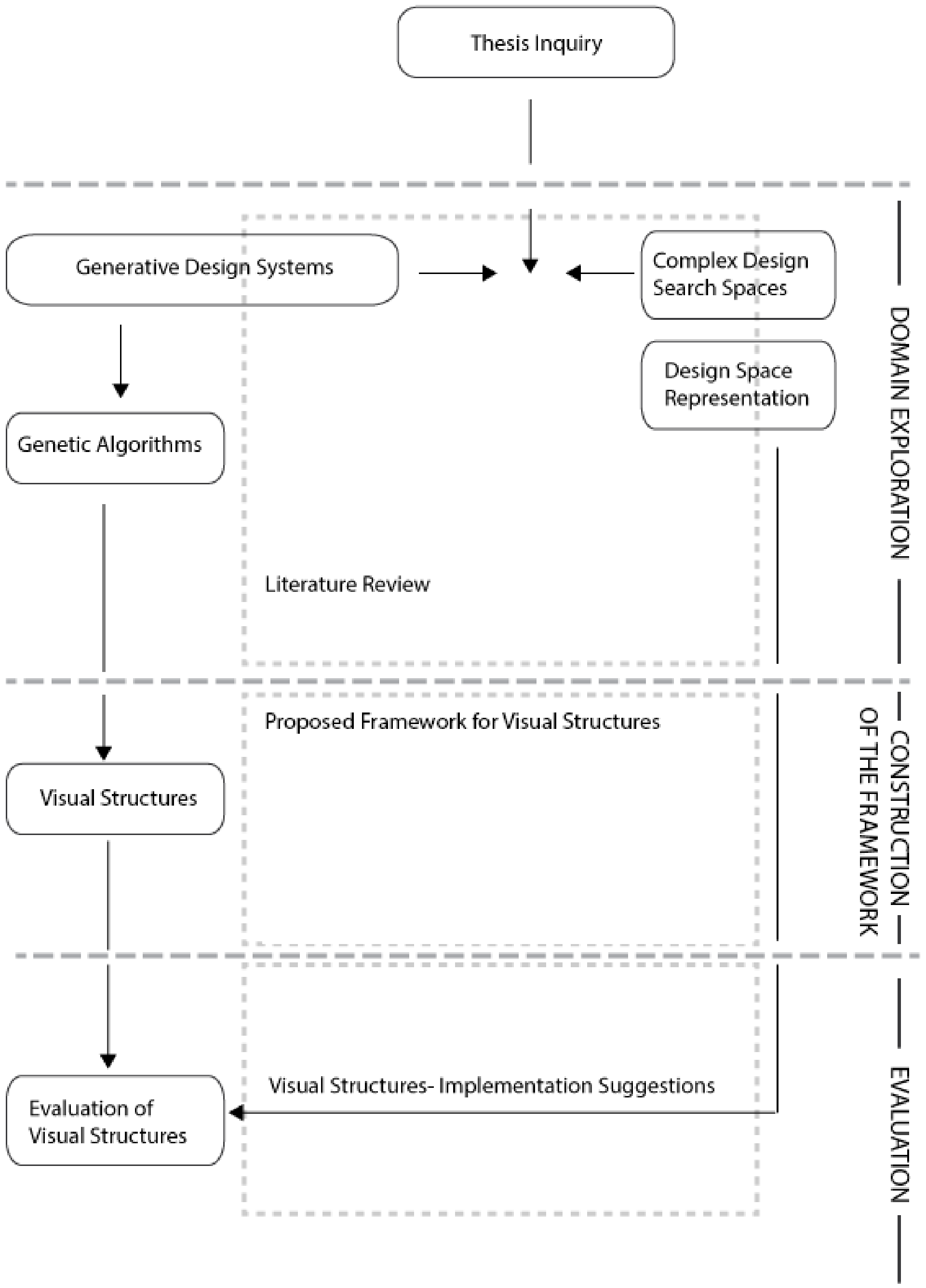

Figure 3 **The Structure of the Thesis**

CHAPTER 2

GENERATIVE DESIGN SYSTEMS

A generative system is one in which the interaction of the rules, and nothing else, will create the thing.

Christopher Alexander
Notes on the Synthesis of Form, 1964

2.1. Defining Generative Design Systems

Generative design systems are systems that employ computational methods as specifications (algorithms or rules) that encode the production of the design artefact.
These systems can be considered as design generation methods that explicitly facilitate design exploration and entail divergence in design search space by generating multiple design alternatives. [35,36, 37] Soddu (2008) asserts that the generative design systems imitate nature to design and simulate the codes and ends with multiple and un-repeatable design variations.[38] According to Shea (2005), generative design systems foster novel and efficient design processes that extend the designer's capabilities through the exploitation of current computation and manufacturing technologies.[39]

---

[35] *Op.Cit.* (Soddu, 2006)

[36] G. Dino, İ. (2012). Creative Design Exploration By Parametric Generative Systems in Architecture. *Middle East Technical University Journal of the Faculty of Architecture* , 204-224.

[37] Krish, S. (2011). A Practical Generative Design method. *Computer-Aided Design*(43), 88–100

[38] *Op. Cit.* (Soddu, 2006)

[39] Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards Integrated Performance-Driven Generative Design Tools. *Automation in Construction*(14), 253-264.

Kalay (2004) defines generative design systems as systems that designate the generating mechanism and the process that generates the design artefact.[40] (Fig.4)

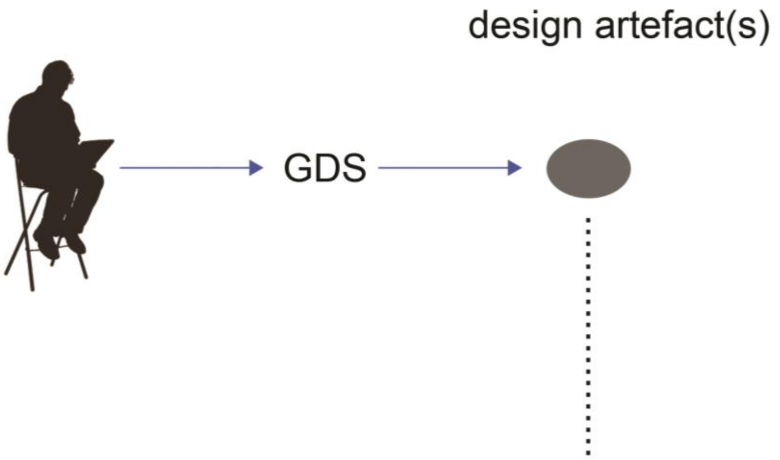

Figure 4 **Generative Design Process**

Generative design systems use algorithms/ schemas/ procedures for design generation.[41] With algorithms, GDS designate the generation logic of a design mechanism.[42] With the designation of the generation logic, rather than articulating a design artefact, the design generation process is designed.[43] The emphasis on the process makes GDS as production mechanisms[44] that enlarged the design search space and enable the divergence in a solution set by its dynamic transformational process that generates variations of design solution.[45] The design artefact in non-computational design processes, which is a singular end product, becomes multiple design alternatives in GDS.[46] As well as the change in the concept of the design artefact, the act of design has changed to the act of generating possible design

---

[40] *Op.Cit.* (Kalay, 2004).

[41] *Op. Cit.* (Dino, 2012)

[42] *Op.Cit.* (Kalay, 2004).

[43] *ibid.*

[44] *ibid.*

[45] *Op.Cit.* (Kolarevic, 2000).

[46] Leach, N. (2009). Digital Morphogenesis. *Architectural Design*, *79*(1), 32-37.

solutions.[47] With GDS, design activity shifted from a form-making to a form-finding process. [48] Form-making is a method of static and traditional form generation that is dominated by the formal decisions of the designer. [49, 50] On the other hand form-finding is a process of computational form exploration that seeks for an appropriate form throughout the design generation process.[51] In form-making, the design decisions predicates the decisions of the designer, while form-finding is a search for the form. [52,53] Accordingly, the concept of form shifted to the concept of 'formation' which means that the form is a result of continuous transformation.[54]

Generative design synthesis processes draw a framework which follow and repeat four operations as; (1) representation, (2) generation, (3) evaluation and (4) guidance (or feedback[55]).[56] These phases have input-output relation; for instance the output of the generation is the input of the evaluation.[57]

According to Cagan et. al. (2005); (1) **representation** is the cognitive modelling of a design problem and "…helps determine the appropriate generation or search mechanism…", (2) **generation** is the "[c]reation of the parts and the whole…", (3) **evaluation** is the "[a]nalysis of how well it meets the design goals and constraints…"

---

[47] Woodburry, R. F. (1990). Searching for Designs : Paradigm and Progress. *School of Architecture*, s. Paper 62.

[48] *Op.Cit.*(Leach, 2009).

[49] *ibid.*

[50] *Op.Cit.* (Kolarevic, 2000).

[51] Coenders, J., & Bosia, D. (2006). Computational Tools for Design and Engineering of Complex Geometrical Structures. K. Oosterhuis, & L. Feireiss (eds.) in, *Game Set And Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies* (pp. 271-278). Rotterdam: Episode Publishers.

[52] *Op.Cit.* (Kolarevic, 2000).

[53] Carpo, M. (2013). The Ebb and Flow of Digital Innovation: From Form Making to Form Finding-and Beyond. *Architectural Design Special Issue: The Innovation Imparative: Architectures of Vitality*, 56-61.

[54] *Op.Cit.*(Leach, 2009).

[55] Mitchell, M. (1996). Genetic Algorithms: An Overview. *Complexity, 1*(1), 31-39.

[56] *Op.Cit*. (Cagan, Campbell, Finger & Tomiyama, 2005)

[57] *ibid.*

and (4) **guidance** is the "[f]eedback on improvements to the design for the next iteration." In case of optimization-based generative procedures, guidance is considered as a part which is inseparable from the generation phase.[58]

Through these steps, the designer gets involved at the beginning of the representation phase by initializing the process; by defining a problem, design procedures, design constraints and boundary conditions. [59]

2.2. Methods of Generative Design Systems:

2.2.1. Agent-Based Systems

Agent-based systems model dynamic local behaviours of individuals and their interactions with each other and their environment. For instance, the cellular automata method operates on cellular structures, where each cell is a discrete state.[60] The process starts with a seed as a determination of a single cell; each cell has a value determined by its neighbouring cells. The iteration of the same rules results in complex patterns of self-similarity.[61,62] (Fig. 5). Swarm intelligence, as a model of non-centralized social agent behaviour inspired by collective behaviour in nature, simulates the interactions between agents and interaction with their environments.[63] In nature, there is an interaction between the agents. For instance, ants communicate with each other through pheromone secretion.[64] Similarly, swarms operate on simple rules that guide

---

[58] *ibid.*

[59] *ibid.*

[60] Wolfram, S. (1994). Cellular Automata and Complexity: Collected Papers. (pp. 211) Westview Press.

[61] Gutowitz, H. (1990). *Cellular Automata: Theory and Experiment.* Amsterdam, Netherlands: Elsevier Science Publishers.

[62] *Op. Cit.* (Wolfram, 1994) (pp. 12)

[63] Kennedy, J., & Eberhart, R. C. (2001). *Swarm Intelligence.* San Diego, CA: Academic Press.

[64] Fleischer, M. (2003). Foundations of Swarm Intelligence: From Principles to Practice. *SWARMING: NETWORK ENABLED C4ISR.* McLean, Virginia.

the interaction between agents based on separation, cohesion and alignment behaviours of agents. [65,66, 67]

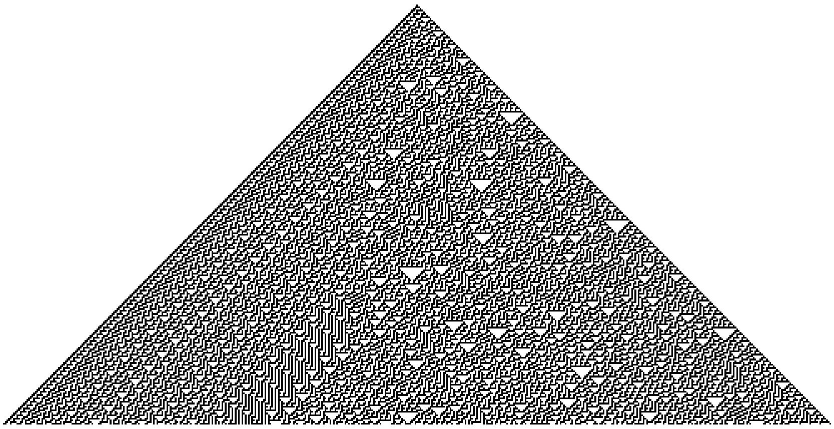

Figure 5 Pattern that is generated by Cellular Automata (Retrieved from: http://upload.wikimedia.org/wikipedia/commons/9/9d/CA_rule30s.png, 16.12.2013.)

Agent-based systems are sensitive and responsive to their environments. Both cellular automata and swarm intelligence are adaptable and self-organizing systems that regulate their behaviour according to the local and environmental conditions.[68] Such simple rules result in complex and holistic behaviour.[69]

In design, cellular automata can be used for those design problems that can formalize its design units as cells. For example; city growth can be modelled by representing urban units as cells; high-density building forms such as social housing by representing each living unit as cells. [70, 71] (Fig. 6, 7.) Swarm intelligence is typically practiced in

---

[65] *Op. Cit.* (Kennedy & Eberhart, 2001)

[66] *Op.Cit.* (Fleischer, 2003)

[67] Coates, P., & Carranza, P. M. (2000). The Use of Swarm Intelligence to Generate Architectural. *In Proceedings of the 3rd Generative Art Conference Generative Art.* Milan, Italy: AleaDesign Publisher.

[68] Wolfram, S. (1984). Cellular Automata as Models of Complexity. *Nature, 311*(5985), 419–424.

[69] *Op. Cit.* (Wolfram, 1994)

[70] Batty, M., Xie, Y., & Sun, Z. (1999). Modeling Urban Dynamics Through GIS-based Cellular Automata. *Computers, Environment and Urban Systems, 23*, 205-233.

[71] Herr, C. M., & Kvan, T. (2005). Using Cellular Automata to Generate High-Density Building Form. B. Martens, & A. Brown , *Computer Aided Architectural Design Futures 2005* (s. 249-258). Springer Netherlands.

search and optimization problems.[72] Human flow paths and interaction patterns can be modelled and simulated for the emergency building evacuation planning with swarm intelligence in architectural design.[73] Furthermore, urban planning strategies can be generated and simulated by swarm intelligence.[74]

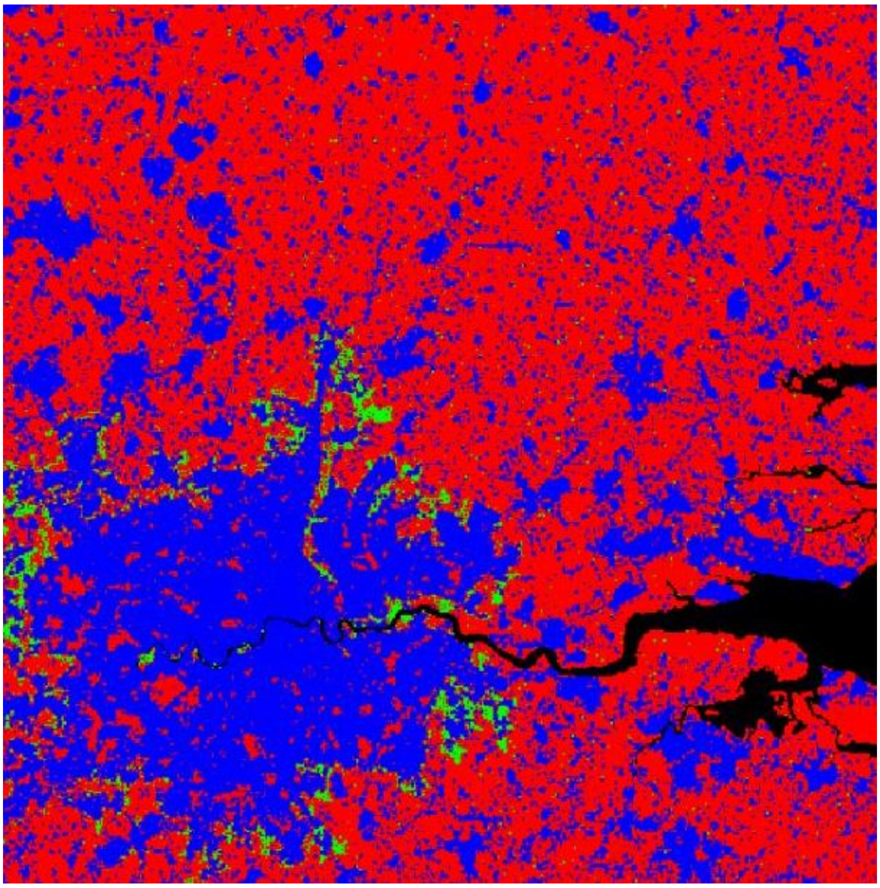

Figure 6 City Growth Simulation by Cellular Automata (Retrieved from: http://www.geocomputation.org/1999/026/gc_026.htm, 20.12.2013.)

[72] Brownlee, J. (2011). Clever Algorithms: Nature-Inspired Programming Recipes. LuLu Press.(pp.229)

[73] Guest, J., Eaglin, T., Kalpathi, S., & William, R. (2013). Visual Analysis of Situationally Aware Building Evacuations. *Visualization and Data Analysis Proceedings of SPIE* . Burlingame, California: The Society for Imaging Science and Technology .

[74] Leach, N. (2009). Swarm Urbanism in *Digital Cities*; pp. 56-63; London: Wiley.

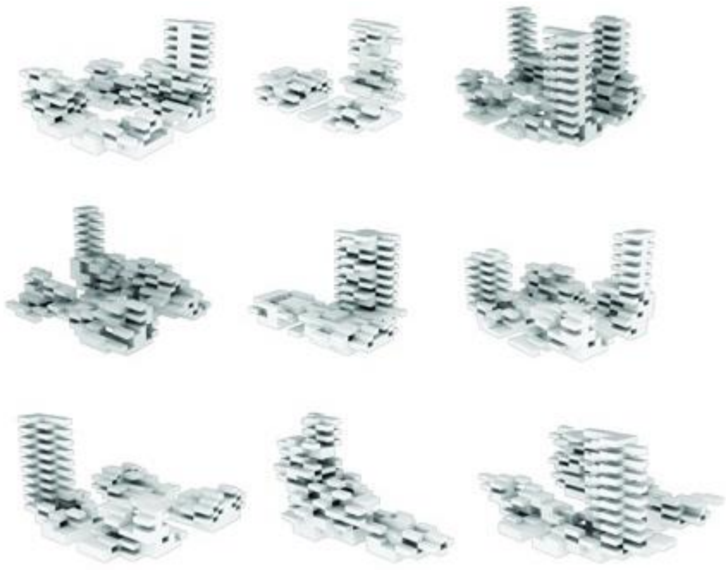

Figure 7 High Density Building Forms (Retrieved from: http:// buablog.wordpress.com/2010/12/21/fargo-comptetition/, 10.03.2014.)

2.2.2. Recursive Growth Systems

Recursive growth systems, such as L-systems and fractals, simulate the growth behaviour in nature.[75] Recursive growth systems operate with production rules that define the generation of new parts from the former ones.[76] The generation mechanism is typically based on recursion.[77] Recursion is a self-referencing process in which a procedure repetitively calls itself, until a stopping condition is satisfied. [78,79] The rule is implemented to all or some parts of a system simultaneously and the recursion process results in self-similarity.[80] Self-similarity is the built-in form of recursion that

---

[75] Flake, G. W. (2000). The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation. Cambridge, MA: MIT Press.

[76] *ibid.*

[77] *ibid.*

[78] *ibid.*

[79] Hunter, D. (2012). *Essentials of Discrete Mathematics.* Sudburry, MA: Jones& Bartlett Learning, LLC.

[80] McCormack, J. (2004). Generative Modelling With Timed L-Systems. J. S. Gero in, *Design Computing and Cognition '04* (s. 157-175). Dordrecht, Netherlands: Kluwer Academic Publishers.

demonstrates the similarity between the parts of the system in multiple scales.[81] There are several recursive growth algorithms like string re-writing, geometric replacement rules and subdivision algorithms. [82, 83] For instance, L-systems use string re-writing that "calls itself" by re-applying the same set of re-writing rules.[84] The re-written string generates the scaled reproduction of the whole system.[85]

Recursive growth systems can be context-sensitive or context-free.[86] Context sensitive recursive growth systems are aware of their neighbouring environment and the recursive algorithm takes into account its environment in the determination of the next state.[87] In case of context sensitive algorithms, the end result (the design) has adaptability to its environment. (Fig. 8)

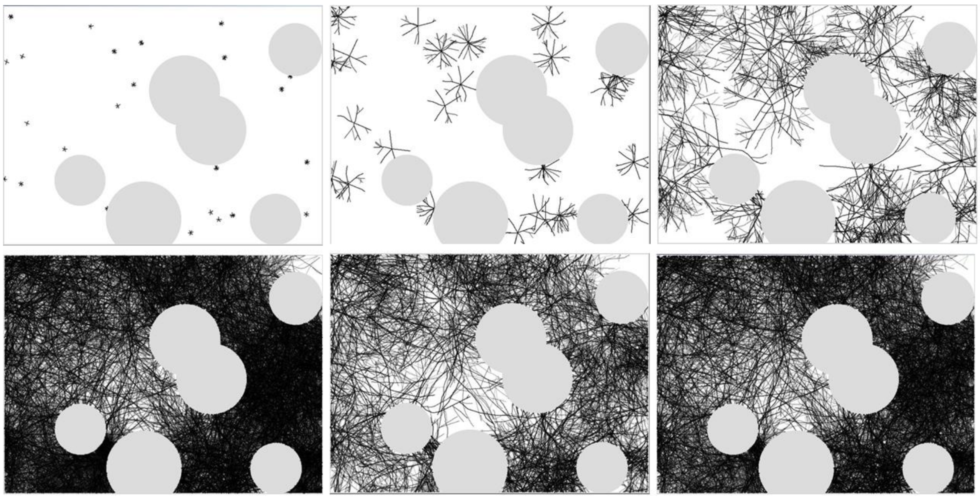

Figure 8 Adaptability to the Environment (Retrieved from: Sakaryalı, Anıl, 2014, Diffusion Growth Algorithms)

---

[81] *Op. Cit.* (Flake, 2000, pp. 61).

[82] Prusinkiewicz, P., & Lindenmayer, A. (1990). *Algorithmic Beauty of Plants.* New York: Springer-Verlag.

[83] Cannon, J. W., Floyd, W. J., & Parry, W. R. (2001). Finite Subdivision Rules. *Conformal Geometry and Dynamics, Vol. 5* , pp.153–196.

[84] Burns, A. M. (2010). Mathscapes- Fractal Scenery. D. Gulick, & J. Scott in, *The Beauty of Fractals: Six Different Views* (pp. 1-21). The Matchematical Association of America (Incorporated).

[85] Shiffman, D. (2013, 12 22). *Chapter-8 Fractals*. The Nature of Code: Retrieved from: http://natureofcode.com/book/chapter-8-fractals/

[86] *Op. Cit.* (McCormack, 2004)

[87] *ibid.*

In design, recursive growth algorithms can be used for those design problems that can formalize its design by growth geometry such as continuous structural systems and patterns. For instance, L-systems are used for the generation of the spatial layouts and for the generation of structural systems in architecture.[88] Based on self-similarity and recursion, fractals are used for the form generation based on orders as symmetry, rhythm and balance for architectural layouts.[89] (Fig. 9, 10.)

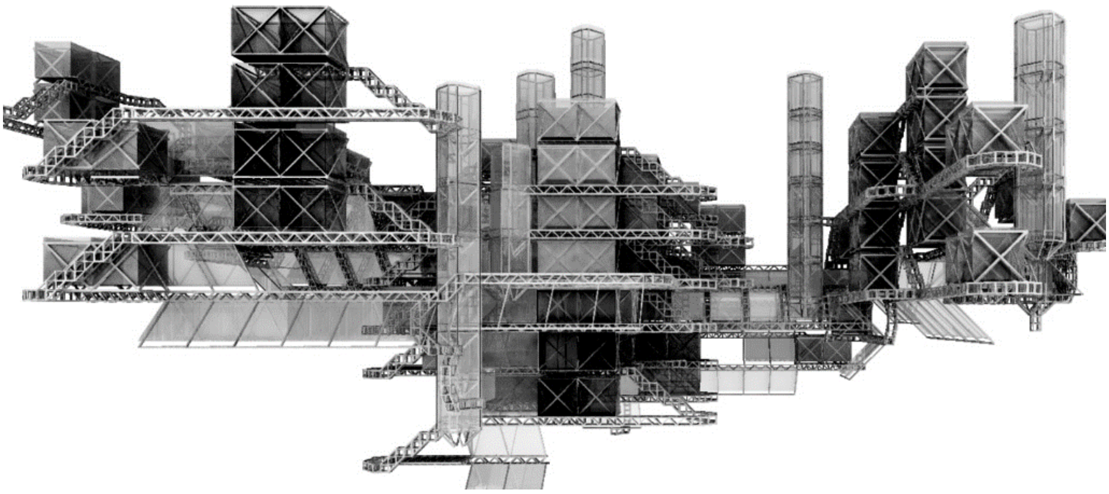

Figure 9 "L-System using three module as leaves" Michael Hansmeyer (Retrieved from:http://www.michaelhansmeyer.com/projects/lsystems.html?screenSize=1&color=0#8, 22.12.2013.)

[88] Hansmeyer, M. (2014, 05 03). *L-Systems*. Computational Architecture: Retrieved from: http://www.michael-hansmeyer.com/projects/l-systems_info.html?screenSize=1&color=1#undefined

[89] Bovill, C. (1996). *Fractal Geometry in Architecture and Design.* Boston: Birkhauser Publishing.

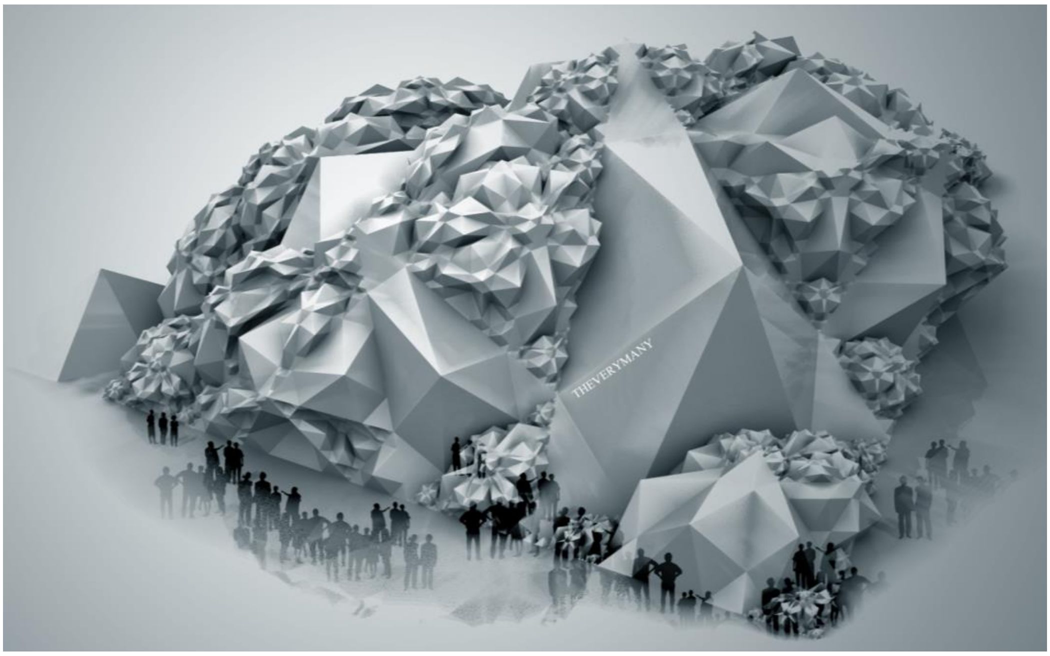

Figure 10 "Droste Effect" Recursive Growth System- An Architectural Implementation (Retrieved from: http://theverymany.com/exploration/08-recursivegrowth/, 10.03.2014.)

2.2.3. Grammar-Based Generative Systems

Grammar-based generative systems such as shape grammars are formalisms that generate variations of form, spatial compositions or a particular language of a set of design artefacts. [90, 91] The generation mechanism is based on a set of shape rules and the initial shape.[92] Shape rules define the transformation procedure and are applied iteratively to the shapes (to the initial shape at the beginning of the generation process).[93] At the end of the synthesis process, a set of design individuals that belong to the same language are generated.[94] (Fig. 11)

---

[90] Knight, T. (1993). Color Grammars: The Representation of Form and Color in Designs. *Leonardo, Vol. 26* (No. 2), 117-124.

[91] Stiny, G., & Gips, J. (1972). Shape Grammars and the Generative Specification of Painting and Sculpture. *Information Processing, 1460-1465. 71*, s. 125-135. Amsterdam: North- Holland.

[92] *ibid.*

[93] *Op. Cit* (Knight, 1993)

[94] *ibid.*

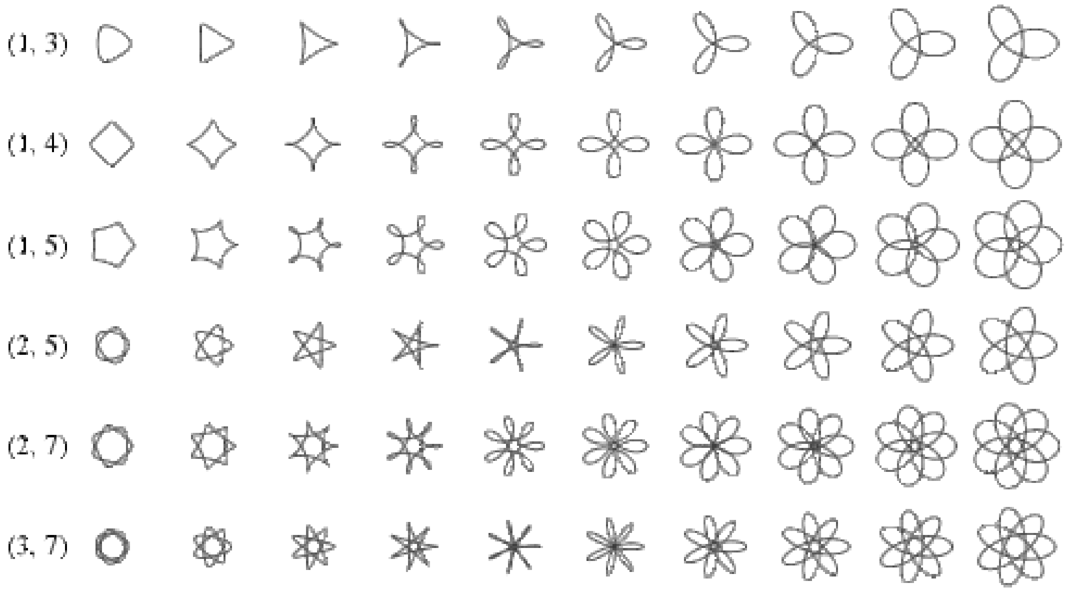

Figure 11 Shapes that are generated by Shape Grammars (Retrieved from:http://drop-ovino.blogspot.com.tr/p/1st-designer-lynne-maclachlan.html, 10.05.2014)

Shape grammars can be grouped as; (1) analytical and, (2) original shape grammars.[95] The analytical shape grammars are used for the analysis of the existing design languages/ styles; and the original shape grammars are used for the generation of new design languages/ styles.[96] In architecture, primarily analytic shape grammars are used to analyse a particular architectural style such as Palladian villas, Queen Anne houses and the buildings of Frank Lloyd Wright.[97] (Fig.12)

---

[95] Knight, T. (1999). *Applications in Architectural Design, and Education and Practice.* Cambridge, MA: Massachusetts Institute of Technology

[96] *ibid.*

[97] *ibid.*

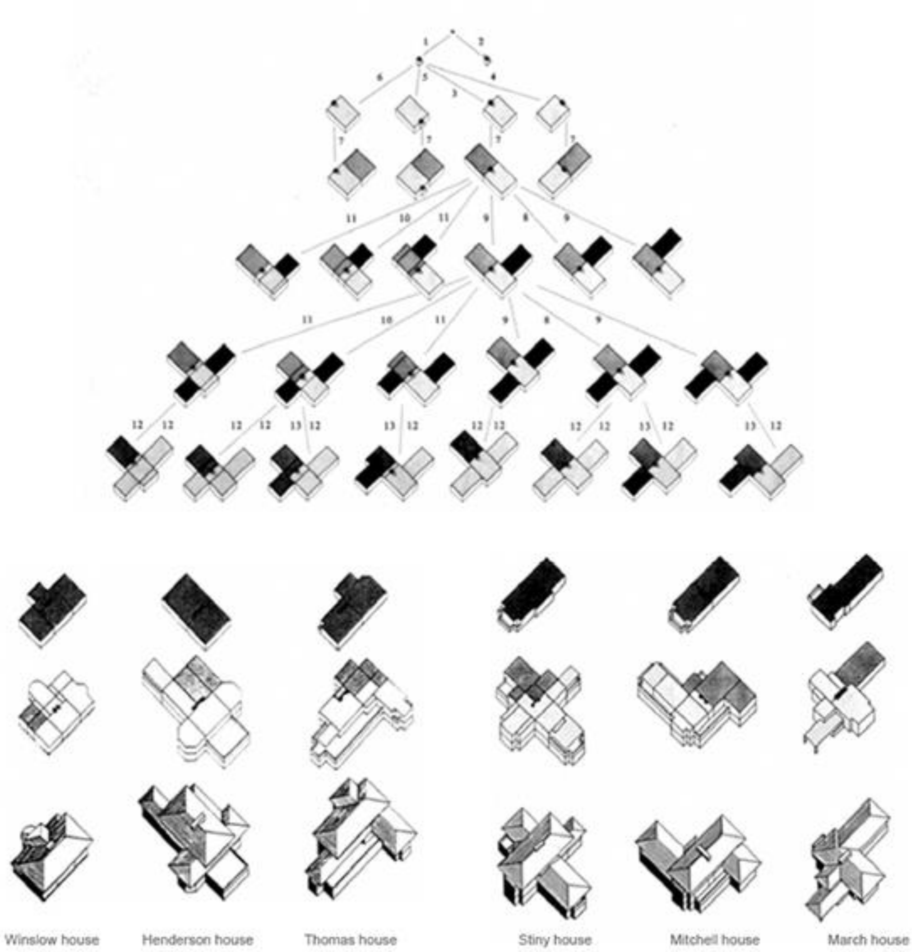

Figure 12 Frank Lloyd Wright Grammar (Retrieved from: http://katakres.wordpress.com/2009/03/05/what-shape-is-this-grammar/, 10.05.2014)

Moreover, the application of original shape grammars are also built on existing grammars which combines the analytic and original grammar approaches at the same time.[98]

2.2.4. Evolutionary Generative Systems

Evolutionary generative systems such as genetic algorithms simulate evolutionary processes; they operate by iterations of mating individual genomes based on trial-error method of problem-solving to achieve better solutions regarding to fitness criteria.[99]

---

[98] *ibid.*

[99] *Op. Cit.* (Brownlee, 2011)

The complexity of living organisms as an emergent result is due to interdependent behaviours of genes.[100] Furthermore, evolutionary algorithms run optimization processes that give way to convergence to the fittest individual of the population.[101] The generation process of evolutionary algorithms is based on operators such as crossover, mutation and selection.[102] After the determination and the representation of genomes, the reproduction mechanism starts by the selection of the genes that are going to be mated.[103] After formation of randomly generated population, the actions are iterated through the generation process.

Genetic Algorithms:

Computational form-generating processes are based on 'genetic engines' that are derived from the mathematical equivalent of the Darwinian model of evolution, and from the biological science of evolutionary development that combines processes of embryological growth and evolutionary development of the species.

Michael Weinstock

Emergent Design Technologies and Design, 2010

Genetic algorithms is an optimization method based on stochastic search mechanisms inspired by natural evolutionary processes.[104] During optimization, there is a search for the optimum solution that is determined by the fitness function, which designates the degree of fitness of the solution.[105] For each generation, the higher score of the fitness is aimed.[106] To achieve a higher fitness score, the fittest individuals are paired off and compete.[107] Regarding the Darwinian concept of natural selection (the term

---

[100] Mitchell, M. (2009). Complexity: A Guided Tour. New York: Oxford University Press. (pp. 276)

[101] *ibid.*

[102] *ibid.*

[103] *Op. Cit.* (Mitchell, 1996)

[104] Frazer, J. (1995). *An Evolutionary Architecture.* London: Architectural Association.

[105] *Op.Cit.* (Kalay, 2004)

[106] Jones, G. (2002, April 15). Genetic and Evolutionary Algorithms. *Encyclopedia of Computational Chemistry*.

[107] *ibid.*

‘survival of the fittest’ coined by Herbert Spencer), lower-scored individuals are eliminated from reproduction, therefore cannot pass on genes to the next generations.[108] The competition and reproduction between the fitters- the local fittest- lead the system to increase the level of fitness, which results in above-average combinations by pairing off above-average counterparts.[109]

In genetic algorithms, there are two genetic concepts that draw the distinction between the schema and the observed characteristics: phenotype and genotype.[110] Genotype is “[t]he genetic composition of an organism” which “[t]he information is contained in the genome”.[111] Phenotype is “[t]he environmentally and genetically determined traits of an organism” which are “[a]ctually observed.”[112] Here genotype can be considered as a schema that contains the characteristics of individuals transmitted throughout generations while phenotype is the observable characteristics of an individual regarding the instructions that are provided by genotype. Here, genotype provides instructions to construct the phenotypes, and phenotypes are the end results of this generation process.[113]

---

[108] *ibid.*

[109] Holland, J. (Winter, 1992). Complex Adaptive Systems. *Daedalus, A New Era in Computation*. MIT Press.

[110] *ibid.*

[111] Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms.* Hoboken, New Jersey: John Wiley & Sons.

[112] *ibid.*

[113] Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic Algortihms. E. K. Burke, & G. Kendall (eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 96-127). New York, NY: Springer Science+ Business Media LLC.

Steps of computational design synthesis in GA: (Fig. 13)

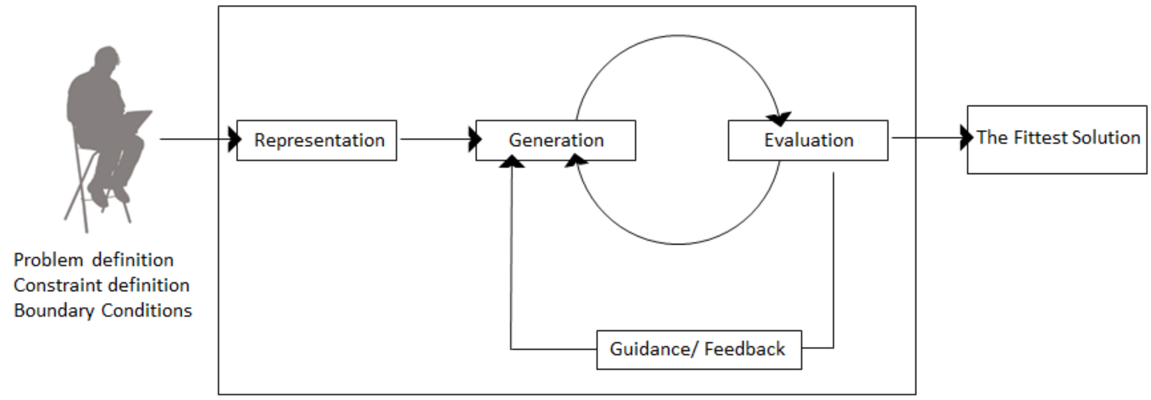

Figure 13 Digital Design Synthesis of GA

Representation:

In genetic algorithms, as well as in other GDS, representation mediates between the design problem and a generative mechanism. Representation in GA is based on the symbolic genome representations.[114] Since symbolic representations are mathematical representations that are conducted via symbols,[115] the representation in genetic algorithms is not visual. (Fig. 14)

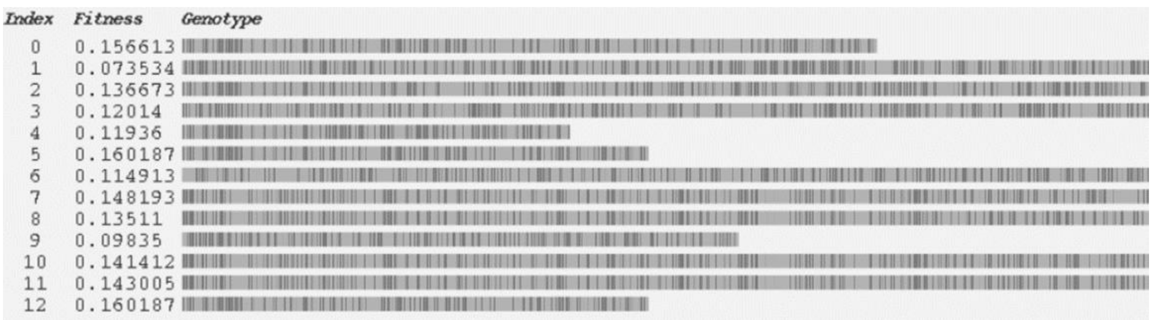

Figure 14 Representation of Genomes (Retrieved from: "Visual Analysis of Evolutionary Algorithms" by Annie S. Wu, Kenneth A. De Jong , Donald S. Burke , John J. Grefenstette , Connie Loggia Ramsey)

[114] *Op. Cit.* (Mitchell, 2009)

[115] *ibid.*

Generation:

After the representation phase, the generation mechanism of GA iteratively generates individuals towards the search for optima.[116] The process first initiates the first generation with the random generation of individuals.[117] After the initial generation is obtained, three genetic operators, selection, crossover and mutation, iteratively produce generations.[118] (Fig. 15)

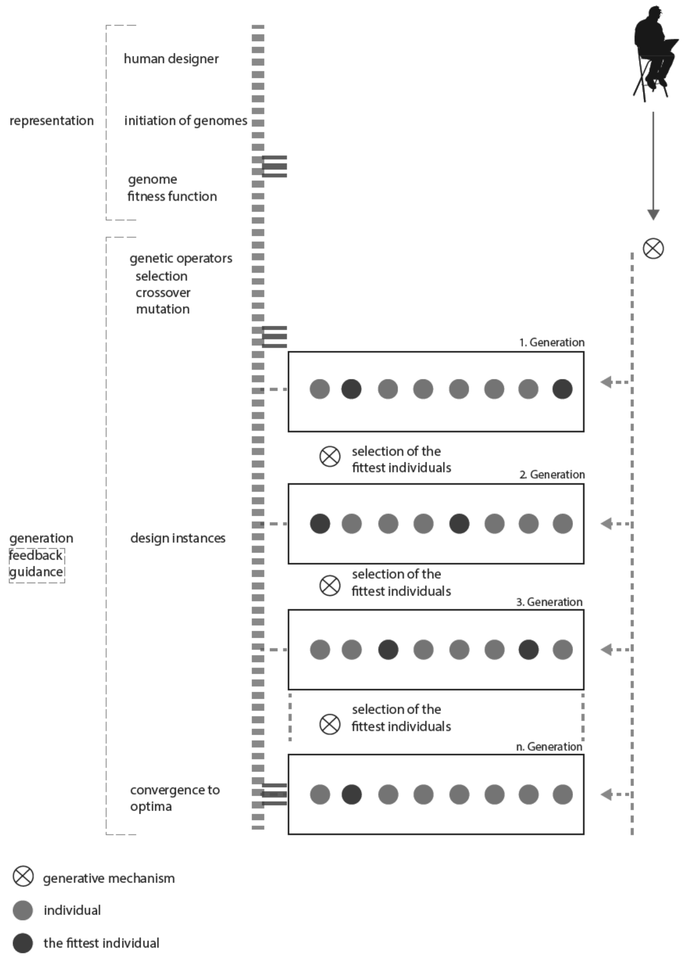

Figure 15 Generation Process of GA

---

[116] Abraham, A., Nedjah, N., & de Macedo Mourelle, L. (2006). Evolutionary Computation: from Genetic Algorithms to Genetic Programming. *Studies in Computational Intelligence (SCI), 13*, 1–20 .

[117] *ibid.*

[118] *ibid.*

Selection is "[t]he process of choosing parent for reproduction"[119] and is based on a fitness function that evaluates the wellness of the solution.[120] The aim of the selection process is to eliminate the weaker individuals such that the stronger genotypes survive and increase the fitness level of the population gradually.[121] For each generation, the surviving individuals (correspondingly the genotypes) are mated to improve the fitness of the population.

Crossover is an operator that combines the parts of genotypes of individuals and generates new individuals.[122] The crossover mechanism is "the random allocation of genes from parents' genotype".[123] (Fig.16)

Mutation is "[r]eproduction operator that randomly alters the values of genes in a parent chromosome." [124] According to Weinstock (2010), "[c]hanges arise in the genome by 'copy errors' and mutations that shuffle the sequence of genes or repeat some segments, which in turn produce changes to the physical form." Mutation increases the diversity in a population by increasing the genotype variation, therefore phenotypic variability.

Evaluation and Guidance:

In GA, during the evaluation phase, the fitness value of each generated genotype is evaluated. Here, with the evaluation of fitness, a feedback mechanism is provided to the generation mechanism such that better individuals contribute to the genetic composition of the next generation. This feedback is also termed as guidance as it informs the process of search to find better iterations for the next generation.[125]

---

[119] *Op. Cit.* (Haupt & Haupt, 2004)

[120] *Op. Cit.* (Abraham, A., Nedjah, N., & de Macedo Mourelle, L., 2006).

[121] Sober, E. (2006). *Conceptual Issues in Evolutionary Biology.* Cambridge, MA: The MIT Press.

[122] *Op Cit.* (Kalay, 2004)

[123] *Op. Cit.* (Hensel, Menges & Weinstock, 2004)

[124] *Op. Cit.* (Haupt & Haupt, 2004)

[125] *Op.Cit*. (Cagan, Campbell, Finger & Tomiyama, 2005)

Therefore the generation phase is guided by evaluation during the synthesis process which leads the system towards better solutions.

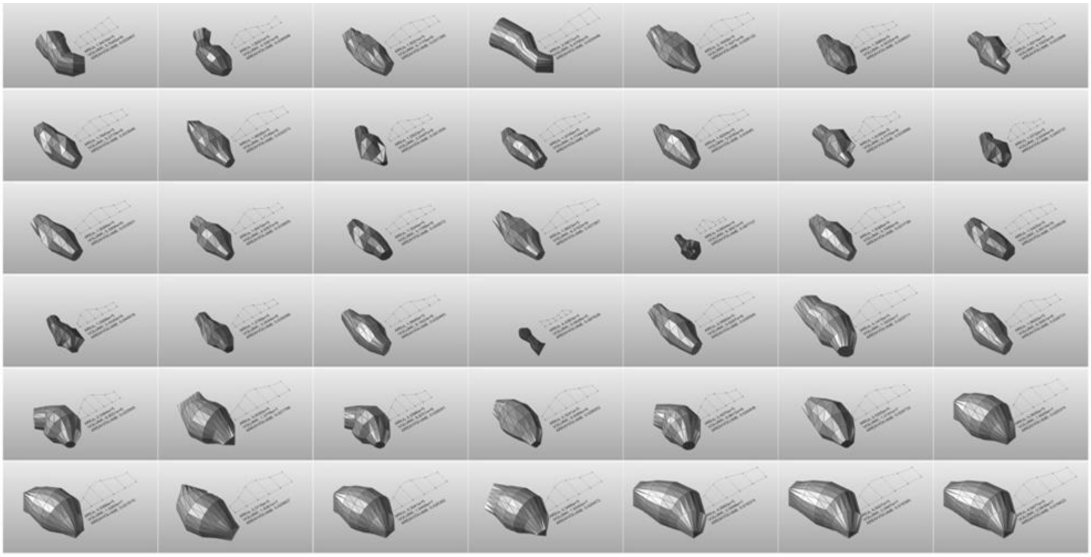

Figure 16 Design Instances Generated by Genetic Algorithms (Retrieved from: http://gracefulspoon.com/blog/tag/evolution/, 12.05.2014)

Due to the process of GA and the designer-process interaction, the genetic algorithms are selectively studied within the scope of this thesis.

2.3. Complexity

Complexity is a characteristic of systems that have many interrelated parts.[126] A part by itself cannot determine the global behaviour of the whole, but the interactions between many parts give rise to complex behaviour or form. [127, 128] In complex systems, the whole is *more than sum of its parts.* The level of complexity increases with the number, interdependency and differentiation of parts.[129]

---

[126] *Op. Cit.* (Mitchell, 2009)

[127] *ibid.*

[128] Holland, J. (1998). *Emergence: From Chaos to Order.* Oxford: Oxford University Press.

[129] Hensel, M., Menges, A., & Weinstock, M. (2010). Emergent Technologies and Design: Towards a Biological Paradigm for Architecture. Oxon: Routledge.

As an example, the social collective behaviour of ants demonstrates complexity. There is a remarkable social organization in ant colonies which is based on inherent class distinctions (queen ants, worker ants and soldier ants).[130] In each colony there is one queen ant as a sole producer of eggs for the next generation. From these eggs only few female ants have wings that indicate them as queen ants for the next generation. The female ants which do not have wings become worker or soldier ants. Here, the social order and organization is maintained by the inherent specifications of ants. Furthermore, each individual behaves locally by following relatively simple rules, such as responding to chemical signals of other ants. However the global behaviour that manifests like the creation of foraging paths followed by ant colony is more and smarter than the behaviour of an individual ant.

Complex Systems:

Complex systems (1) are composed of **large networks** of relatively **simple parts, and** (2) exhibit **complex collective behaviour** that emerges without any top-down controlling mechanism.[131] (3) Complex systems are **self-organizing** systems that have bottom-up organizational principles.[132]

Human neural networks, the immune system, the World Wide Web, social networks (Twitter, Facebook, etc.), cities, climate, and insect colonies are several examples of complex systems.

As complex systems are structured on **bottom-up principles** that are without any **top-down controlling** mechanisms, there is a need to clarify the concepts of bottom-up and top-down. Top-down is a term that defines hierarchical systems that develop from the whole to the parts.[133] As top-down systems are based on the reduction of the system to a single state or a scheme, the characteristics of the whole define and control the

---

[130] Goetsch, W. (1957). *The Ants.* Ann Arbor, MI: University of Michigan Press.

[131] *Op. Cit.* (Mitchell, 2009, pp .13)

[132] *ibid.*

[133] Mitchell, W. J., Liggett, R. S., & Tan, M. (1989). Top-Down Knowledge-Based Design. *CAAD futures Digital Proceedings* (pp. 137-148). Boston: The MIT Press.

characteristics of the parts. [134] On the other hand, bottom-up is a term that defines non-hierarchical systems that develop from the parts to the whole.[135] In bottom-up systems, the specifications and the local interactions of the parts are determined and the behaviour of a system emerges by parts and their interactions.[136]

To illustrate top-down and bottom-up approaches, when city planning schemes of New York and İstanbul are considered; due to the masterplan of New York that was designed as a grid parcelling in 19th century,[137] grid-planning as an urban scheme that corresponds with the high-level structure hierarchy that determines the the parts of lands as low-level hiearchies. On the other hand, İstanbul demonstrates emergent form of urbanization due to the organic development and growth of the city,[138] which the organic formation of parts of lands (low-level hierarchies) constitutes the emergent urbanization (high-level hierarchy). It can be claimed that New York demonstrates top-down and İstanbul demonstrates bottom-up urbanization. (Fig. 17, 18.) As it is seen in the images below, the grid-planning of New York proposes a rigid scheme that defines the parcelling of each land. However, the formation of the land parcels and roads in the plan of İstanbul proposes an emergent and adaptive growth and organization to the city.

---

134 *ibid.*

135 Oxman, R. E., & Oxman, R. M. (1992). Refinement and Adaptation: Two Paradigms of Form Generation in CAAD. *CAAD futures Digital Proceedings 1991* (pp. 313-328). Braunschweig/Wiesbaden: Friedrich Vieweg & Sohn Verlagsgesellschaft mbH.

136 *ibid.*

137 Marcuse, P. (2007). The Grid as City Plan: New York City and Laissez-Faire Planning in the Nineteenth Century. *Planning Perpectives*, 287-310.

138 Çelik, Z. (1986). The Remaking of Istanbul: Portrait of an Ottoman City in the Nineteenth Century. Seattle: University of Washington Press.

Figure 17 Top-Down City Planning of New York (Retrieved from: http://www.npr.org/blogs/krulwich/2012/09/12/160996525/odd-things-happen-when-you-chop-up-cities-and-stack-them-sideways, 13.05.2014)

Figure 18 Bottom-Up (Emergent) City Planning of İstanbul (Retrieved from: http://www.npr.org/blogs/krulwich/2012/09/12/160996525/odd-things-happen-when-you-chop-up-cities-and-stack-them-sideways, 13.05.2014)

Relationship between GDS and Complexity

Generative design systems demonstrate the characteristics of complex systems. When GDS methods are investigated as agent-based systems, recursive growth systems, grammar-based systems and evolutionary systems, all types of GDS demonstrate complex emergent behaviour in various ways.

For instance, in cellular automata the complexity of patterns is a result of the interactions between simple identical parts.[139] Throughout the generation process of cellular automata, the complexity in generating pattern increases with the iterative rule application to the identical parts. (Fig. 19)

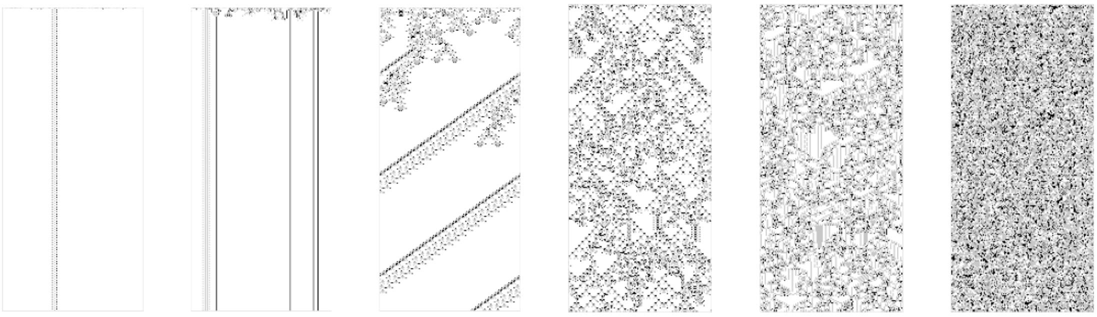

Figure 19 Increase in Complexity of CA Pattern (Retrieved from: http:// theory.org/complexity/cdpt/html/node4.html#foot546, 18.05.2014)

Local parts, Simple Procedures

Both generative design systems and complex systems are based on local parts and emergent complex behaviour. The system parts are the agents in agent-based systems; the cells in cellular automata, and the genes in evolutionary algorithms. Procedures/ rules regulate the interactions of agents in agent-based systems, procedures/ rules determine the reproduction mechanisms and convenience of the solutions in evolutionary algorithms and development logic in recursive growth systems.

Non-linearity, Self-Organization and Emergent Behaviour

Self-organization, emergence (emergent behaviour) and non-linear behaviour emergence are three major characteristics of complex systems as well as generative design systems. [140,141]

Non-linearity is a property that stands for a system that does not have any direct relation between the initial conditions. Both generative design systems and complex

[139] *Op. Cit.* (Wolfram, 1994)

[140] *Op.Cit*. (Holland, 1998)

[141] *Op. Cit.* (Shiffman, 2013)

systems show non-linear behaviour due to the emergent result which starts in simplicity and results in complexity. Non-linear interactions between parts give rise to self-organizations of such systems.

Self-organization is an emergent behaviour that results in emergence of order by the system that organizes itself according to its dynamic internal processes.[142] The cells that structure organs, the behaviour or flocks of birds and schools of fishes are several examples for self-organization of systems in nature.[143] In agent-based systems, in cellular automata, the states of cells determine the states of neighbouring cells, therefore the behaviour of the system and in swarm intelligence the swarm's behaviour is determined by the behaviours of single agents.[144] Here, systems reorganize themselves according to their parts that constitute their internal dynamics.

Another concept that characterizes complex systems is adaptability. Adaptation is the change in the behaviour of a system as a response to its environment. The adaptive system re-structures itself according to the external conditions it encounters. Adaptive behaviour enhances the fittingness/ resilience/survivability of a system to its environment.[145] Such systems are dynamic and far from the equilibrium state by the flow of energy and information based on feedback that enables system to adapt to its environment.[146] All complex systems are not adaptive systems. A complex system is qualified as adaptive if it interacts with its environment and reacts accordingly. Therefore such systems have permeable system boundaries that enable the interaction with the environment. This reaction may involve exchanging material and information, changing its behaviour, and / or reorganizing itself. Small changes in conditions or

---

[142] Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-organization in Biological Systems.* New Jersey: Princeton University Press.

[143] *ibid.*

[144] *Op. Cit*. (Holland, 1992)

[145] *Op. Cit.* (Mitchell, 2009, pp .13)

[146] *ibid.*

interactions, due to the non-linearity in interactions of parts, may result in global effects.[147]

Emergence (emergent behaviour) is a result of non-linearity and self-organization in complex systems.[148,149] It is "[t]he movement from low-level rules to higher-level phenomena."[150] Here, an emergent behaviour can be a part of higher-level complex system or another emergent behaviour.[151] In agent-based systems, the emergent complex result is collective behaviour, while complex structures as a development of a plant in recursive growth systems and the living organisms as phenotypes in evolutionary algorithms.

DeLanda (1997) exemplifies emergence with the formation of sedimentary rocks.[152] There are pebbles with various shapes, size and weight that are transported by water. Each pebble reacts to water differently, these pebbles are segregated through water transportation according to their reaction to water. Such segregation results in the formation of homogenous groups of pebbles at the bottom of the sea. DeLanda terms this process as sedimentation. After the sedimentation, nature is *cementing* (pebbles are cemented by the dissolved substances in the water) these pebbles and a sedimentary rock as a new entity emerges with emergent properties. Such emergent properties as strength or permeability are not observed in pebbles. As a result of sedimentation and cementing processes, a new structure has emerged. Furthermore, DeLanda proposes emergence of sedimentary rocks as a model for emergence of social classes, species and institutional hierarchies.[153]

---

[147] *Op.Cit.* (Mitchell, 2009)

[148] *Op.Cit*. (Holland, 1998)

[149] *Op. Cit.* (Shiffman, 2013)

[150] Johnson, S. (2001). Emergence: The Connected Lives of Ants, Brains, Cities, and Software. New York: Scribner.

[151] *Op.Cit.* (Mitchell, 2009)

[152] DeLanda, M. (1997). *A Thousand Years of Nonlinear History*. New York: Zone Books.

[153] *ibid.*

Emergent behaviour is also defined as a change and reconfiguration of parts as a result of the underlying order of a system. [154, 155] As complex systems are subject to constant change, emergence demonstrates 'changeable form' and iterative and recognizable patterns.[156] As discussed previously, in complex systems, there is an emphasis on local interactions and their global results that lead to emergent behaviour. Each part of a system act according to procedures. The local interactions between parts are determined by the schema; therefore the emergent behaviour is set by the schema.[157] Holland defines algorithms as mechanisms for the explanation of emergent behaviour; the mechanism which is a mediator between the parts of a system.[158] Here, there is an emphasis on schema-based structure of complex systems as such structure behaves like a backbone that constitutes internal structure of a system.[159] The internal structure is termed by Holland as internal model and termed by Gell-Mann as schema.[160] The relationship and dependency between the parts of a system describes its internal structure.[161] For instance, seeds that represent genetic specifications determine the biochemical interactions between parts, or board games that have rules that designate the moves in the game.[162] The seed that follows the specifications of growth transforms into a plant. Emergent behaviour results in states or forms that are unpredictable and new, but each state/ form/ instance exhibits recognizable pattern/ feature/ organization due to the common schema. Here, there is a causality in which

---

[154] Weinstock, M. (2010). *The Architecture of Emergence.* West Sussex: John Wiley & Sons.

[155] Kauffman, S. (1995). At Home in the Universe: The Search for Laws of Self-Organization and Complexity. New York: Oxford University Press.

[156] *Op.Cit.*(Holland, 1998)

[157] Holland, J. (2011). Constrained Generating Procedures. A. Menges, & S. Ahlquist, *Computational Design thinking* (pp. 131-142). Chichester, UK: John Wiley & Sons.

[158] *ibid.*

[159] Manson, S. M. (2001). Simplifying Complexity: A Review of Complexity Theory. *Geoforum*(32), 405-414.

[160] Gell-Mann, M. (1994). Complex Adaptive Systems.(eds. G. A. Cowan, D. Pines, & D. Meltzer), *Complexity: Metaphors, Models, and Reality* (pp. 17-45). Addison-Wesley.

[161] *Op. Cit.* (Manson, 2001)

[162] *Op.Cit.* (Holland, 1998)

the schema guides the complex interactions, these complex interactions demonstrate emergent behaviour; and emergent behaviour demonstrates iterative patterns.[163]

Emergence, both is natural and artificial systems can create complex forms.[164] The emergence of living forms and constructed forms are both the products of the complex processes based on information transmission.[165] In GDS, the design **artefact** as a constructed form is considered as a complex system that interacts with the external conditions.[166] One design instance is only one state of a process that is constructed with generative processes and procedure / algorithms. As the idea/ concept of form shifted to continuous transformation through the generative design synthesis process, each form represents different state of the system. The artefact shifts to procedural design synthesis process; and borrows natural and artificial algorithms to synthesis and evaluation. Here, the process generates multiple design forms by generative procedures. (Fig. 20). Furthermore, by non-linearity of emergence, the unpredictability of the end results from the start conditions lead to an emergent design form.

Some complex systems are visual, such as cellular automata that simulates the divergence and complexity in complex patterns of nature. Genetic algorithms can give way to complex phenomena and emergent results;[167] and through the synthesis process GA generates visual variations and emergence between the design instances. As genetic algorithms are based on evolutionary development in nature, natural processes can be a model for emergence in evolutionary algorithms. GAs are structured on three mechanisms as selection, crossover and mutation. The changes in genomes appear with reproduction operators that change the phenotype of the individual.[168] Crossover

---

[163] *Op.Cit.* (Hensel, Menges, & Weinstock, 2004)

[164] *Op.cit.* (Hensel, Menges & Weinstock, 2010)

[165] *ibid.*

[166] *ibid.*

[167] Terzidis, K. (2011). Algorithmic Form. A. Menges, & S. Ahlquist in, *Computational Design Thinking* (s. 94-102). London: Wiley.

[168] *Op. Cit.* (Weinstock, 2010)

and mutations change the genetic information passing through the genes.[169] As a result, new forms and patterns emerge.[170] Correspondingly the transmission of new genes supports the emergence of new patterns.

Figure 20 Emergence of Multiple Forms through the Design Synthesis Process (Retrieved from: http://emtech.aaschool.ac.uk/emergence-and-design-seminar-2/, 15.05.2014)

Independent State of Emergence:

Emergence is independent by demonstrating non-linearity that obscures the idea of causality of emergence.[171] Here, emergence is a part of the system as a result of system's self-determinacy, yet it is partially independent from the system by being in a state of more than the summation of the parts itself.[172]

From the nature, considering flocks of birds, schools of fish and ant colonies, the collective behaviour of the whole (flock, schools and the colony) is more than its agent (a bird, a fish, an ant). This means each individual organism performs its individual task and as a summation of all individual performance, there emerges an intelligent collective behaviour. Emergent collective behaviour is at an independent state because it emerges by individual performance of its part, yet it cannot be deducible from the performance of its parts. (Fig. 21)

---

[169] *Op.cit.* (Hensel, Menges & Weinstock, 2010)

[170] *ibid.*

[171] Fromm, J. (2005, June 13). *Types and Forms of Emergence.* May 23, 2013 Retrieved from: Cornell University Library: http://arxiv.org/abs/nlin/0506028

[172] *ibid.*

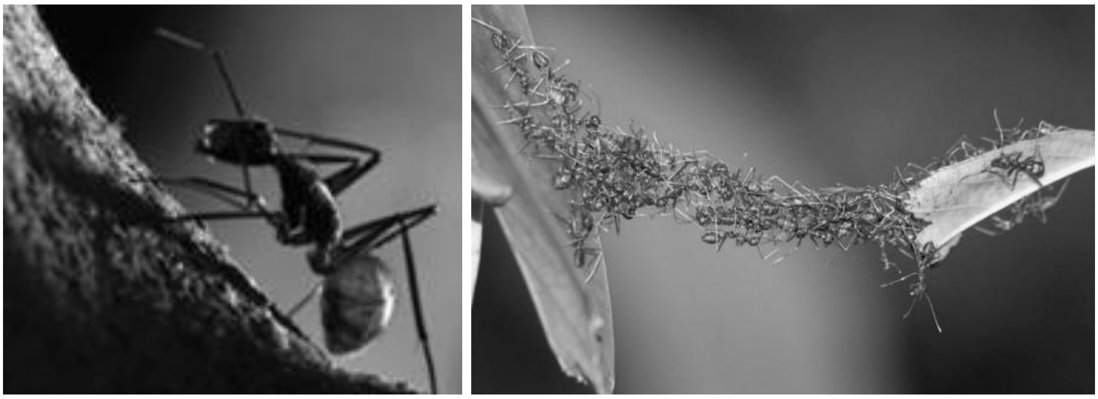

Figure 21 The Agent and the Whole and its Collective Behaviour (Retrieved from: (1) http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications and (2) http://www.nextnature.net/2013/07/what-ant-colony-networks-can-te, 14.03.2014 11:38)

The independent state of emergence, leads the process to reach unpredictable states which are governed by its own procedures and that makes harder to conceptualize a model or a map for the emergence.[173] According to that, each emergent case must be considered as unique to be explored; like emergence of plants from seeds, emergence with weathering, erosion and deposition that is created by the climate change that changes the behaviour of the process that reveals the emergence of the new landforms.[174]

2.4. Population Thinking and Design Variations

For the typologists, the type (eidos) is real and the variation an illusion, while for the populationist the type (average) is an abstraction and only the variation is real.

Ernst Mayr

Typological versus Population Thinking, 1976

---

[173] Dodder, R., & Dare, R. (2000). Complex Adaptive Systems and Complexity Theory: Inter-related Knowledge Domains. Cambride, MA: MIT.

[174] *Op. Cit.* (Weinstock, 2010)

Population thinking is a concept in evolutionary biology, which proposes the uniqueness of each individual as a result of inbreeding that forms the population.[175]

To understand the position of population thinking in design, it is necessary to understand the term in biology. Population thinking is a term presented by Ernst Mayr based on Darwin's early theories on evolution and natural selection.[176] In biology there are two opposing approaches to the genesis of forms as; (1) typological thinking associated with traditional biology and (2) population thinking associated with modern biology.[177] Population thinking opposes to typological thinking by accentuating the singularity of each individual in a population. [178,179] For typological thinking, the type is real and the variations which are dependent on type are illusions.[180] On the other hand, population thinking defines type as an average abstraction of the common features of a population, and considers variations as real.[181] With Darwin's *Origin of Species* published in 1859, the ideas of typological thinking are replaced by population thinking.[182] Typological thinking takes its roots from Plato's concept of *eidos.*[183] *Eidos* signifies the unalterable fixed ideas, which are real and contrasting with the illusoriness of the shadows reflecting onto Plato's cave, which are variable.[184] Typological thinking opposes to evolution, which is mainly based on the gradual alterations on species.[185] Unlike typological thinking, population thinking supports the

---

[175] Mayr, E. ((1976) 1994). Typological versus Population Thinking. E. Sober, *Conceptual Issues in Evolutionary Biology: An Anthology* (pp. 156-160). Cambridge, MA: The MIT Press.

[176] Trummer, P. (2011). Associative Design: From Type to Population. A. Menges, & S. Ahlquist (eds.), *Computational Design Thinking* (pp. 179-197). Chichester, UK: John Wiley & Sons.

[177] Sober, E. (1980, September). Evolution, Population Thinking, and Essentialism. *Philosophy of Science,, Vol. 47*(No. 3), 350-383.

[178] *Op.Cit.* (Mayr, E. (1976) 1994).

[179] *Op.Cit.* (Trummer, 2011).

[180] *ibid.*

[181] *ibid.*

[182] *Op. Cit.* (Mayr, (1976) 1994)

[183] *ibid.*

[184] *ibid.*

[185] *Op. Cit.* (Sober, 1980)

evolution and singularity of the individual.[186] For populationists, the type is an abstraction, which bears the average characteristics of a species that consolidates the singularity of each organism that is composed of unique features can be collectively described by generic characteristics based on shared features.[187] Here, shared features constitute generic characteristics of a type which corresponds to repeating patterns through design instances that are generated by the same schema.

Entity and Instance:

Within the scope of population thinking, there are two terms, which are used in data-modelling that associate with type and singularity of the individual; (1) entity and (2) instance. Entity signifies a group of something that shares the attributes and behaviour.[188] Instance is a specific realization of any object that emphasizes the distinct identity of the object.[189] As stated "[a]ttributes describe an entity's characteristics. All entity instances of a given entity class have the same attributes, but vary in the values of those attributes."[190] According to these definitions, the concept of an entity corresponds to the type, and the instance corresponds to an individual in population thinking.

In genetic algorithms, with schema which is the genotype, the generic features which refers to the type/ the entity is determined. As well as generic characteristics of the type/ entity, phenotypes with different attribute values correspond to new design instances. In genetics, genes control the growth of the individual by making changes in the body plan.[191] Here, each individual is a variation of the schema. The schema remains the same but the values that are assigned to the parameters generate the design

---

[186] *Op.Cit.* (Mayr, (1976) 1994)

[187] *ibid.*

[188]Oracle Corporation. (2013, 11 26). *Oracle-Define an entity*. Retrieved from: Oracle:http://docs.oracle.com/html/E24270_01/Content/Data%20model/Define_an_entity.htm

[189] *ibid.*

[190] Kroenke, D. M., & Auer, D. (2009). *Database Processing: Fundamentals, Design, and Implementation* (11th Ed. b.). New Jersey: Prentice Hall.

[191] *Op.Cit.* (Weinstock, 2010)

instances. They are of the same type, but not identical.

This thesis argues that with GDS, and specifically genetic algorithms, the generation of multiple solutions from the same schema may form a *design family*. With GA, a design family can be generated by instantiating new individuals.[192] The population is generated as an evolutionary lineage.[193] As previously discussed, such population contains design instances they are of the same type but variations of the schema.[194] In a population, design instances that share generic characteristics form a design family by demonstrating similar phenotype/physical characteristics. In design search spaces the formation of design families through the population may propose a potential management method based on generic phenotype characteristics.

---

[192] *Op.Cit.* (DeLanda, 2002)

[193] *ibid.*

[194] *ibid.*

CHAPTER 3

VISUALITY AND GENERATIVE DESIGN SYSTEMS

Chapter III aims to discuss the role of visuality in GDA and particularly in GA, and how it has changed the understanding of the act of design and the role of the designer.

3.1. Visuality, Design and GDS

Human-environment interaction is built on visuality and based on visual interaction.[195] From the perception of the environment to aesthetic judgement, humans interact with their environment predominantly with visual information.[196] Design is a visual act,[197] and visuality is one of the principal characteristics of design. The designer's interaction with the design problem and the solution co-evolves with the flow of visual information.[198] Furthermore, visualization assists the correlation and communication of other disciplines or designers by demonstrating designer's individual approach to design problem and the design solution.[199]

Traditionally in design, visual representations facilitate the communication of the design artefacts and ideas.[200] However, in GDS, the broad solution space of the design alternatives makes visual representation of design individuals, therefore visuality, unmanageable for designers and decreases the visualization.[201] Accordingly, the lack

---

[195] Koutamanis, A. (2000). Digital Architectural Visualization. *Automation in Construction, 9,* pp. 347-360.

[196] *ibid.*

[197] *Op.Cit.* (Visser, 2009)

[198] *Op. Cit.* (Kolarevic, 2000)

[199] *Op. Cit.* (Koutamanis, 2000)

[200] *ibid.*

[201] *ibid.*

of mediatory visual aid that can facilitate a more effective interaction between the designer and the design space challenges design search act. Therefore, the designerly involvement and evaluation during design synthesis is hindered. Consequently, in generative design, there runs a risk of a decrease in the active role that the designer plays in the design synthesis process.

Specifically in optimization processes (GA processes) that aim to converge to the global optima; the system generates a large number of design variations but directs the process towards one optimal solution. The tendency to convergence to a global optima may reduce the design process to a solely problem-solving process rather than the exploration of design alternatives. This may cause the whole range of design instances to be disregarded. Moreover, the designer may need to evaluate other design instances because the optimal solution may not be the satisficing solution in correspondence to the subjective evaluation criteria of a designer. Within design search space there may be a design instance that is suitable for both designer's subjective criteria and fitness criteria. Here, this thesis supports that visual representations can play an important role in design exploration for the subjective evaluation of generated design instances.

3.2. Visual Resemblance

[…] resemblance remains an appropriate vehicle for investigating perceptual and cognitive issues involved in visual perception.

Alexander Koutamanis
Digital Architectural Visualization, 2000

Resemblance is a state of being or looking similar to something.[202] In this thesis, it is considered that the resemblance is constituted by the common features of design instances that is generated by the same schema. The designer designates the schema and the schema determines the common features. These common features are associated with Arnheim's concept of 'structural pattern' in correspondence to shape

[202] Hornby, A. S. (2005). Resemblance. *Oxford Advanced Learner's Dictionary of Current English*. Oxford: Oxford University Press.

perception and visual reasoning and Holland's concept of 'recognizable iterative pattern' in correspondence to generative design synthesis and design instances. Throughout this thesis identified common features are named as *generic characteristics*. Also, in this thesis the concept of resemblance is grounded on Arnheim's description of shape perception and process of visual reasoning.

Arnheim (1969) asserts that "[t]he perception of the shape is the grasping of structural features found in, or imposed upon, the stimulus material." Accordingly, the perception of a shape is based on its generic structural characteristics that are perceived by the observer.[203] Here, there are two important aspects; (1) **perception** is based on the main features that are captured by the observer, and (2) this perception is observer-dependent, therefore is **subjective.**

The perception of a shape can also be correlated with the 'abstraction' as a mechanism of visual reasoning.[204] Visual reasoning is based on the perception of an object followed by the engagement of an abstraction and generalization processes.[205] Abstraction in perception leads the observer to perceive the generic characteristics of a shape without considering the details, which leads to generalization.[206] Holland defines this abstraction process as model-building, which discards the details in order to capture the essential characteristics of a shape.[207]

Resemblance in relation to visual reasoning associates with the subjective recognition of the repeating generic characteristics of a set of design instances. As the designer identifies the generic characteristics that are common to the design instances, he/she can form resemblance relations between them.

D'arcy Thompson's ideation of form as a "diagram of forces" can be discussed to explain visual resemblance in nature. Thompson claims that continuous transformations apply on form, and therefore form is a diagram of forces. Thompson

---

[203] Arnheim, R. (1969). *Visual Thinking.* California: University of California Press Berkeley and Los Angeles.

[204] Les, Z., & Les, M. (2008). *Shape Understanding System.* Berlin: Springer-Verlag.

[205] *ibid.*

[206] *ibid.*

[207] *Op.Cit.* (Holland, 1998)

proposes a mathematical model to define this transformation process using the Cartesian coordinate system. This system mathematically explains the transformation of form in a species based on the external forces.[208] This process can yield in countless variations in a species. The common formal characteristics that are observed in each individual can be explained with the concept of the *homology.*[209] (Fig. 22).
During evolution, new species emerge through transformation. Each species shares generic characteristics with the others. By demonstrating these generic characteristics, the species demonstrate visual resemblance.

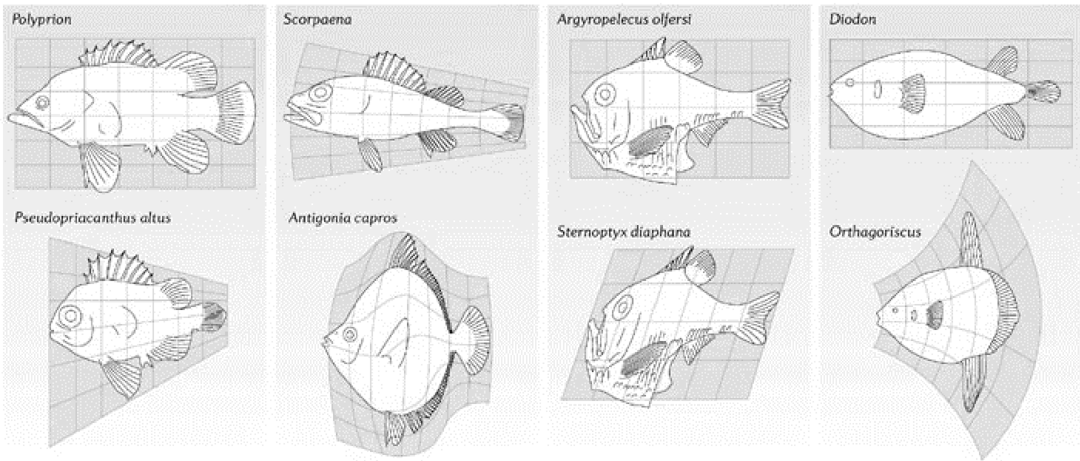

Figure 22 Form is a deformation of another form, Thompson (Retrieved from: http://www.nature.com/nrg/journal/v7/n5/fig_tab/nrg1835_F3.html, 31.01.2014)

Resemblance in relation to generative design synthesis, as discussed above, the generic characteristics of the design instances are determined by a schema. [210] Each design instance is unique, but also resembles other alternative design instances as they share the same schema. A design instance can be distinguished from another as based on visual similarity or difference.

For instance, the generic characteristic of the Folium design instances is the branching pattern. (Fig. 23) In more details, 2nd -8th -9th -11th design instances resemble each other

[208] Thompson, D. W. (1917). *On Growth and Form.* Cambridge University Press.

[209] *ibid.* (pp, 12, 13).

[210] *Op.Cit.*(Holland, 1998)

more than the other instances by demonstrating A-like form as a generic characteristic. (Fig. 24)

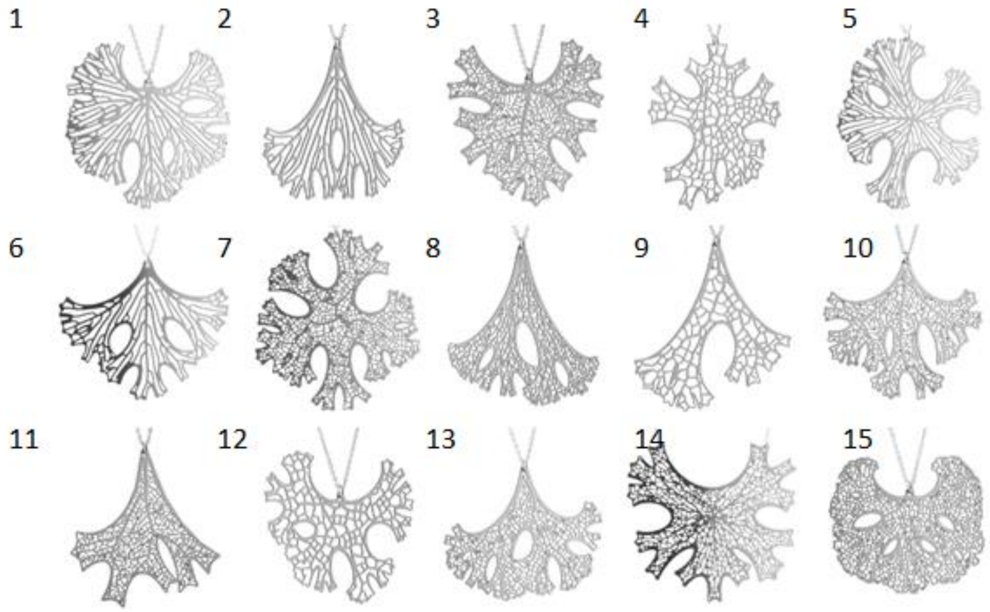

Figure 23 'Folium' Design Instances (Retrieved from: http://n-e-r-v-o-u-s.com/blog/?p=3983, 11.05.2014.)

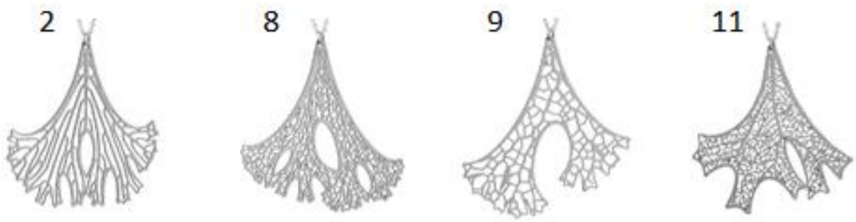

Figure 24 A Group of Resembling Design Instances

As an illustration, there has been an experimental research based on shape exploration conducted by McKay et al. that documents the transformational process of designs.[211] In this research, design professionals are identified, these professionals are required to complete given tasks of different design processes. In one experiment, a group of industrial designers are asked to design a kettle starting from an initial shape. (Fig.25)

[211] Lim, S., Prats, M. P., Jowers, I., Chase, S., Garner, S., & McKay, A. (2008). Shape Exploration in Design: Formalising and Supporting a Transformational Process. *International Journal of Architectural Computing*, *6*(4), 415-433.

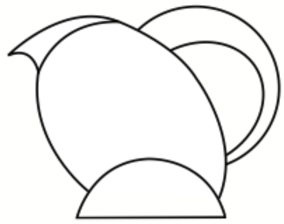

Figure 25 **Initial Concept of the design task (Retrieved from: "Shape Exploration in Design: Formalising and Supporting a Transformational Process by McKay et al.)**

Throughout the design process industrial designers used certain shape rules to transform the initial form. As a result, design families emerge according to the visual commonness between the design instances. The design instances that share similar handles constitute a design family, and the design instances that share the similar kettle bases constitute another design family.[212] (Fig.26)

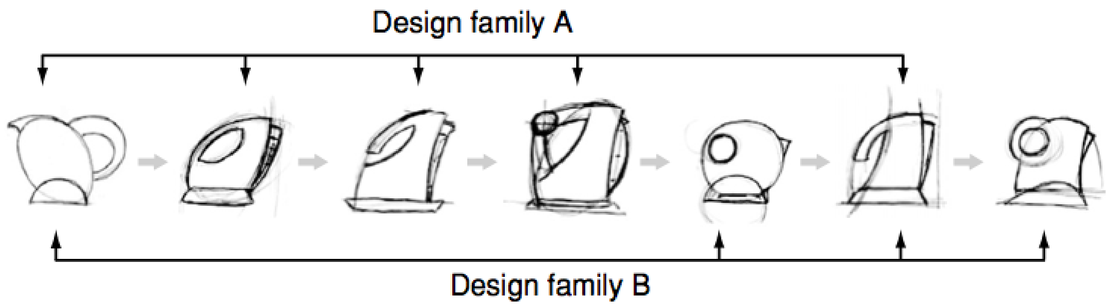

Figure 26 **Design Families (Retrieved from: "Shape Exploration in Design: Formalising and Supporting a Transformational Process by McKay et al.)**

The kettles' common physical features determine the design families, which corresponds to the concepts of repeating patterns that Holland[213] discusses.
The formation of design families corresponds with the concept of type. (See chapter 2.4.) The identification of generic characteristics that are observed through design instances enables the organization of the design search space by categorizing design instances according to resemblance. Such categorization enables the formation of types as an abstraction of a class of resembling design instances. Each type forms a

---

[212] *ibid.*

[213] *Op.Cit.*(Holland, 1998)

different design family. Here, with the formation of design families the complex design search spaces can be structured in correspondence with resemblance.

Resemblance that is a result of the repetition of certain features corresponds to concept of visual language in graphic and communication design. Visual language is the consistent and repetitive use of design elements such as colour, texture and/ or font.[214] Visual language enhances the recognisability and therefore is also exploited by product design and brand identity.[215] The recognisability corresponds with the designer's recognition of generic characteristics through the design instances. In fine arts, for example, visual language can be the texture by heavy and broken brush strokes in the paintings below can point out to a distinctive language of paintings. (Fig. 27)
The repetition of the curvilinear edges through the Apple Inc.'s product family and the use of the same colour scheme for iPhone and iPod product families constitutes visual consistency. (Fig. 28a) The use of charcoal colour scheme in a way that composes white background with black figures are the repeating design elements in RAF magazine. Also the font and the location of 'RAF' title through the same paper layout strengthen the design identity and recognisability. (Fig. 28b)

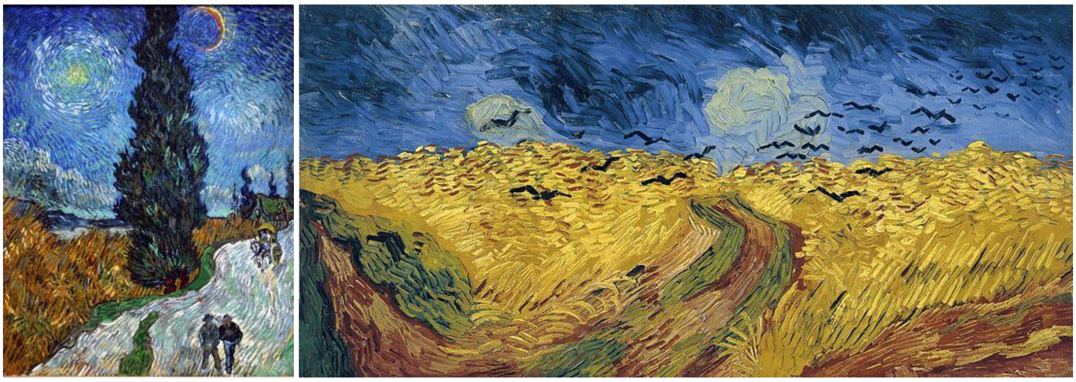

Figure 27 Vincent Van Gogh Paintings: Country Road in Provence by Night (left-top), Wheatfield with Crows (right-top) (Retrieved from: (1)http://en.wikipedia.org/wiki/Wheatfield_with_Crows, (2)http://en.wikipedia.org/wiki/File:Van_Gogh_-_Country_road_in_Provence_by_night.jpg, 05.03.2014.)

---

[214] Lidwell, W., Holden, K., & Butler, J. (2003). *Universal Principles of Design.* Beverly, MA: Rockport Publishing.

[215] *ibid.*

Figure 28 28a-Apple Inc. Product Series and 28b-Raf Product Magazine- Exploitation of visual consistency in product and graphic design (Retrieved from: http://www.apple.com/ and http://v1.raf.com.tr/competitioncovers.php, 25.02.2014.)

In architecture, the resemblance between Frank Lloyd Wright’s Prairie Houses is a result of the use of similar building elements and features such as brick, continuous transverse façade, low roof, continuous window sills and monolithic chimneys.[216] (Fig. 29 to 32)

[216] *ibid.*

Figure 29 Robie House by Frank Lloyd Wright, 1910 (Retrieved from: http://arthistorygalore.files.wordpress.com/2011/03/robie-house-by-flw.jpg, 02.01.2014)

Figure 30 Thomas House by Frank Lloyd Wright, 1901 (Retrieved from: http://www.peterbeers.net/interests/flw_rt/Illinois/Frank_Wright_Thomas_House/Frank_Wright_Thomas_House.htm, 02.01.2014)

Figure 31 Willits House by Frank Lloyd Wright, 1901 (Retrieved from: http://en.wikipedia.org/wiki/File:Willits_House.jpg, 02.01.2014.)

Figure 32 Henderson House by Frank Lloyd Wright, 1901 (Retrieved from: http://www.prairiestyles.com/images/architects/wright/henderson.jpg, 02.01.2014.)

When Zaha Hadid's design work (architecture, installations, and urban design, interior design and product design) is considered, the use of elongated, continuous and transforming curvatures appear as a common visual characteristic of her designs. Such similarities underline a common language in her design work. (Fig. 33)

Frank Gehry designs result in resemblance by the use of folding surfaces as a building façade and form. Also in some Gehry buildings, the folding façades creates rhythm by repeating rectangular windows that are embedded into the surface. The metal siding that is use for the surface coating material forms resemblance in between Gehry buildings. Furthermore, the fragmentation that is observed in Gehry buildings strengthens the deconstructivist approach of Frank Gehry. (Fig. 34)

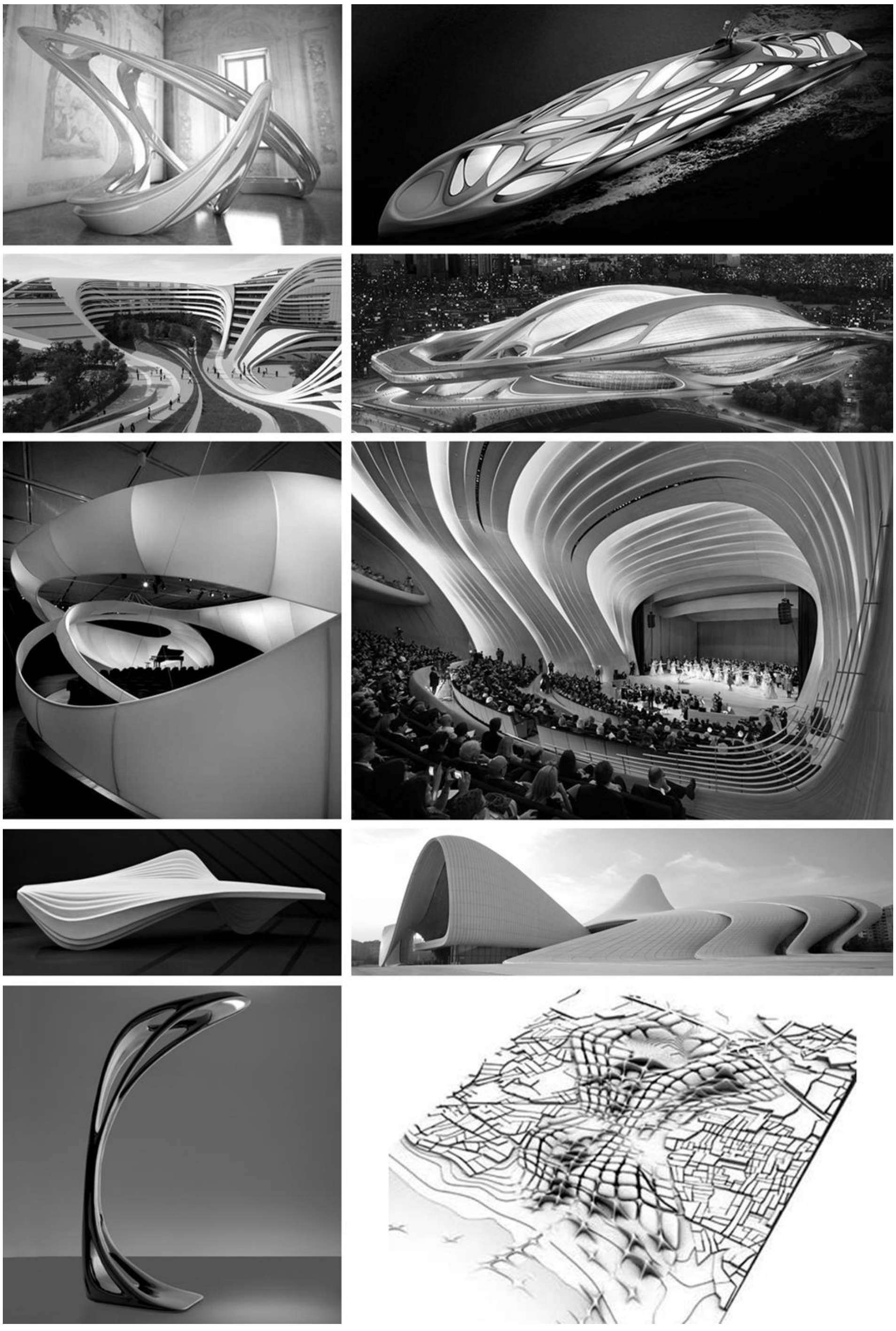

Figure 33 Zaha Hadid Design (10.05.2014, Retrieved from:

1-http://www.dezeen.com/2008/08/10/aura-by-zaha-hadid-architects/

2-http://www.wallpaper.com/architecture/zaha-hadid-designs-superyacht-for-blohm-voss/6843

3-http://www.designboom.com/architecture/zaha-hadid-beko-masterplan-in-belgrade/

4-http://www.dezeen.com/2013/10/25/zaha-hadids-tokyo-stadium-to-be/

http://www.designboom.com/architecture/zaha-hadid-architects-js-bach-chamber-music-hall/

5-http://www.designboom.com/architecture/new-images-of-heydar-aliyev-center-by-zaha-hadid-11-14-2013/

6-http://www.interiorholic.com/other/furniture/futuristic-bench-by-zaha-hadid/

7-http://architecture.yale.edu/gallery/heydar-aliyev-cultural-center

8-http://www.dezeen.com/2009/04/27/genesy-by-zaha-hadid-for-artemide/

9-http://www.arcspace.com/features/zaha-hadid-architects/kartal--pendik-masterplan/

Figure 34 Frank Gehry Buildings (10.05.2014, Retrieved from:

1-http://www.dezeen.com/2010/06/17/lou-ruvo-center-for-brain-health-by-frank-gehry/

2-http://www.arch2o.com/shotcrete-experience-music-project-frank-ghery/

3-http://collectingmythoughts.blogspot.com.tr/2011_11_01_archive.html

4-http://urbantoronto.ca/forum/showthread.php/2547-Art-Gallery-of-Alberta-(formerly-Edmonton-Art-Gallery)-goes-to-Stout/page2

5-http://www.flickriver.com/photos/bill_in_stl/3424745763/

6-http://www.starwoodhotels.com/resorts/explore/member_favorites.html?awardId=1006311832

3.3. Revisiting the Research Problem

In non-computational design synthesis processes, designer is in direct contact with the design artefact by means of visual representations (digital and physical models, sketches, drawings). However, as Kolarevic (2000) points out, the role of visual representations in computational design synthesis has been transformed to generation process.[217] Here, there may arise a problem that the design artefact generated by an algorithm may distance the visual representation from the designer. Specifically in generation processes of genetic algorithms, the designer gets involved only at the beginning (by determining the design problem, problem constraints, parameters and parameter boundaries).[218] (Fig.35) Here, this indirect relationship between the designer and the design artefact might reduce the designer's dialogue with the artefact, thereby limiting with design space exploration. Furthermore, **due to the automation of design synthesis and the weakened interaction between the designer and the design artefact, the subjectivity of the act of designing and the creative design processes are challenged**. However, design is a creative production process that is based on subjective decisions and evaluation.[219] The priorities and modalities of the designer characterize the design process and the design solution.[220] Design fosters the selection of the "satisficing" design solution which depends on designer's subjective evaluation and decision on the satisfactory and convenience in means of design requirements.[221, 222] Here, the subjectivity in design evaluation contradicts to objective-evaluation based design processes.

---

[217] *Op.Cit.* (Kolarevic, 2000)

[218] Kolarevic, B. (2003). Architecture in the Digital Age: Design and Manufacturing. New York: Spon Press, pp.24.

[219] *ibid.*

[220] *Op. Cit.* (Visser, 2009)

[221] *Op. Cit.* (Simon, 1970).

[222] Simon, H. A. (1956). Rational Choice and the Structure of the Environment. *Psychological Review, Vol: 63*(No: 2), 129–138.

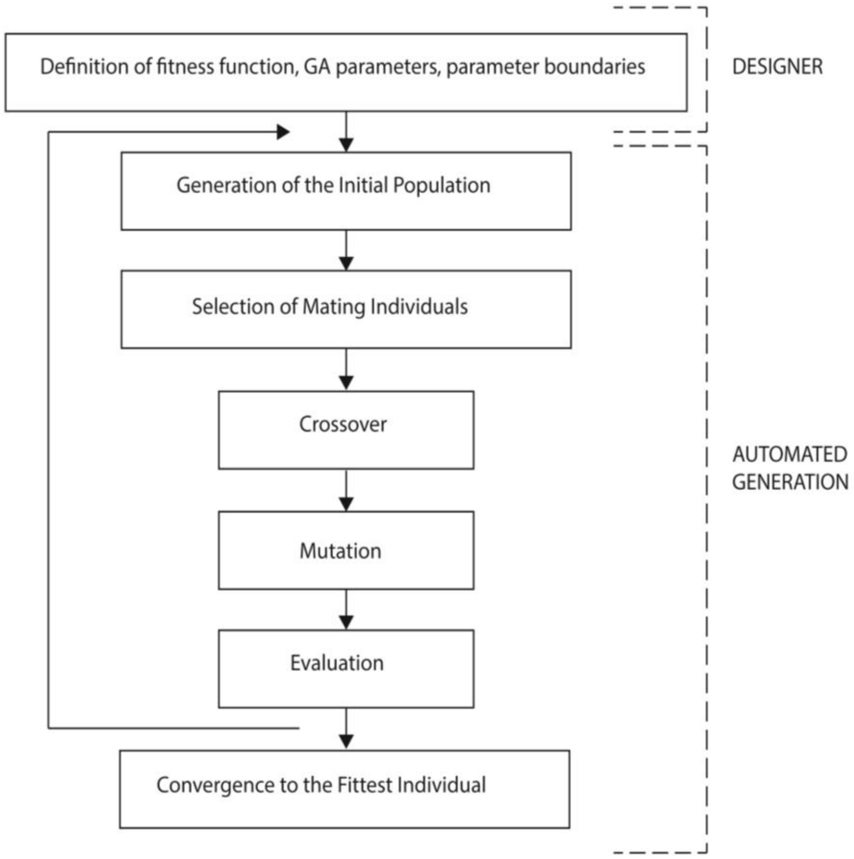

Figure 35 Designer and GA Process

Furthermore, with generative design systems, the generation of a large number of design alternatives broaden the design search space. **This causes problems in the exploration of the design alternatives.** The inefficient management of design search space might challenge the dialogue between the designer and the design object that obstructs the design space navigation of the designer. Moreover, as they are large and high-dimensional solution spaces, the visualization of design search space becomes almost impossible. Therefore, the designer is not able to visually communicate with a set of solutions.

According to the identified problems throughout the design generation processes of GA, this research proposes solutions (1) that can facilitate the mediation between design generation and the subjectivity of the creative design by the amplification of the designer's involvement in the design synthesis via providing a visual representation of design individuals and (2) to manage the complexity of large design search spaces and facilitate the designer's involvement in the decision-making process following generation. (Fig. 36) Here, the visual representation of the design space proposes the possibility to consider a wider range of design alternatives and enable designerly evaluation of design search spaces. Within the scope of visual structuring, several approaches are discussed in following chapters but not limited to: (1) perception-based, (2) retrieval-based and (3) optimality-based visual structuring approaches.

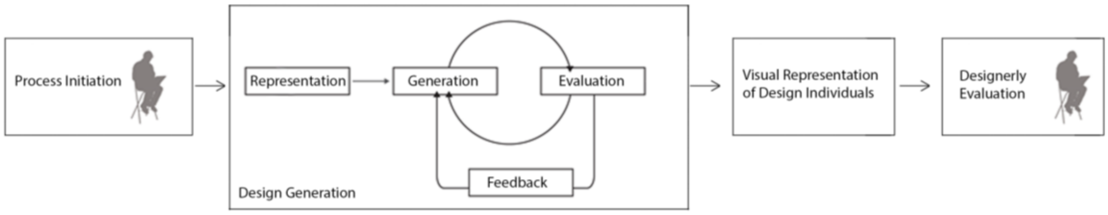

Figure 36 **The Proposed Model of Interaction between the Designer and the Design Synthesis Process**

CHAPTER 4

STRUCTURING THE DESIGN SEARCH SPACE

Chapter IV presents and discusses the role of visual structuring in the design search space based on the design search spaces of genetic algorithms as a method to manage large solution spaces. Several approaches for design search space structuring are proposed and discussed.

4.1. Visual Structures and Design Search Space

Search characterizes design as a path planning problem through a space of possibilities.

Robert F. Woodbury

Searching for Designs: Paradigm and Practice, 1989

Design search space is a concept that plays an important role in design exploration. A design search space contains a set of possible solutions for a design problem.[223] In the context of this thesis, such possible solutions are referred as design instances, or individuals in genetic algorithms. As discussed above, GA uses genetic processes that generate many design instances and large solution spaces. To illustrate, genetic algorithms conduct design space exploration that fosters a search and evaluation for the optimum by re-combining the genetic substance of successive generations. Generative design methods have an influence on design search, as design search space is broadened and its level of complexity increased.[224] Such broad search spaces can be challenging for the designers to explore and navigate.

---

[223] *Op. Cit.* (Cagan, Campbell, Finger, Tomiyama, 2005)

[224] *ibid.*

This thesis proposes that structuring the design search space can help manage its complexity and support its navigation by the designer to evaluate different design alternatives, thereby increasing his/her involvement in the process. As such, a design instance can be selected by the designer not only based on optimality as framed by the generative system (GA), but also the subjective values and considerations of the designer.

According to Stouffs (2006), design space exploration requires the representation of many designs, the structural organization that forms the space, and facilitating the designer's exploration of this space which also involves the production of new designs and to move among previously discovered designs in the network. [225] Design is a process of creative exploration, generation and problem-solving [226] and visual representations play an important role in design exploration. However, as discussed previously (see Chapter 3.1) visual representations are not being used in GA processes which obstructs the visual interaction of the designer. Accordingly, this chapter discusses solutions that can increase the visual interaction and subjective evaluation of a designer within design spaces of GA. Accordingly, within the scope of this research, several considerations in structuring of design search spaces will be discussed, including the subjective perception of designers, the concept of optimality and resemblance.

Visual structuring is meant to give structure to the parts (design instances) of a design search space that is to guide the interrelations between the design individuals. Visual structures can support the process of design search. In GA, the designer is in contact with the design artefact by means of symbolic representations, but not in dialogue with the artefact's formal and geometric features. As a solution for this challenge, the visual structuring of design search space that may take place after the generation process may assist the amplification of designer's dialogue with the design instances and the design

---

[225] *Op .Cit.* (Stouffs, 2006)

[226] *Op. Cit.* (Simon, 1970).

search space and the strategies to structure the design space that reconnect the designer with the design artefact during generative process are necessary. Here, visual structuring has the potential to establish a visual dialogue between the design search space and the designer. Visual structures can be used in the visualization of the design search space, and increase the designer's involvement in the evaluation process by through a comparative evaluation between the design instances.

4.2. Visual Structures and Resemblance Relations

Previously, Arnheim's approach to perception of resemblance (see chapter 3.2) has been discussed. This thesis proposes the use of visual structures based on the visual resemblance between design instances of generative systems. Such visual structures first require the identification of the generic characteristics of design instances. These generic characteristics give way to a classification of individuals.

Forming categories is one of the major abilities of human cognition, and the formation of categories is based on similarity of some kind.[227] Similarity, also named as family-resemblance, points out to the resemblance of features between family members.[228] Family-resemblance manifests itself in the common physical features (hair colour, eye colour, skin colour, hair-type, etc.) that siblings share. In case of genetic algorithms, as design instances (individuals) form a design family, family-resemblance can be observed between design instances. Accordingly, resemblance in generative systems motivates this research to choose visual resemblance as a criterion for visual structures. Formation of resemblance relationships between the design instances enables the comparative evaluation of design alternatives. Furthermore, grouping the resembling design instances in a design search space has the possibility to reduce the complexity

---

[227] Couchman, J. J., Coutinho, M. V., & Smith, J. D. (2010). Rules and Resemblance: Their Changing Balance in the Category Learning in Humans (Homo Sapiens) and Monkeys (Macaca Mulatta). *J Exp Psychol Anim Behav Process., 36*(2), 172–183.

[228] *ibid.*

in design search space. Such grouping gives way to organization of the design search space.

4.3. Several Visual Structuring Behaviours

Different methods can support the build-up of visual structures. Within the scope of this thesis, structuring behaviours used to build visual structures include, but are not limited to: **(1) perception-based, (2) retrieval-based** and **(3) optimality-based.**

4.2.1. Perception-Based Structuring Behaviour

Perception-based structuring is intuitive, case-based, observer-dependent and subjective act which is based on the designer's perception and identification of common features/ visual resemblance between the design instances. Each structure is unique and personal. These visual structures have a potential to amplify designerly evaluation and the subjectivity throughout the generation process.

Perception-based structuring behaviour requires manual build-up of visual structures and such structuring behaviour is applicable for refined, manageable and small design search spaces. Designer identifies the hierarchical resemblance relationships between the design instances and groups them according to their visual similarity. Such groups are termed as *resemblance clusters*. All design instances can be represented in these visual structures or filtration can be conducted in order to decrease the number of design instances for the formation of resemblance clusters in a design search space. (Fig. 37). However, manual build-up of visual structures may take a long time and require too much effort for a designer.

In design search spaces, each parameter can lead to a separate clustering. In case of multi-parameters, the act of classification is multi-phase; such that one clustering follows another, forming *sub-clusters*. Visually distinguishing first parameter structures the whole classification and this influences the whole taxonomy. As each designer has different visual perception, visually distinguishing parameters demonstrate differences for each designer. Therefore, each designer demonstrates different clustering behaviour and each time clustering behaviour changes according to the perception of a designer.

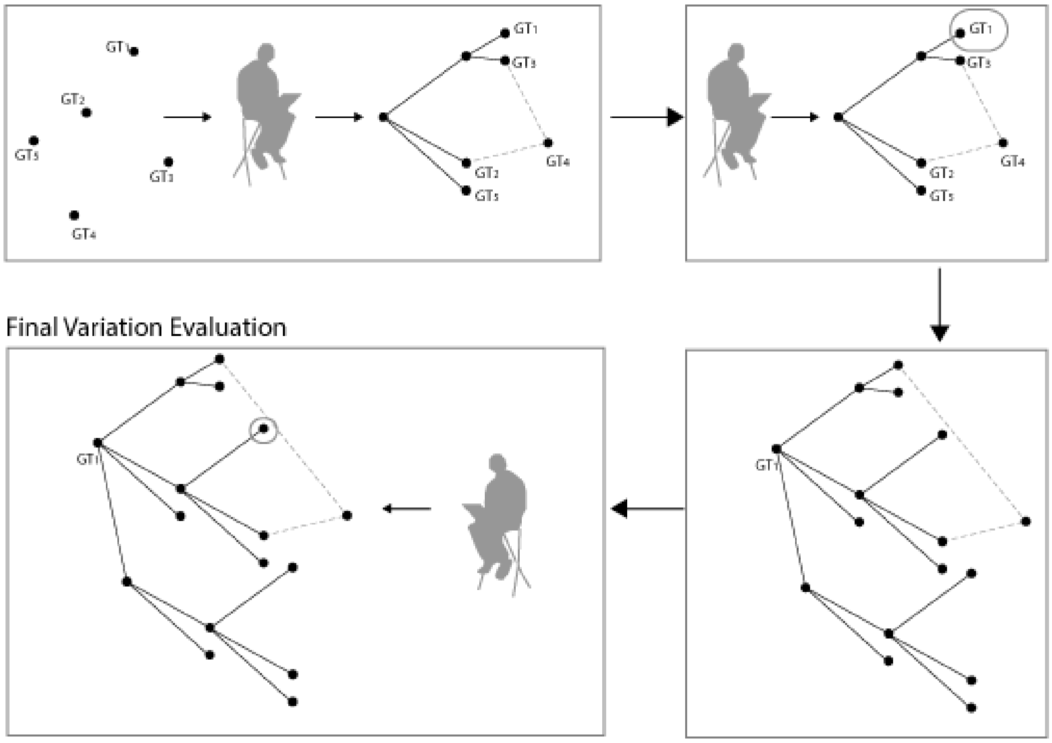

Figure 37 **Designer and Perception-Based Visual Structures**

A case study on the formation of Resemblance Clusters:

To illustrate the formation of resemblance clusters, a case-study is conducted within the scope of this thesis. An existing design search space that is experimented by Soddu is used for this case-study which the design instances are generated by the evolutionary generative mechanism *IDEA.*[229] *IDEA* defines the generation and transformation procedure, but does not have an evaluation mechanism that forces the design space to converge towards the fittest, as in genetic algorithms. Soddu states that the generative mechanism without optimization is chosen on purpose to explore all possible design variations.[230] As a result, unique chairs emerged.[231]

---

[229] Soddu, C. (2005). Generative Art in Visionary Variations. *Art+Math=X Proceedings.* University of Colorado Boulder.

[230] *ibid.*

[231] *ibid.*

The chairs are given to three designers to organize resemblance clusters and a brief statement for their grouping criteria. (Fig. 38). The designers are post-graduate level architects with average one-year professional experience. Architects are chosen as an experiment group due to their perception of a design object as just a form. Their task is to categorize this design space according to resemblance relationships between design instances. An image editing software tool was used by these designers to visually structure and organize the given design search space.

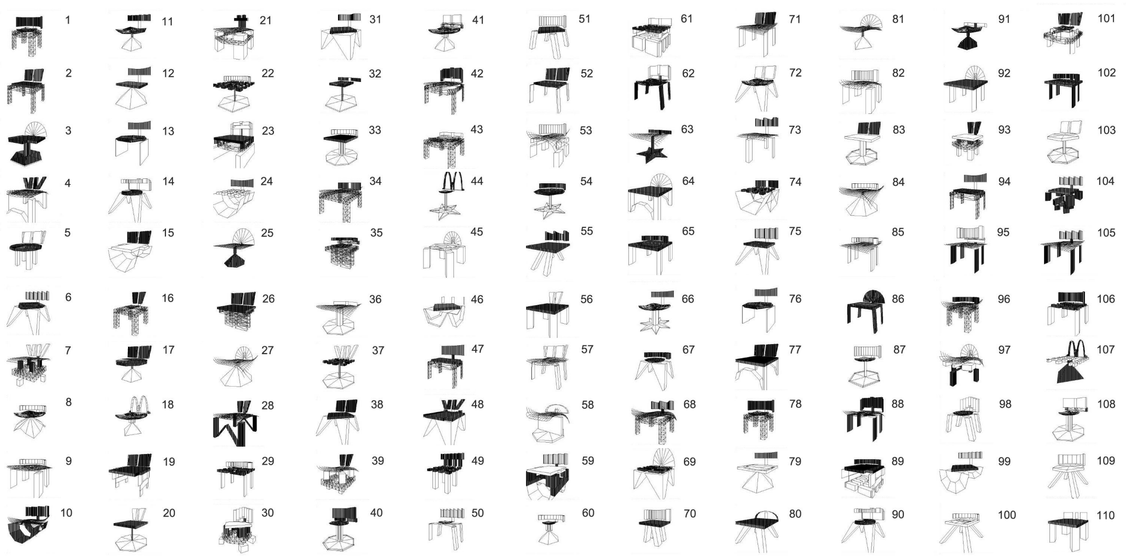

Figure 38 Design instances of Soddu (For bigger image please see Appendix A)

Designer A:

The designer A selected the chair legs as the visually distinguishing feature of the design space. Here, the criteria for the formation of resemblance clusters is the types of the chair legs. Accordingly, she identifies eleven types of chair legs and divided design search spaces into eleven clusters. (Fig. 39, 40).

Figure 39 Eleven Different Chair-legs indicated by Designer A

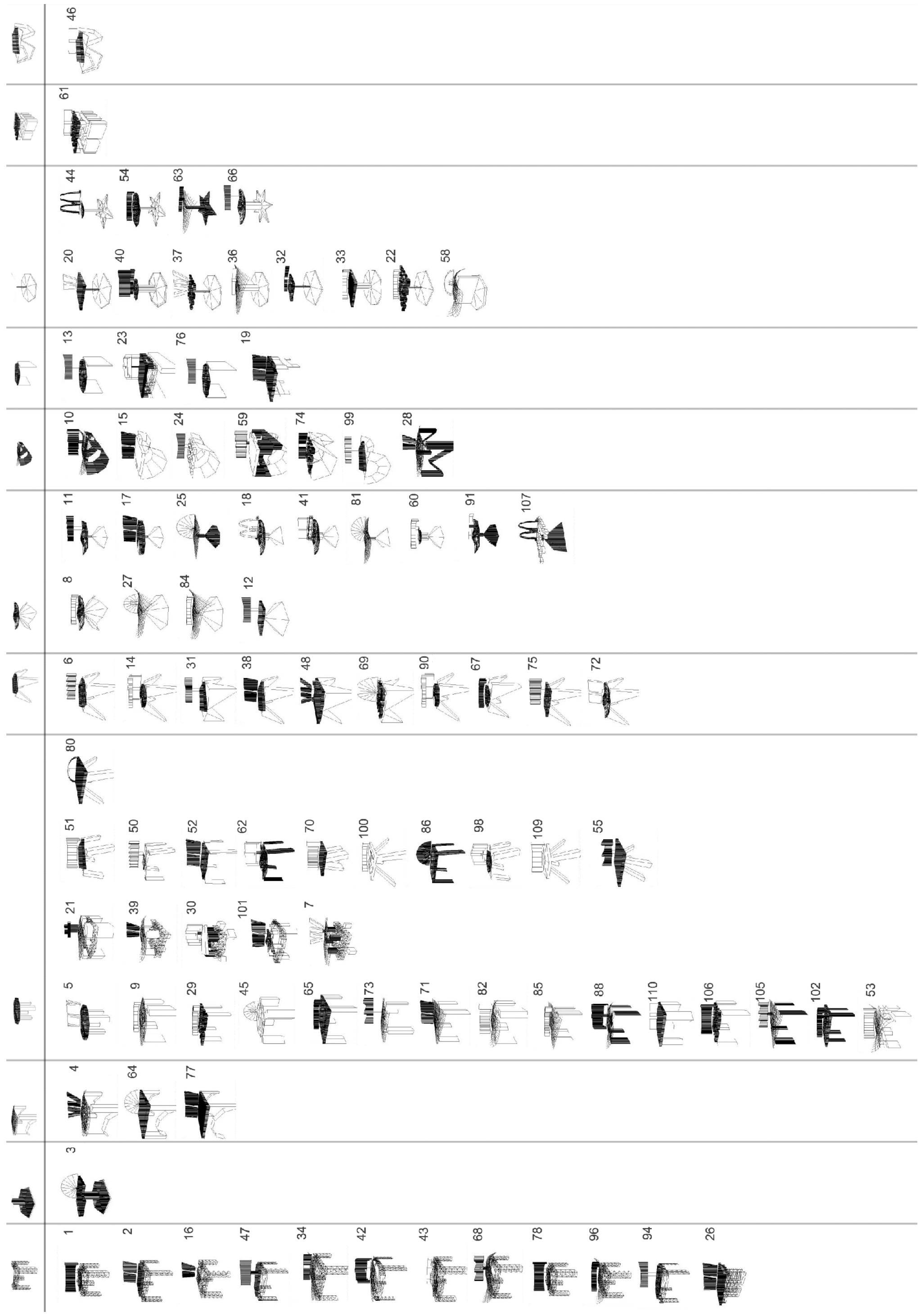

Figure 40 Resemblance Clusters of Designer A (For bigger image please see Appendix B)

Designer B:

According to designer B, there are five resemblance clusters based on the form and number of chair legs; cluster A is formed according to the straight angle chair legs, cluster B is according to the circular base of chair legs, cluster C is formed according to the widening angle of chair legs, cluster D is according to the box-form of chair legs and cluster E is formed according to the rounded form of the chair legs. In each cluster, there are five sub-clusters based on the form of the backrests; (1) singular, (2) dual, (3) triple, (4) elongated and (5) semi-circular. (Fig. 41). As such, an internal structure inside of each cluster is formed.

Figure 41 Resemblance Clusters of Designer B (For bigger image please see Appendix D)

Designer C:

According to designer C, there are 2 criteria for the formation of resemblance clusters; (1) backrests, and (2) chair legs. Primarily, designer C formed 15 resemblance clusters according to the backrests of the chairs and within these resemblance cluster the designer C formed sub-clusters based on chair legs. (Fig. 42)

Figure 42 Resemblance Clusters of Designer C (For bigger image please see Appendix F)

The resemblance relations are structured on the common/ repeating elements. For instance, similar chair-legs form a resemblance cluster for Designer A and chair-leg forms and number for Designer B. For designer C, backrests form a resemblance cluster. (Fig.43). The visually distinguishing criteria that form resemblance clusters are different, but the identification of common features which causes resemblance is the same tendency to form a resemblance cluster. Accordingly, as discussed previously resemblance clusters are formed according to the designer's perception and identification of common/ repeating elements. Therefore, the formation of resemblance clusters, accordingly the build-up of visual structures, is a subjective and designer-dependent.

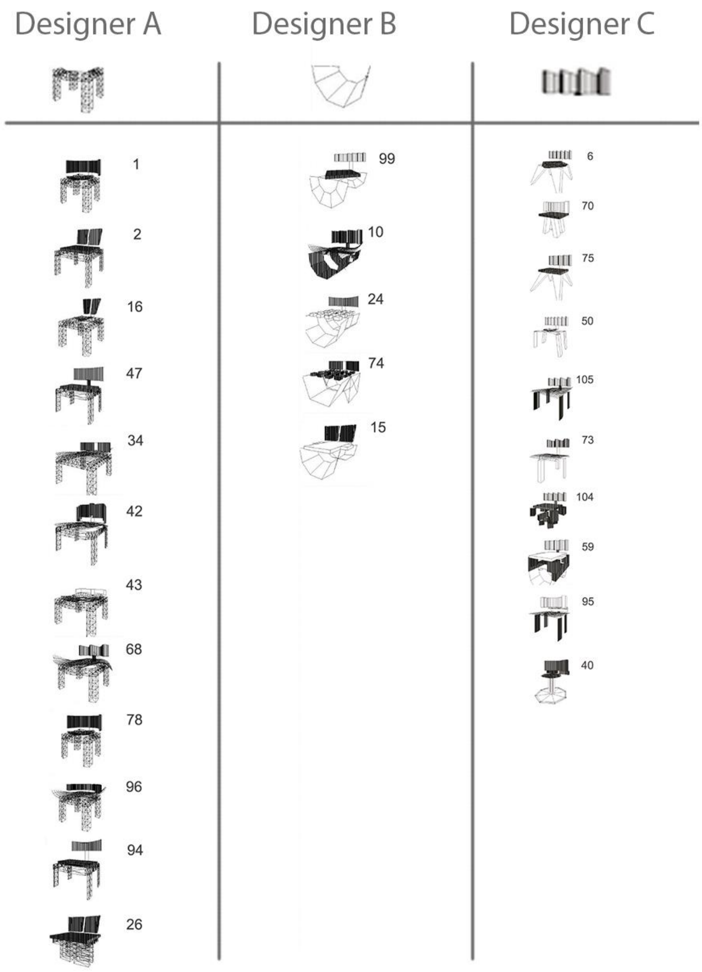

Figure 43 The repetition of similar chair-legs, backrests and the texture in resemblance clusters

4.2.2. Retrieval-Based Structuring Behaviour

Retrieval-based structuring is based on the exploitation of 3D object retrieval methods for the visual structuring of large design search spaces. Automated build-up of visual structures are required in retrieval-based structuring. Such structuring behaviours are objective filtration processes and have a potential to identify similarity between design instances after the generation process. These visual structures have a potential to structure large design search spaces with automated structuring which may amplify the designerly interaction in design search spaces.

3D object retrieval methods are geometry-based search algorithms that find use in genetics and engineering.[232] They are used for the evaluation of similarity and the classification of 3D objects based on benchmarking.[233] 3D object search requires the shape descriptor which defines and represents the features and information of an object and 3D object repository that provides a benchmarking scheme.[234] In the 3D object retrieval research, benchmarking schemes are used for the evaluation of the retrieval algorithms.[235] The evaluation is based on testing and comparison of the search algorithms and provides a feedback for 3D objects retrieval methods.[236] Besides its research purposes, benchmark schemes are used for the comparison of various types of multimedia data such as images and models.[237] However, for the evaluation of images, there may appear inaccuracies in means of description of the geometry.[238] The features of the object may not be completely analysed from the image.[239] For instance,

---

[232] Fang, R., Godil, A., Li, X., & Wagan, A. (2008). A New Shape Benchmark for 3D Object Retrieval. *ISVC 2008 Proceedings* (pp. 381-392). Berlin Heidelberg: Springer- Verlag.

[233] *ibid.*

[234] *ibid.* (pp. 382)

[235] *ibid.* (pp. 384)

[236] *ibid.*

[237] Funkhouser, T., Kazhdan, M., Min, P., & Shilane, P. (2005, June). Shape-Based Retrieval and Analysis of 3D Models. *Communications of the ACM*, *48*(6), 58-64.

[238] *ibid.*

[239] *ibid.* (pp. 62)

for the evaluation of the chair images; all of the legs of a chair may not be visible depending on the perspective of the image and this may cause inaccuracies for the evaluation. Unlike images, the evaluation of 3D models may overcome such problems of image-based methods.[240]

In architecture, 3D object retrieval methods are used for 3D models and their classification in digital object libraries such as class of structural elements, building elements and furnishing elements as they are classified according to their similarity in form and function.[241] Furthermore, 3D model retrieval methods facilitate indexing and storage of 3D media in modelling software.[242]

In general terms, similarity evaluation is conducted by matching the geometries and geometric features of compared objects in a benchmark scheme. [243] According to the matching methods, 3D object retrieval based on geometry-search can be grouped into two classes as; (1) shape-based and (2) topology-based.[244] For shape-based retrieval methods similarity evaluation is conducted by the distribution of vertices and polygons which decomposes and defines the object's geometry with polygons and vertices.[245] The topology-based methods uses the object's topology for the similarity evaluation.[246]

3D object retrieval is based on four phases as; (1) query formation, (2) feature extraction, (3) dissimilarity computation and (4) retrieval. [247] Query formation is the selection of the 3D object to be compared; feature extraction is the determination of

---

[240] *ibid.*

[241] Wessel, R., Blümel, I., & Reinhard, K. (2009). A 3D Shape Benchmark for Retrieval and Automatic Classification of Architectural Data. *Eurographics Workshop on 3D Object Retrieval*.

[242] Berndt, R., Blümel, I., & Raoul, W. (2010). PROBADO3D – Towards an Automatic Multimedia Indexing Workflow for Architectural 3D Models. *In proceedings of 14th International Conference on Electronic Publishing* (pp. 79-88). Hanken School of Economics.

[243] Chen, D.-Y., Tian, X.-P., Shen, Y.-T., & Ming, O. (2003). On Visual Similarity Based 3D Model Retrieval. *EUROGRAPHICS, 22*(3), 223-232.

[244] *ibid.*

[245] *ibid.*

[246] *ibid.*

[247] Ohbuchi, R., Nakazawa, M., & Takei, T. (November 7, 2003). Retrieving 3D Shapes Based On Their Appearance . *MIR'03 Proceedings* (pp. 39-46). Berkeley, California, USA.: ACM.

the features of the selected object; dissimilarity computation is based on the comparison of the object in a 3D object repository; and the retrieval phase is based on the detection of the objects which have the lowest dissimilarity value obtained in the third phase.[248] (Fig. 44)

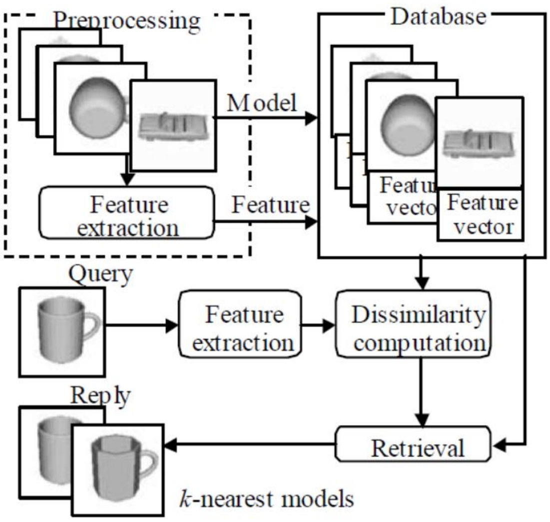

Figure 44 "A shape similarity search database for 3D shapes." (Retrieved from: Ohbuchi, R., Nakazawa, M., & Takei, T. (November 7, 2003). Retrieving 3D Shapes Based On Their Appearance. MIR'03 Proceedings (pp. 39-46). Berkeley, California, USA: AC)

Within the context of visual structures, 3D benchmarking process may be used at the end of the design synthesis process, the design search space can be considered as a 3D model repository, a benchmarking scheme. The designer identifies the reference object that is used for benchmarking. (The methods/ways to generate/identify the reference object can be a subject of another research. Within the scope of this thesis, it is assumed that there is a reference object that is identified/ generated by the designer.) The reference object can be used to query the visually similar design instances. Also in correspondence with type/instance discussion in chapter 2.4, the reference object is the type that defines the group of similar design instances. Furthermore, the identification of the reference object becomes the filtration criteria through the benchmarking process. After the identification of the reference object, the retrieval process carries out the comparative evaluation between design instances in a design search space. With such evaluation 3D retrieval algorithm identifies similar design instances.

[248] *Op.Cit.* (Chen et.al. 2013, pp. 40.)

Therefore, resemblance clusters are formed accordingly in an automated fashion. (Fig. 45).

The retrieval-based visual structuring process is conducted objectively. Here, the automated generation of resemblance clusters is not as subjective as perception-based visual structuring and from this point such structures challenge the designer's evaluation and subjectivity throughout the formation of resemblance clusters. However, retrieval-based visual structuring facilitates designer' interaction to complex design search spaces by structuring the design search space. Furthermore, there are developing 3D object retrieval methods that simulate the designerly actions of classifications and detection of similarity.[249]

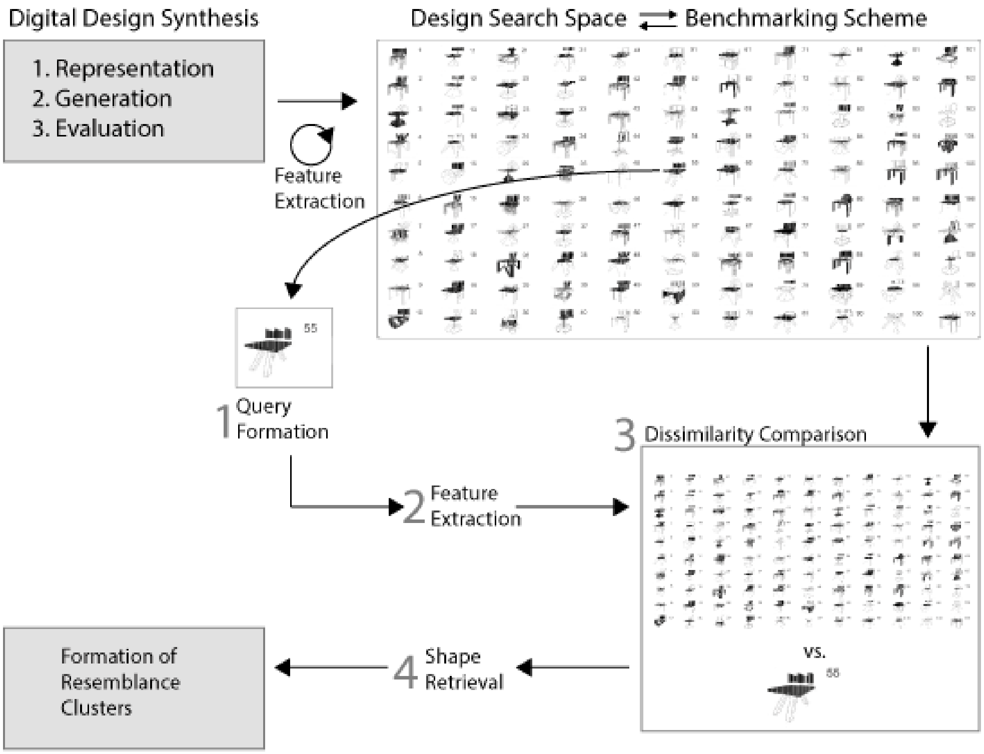

Figure 45 **Diagram for Retrieval-Based Structuring Behaviour**

[249] *Op.Cit.* (Chen et.al., 2013, pp. 40.)

4.2.3. Optimality-Based Structuring Behaviour

Optimality-based structuring is an objective filtration behaviour that is based on the optimality for large design search spaces. The visualization/ analogue representation and clustering of design instances which are located on fitness peaks in a fitness landscape are centric for such structuring behaviour. In comparison to perception-based and retrieval-based visual structuring behaviours, the relationships between the design instances are not structured in optimality-based visual structuring. Here, such structures end up with a group of non-structured design instances and inform designers about formal properties and fitness states of the design instances.

The most common method for visualization of GA is based on fitness evaluation of genomes illustrated by fitness-time graphs. [250] Sewall Wright suggests fitness landscapes as a visual metaphor to depict gene combination space and the evolution of a population. [251] Fitness landscapes illustrate the fitness status for generated population according to the fitness criteria.[252] Wright depicts fitness landscapes with hills and valleys that generated populations move through.[253] Here, the optimization process associates with hill-climbing metaphor that signifies the movement of population through regions of low fitness to high fitness, a random walk through a surface in a three-dimensional space. [254,255]

---

[250] Wu, H.-C., Sun, C.-T., & Lee, S.-S. (2004). Visualization of Evolutionary Computation Processes From a Population Perspective. *Intelligent Data Analysis , 8*, 543-561.

[251] Zaman, L., Ofria, C., & Lenski, E. R. (2012). Finger-Painting Fitness Landscapes: An Interactive Tool for Exploring Complex Evolutionary Dynamics. *Artificial Life 13,* 499-505.

[252] Gavrilets, S. (2004). High-dimensional Fitness Landscapes and Speciation. In M. Pigliucci, & K. Preston, Phenotypic Integration: Studying the Ecology and Evolution of Complex Phenotypes (pp. 45-80). New York: Oxford University Press.

[253] *Op.Cit.* (Zaman et.al.; 2012; pp.499)

[254] *ibid.* (pp.501)

[255] McCandlish, D. M. (2011, June). Visualizing Fitness Landscapes. *65*(6), pp. 1544-1558.

In Wright’s illustration, the fitness landscape is demonstrated in two dimensions; and dotted lines are the areas formed according to the adaptiveness of the population.[256] (Fig. 46).

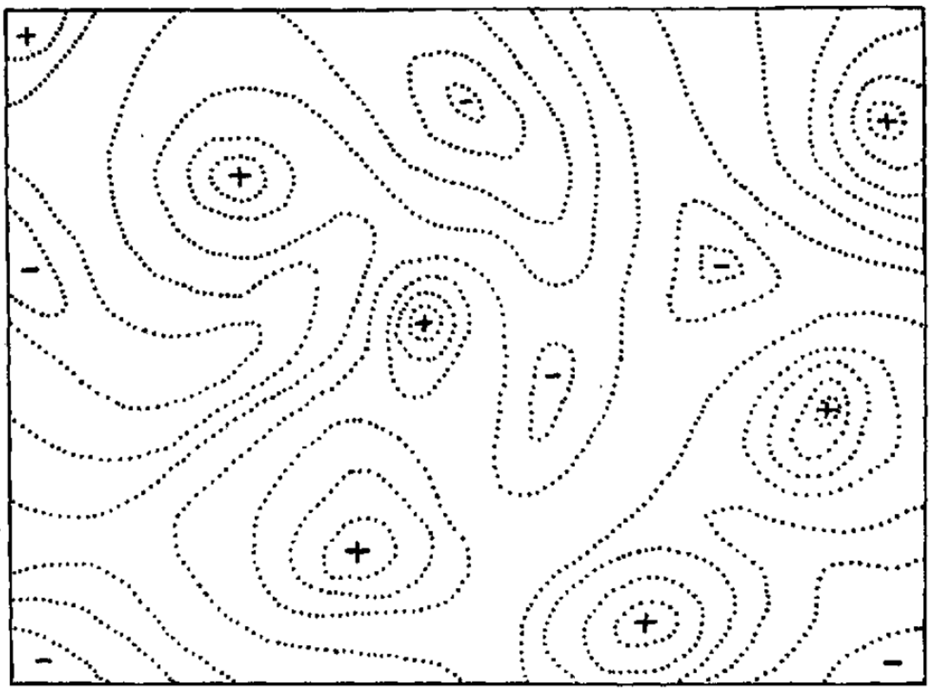

Figure 46 Sewall Wright’s Illustration of Fitness Landscape (Retrieved From: The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution by Sewall Wright, 1932)

Fitness landscapes can visualize high-dimensional search spaces by low-dimensional representations; and fitness landscape visualization is limited to maximum two dimensions.[257] Therefore higher than two degrees cause problems to visualize search spaces. But with the developments in current data visualization technologies, new methods for complex search space visualization models based on dimension reduction are developing.

One of these developing methods is proposed by McCandlish based on the segregation of the genomes that are located at different fitness peaks.[258] By segregating the genomes on different peaks, the major features of the peak can be displayed

---

[256] Wright, S. (1932). The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. *Proceedings of the VI International Congress of Genetics*, (pp. 356-366).

[257] *Op.Cit.* (McCandlish, D. M.; 2011)

[258] *ibid.*

graphically.[259] Here, McCandlish (2011) proposes symbolic representation of design instances at the same peak. For optimality-based visual structuring, McCandlish's method can be used as long as the design instances visualized with analogue representation. The segregation of design instances that are located on different peaks automatically structures design search space according to the fitness scores of design instances. The identification and visualization of these design instances may inform designer about formal features of design instances that belong to the same peak. This facilitates the visual interaction and guidance of a designer through the design search space. Furthermore, due to the convergence in the optimization processes, design instances at the same peak are visually similar. Such similarity has a potential to form resemblance clusters after the generation phase. Furthermore, the determination of major features of fitness regions corresponds with the type.

As optimality-based visual structuring behaviour demonstrates objective filtration based on fitness scores, such structuring may pose a challenge for the subjectivity of designerly evaluation. However, these structures reduce the complexity in a design search space by limiting the number of design instances with the number of the design instances that are located at the peaks. This reduction facilitates the designerly interaction with refined and smaller design search spaces. Here, such structuring behaviour may amplify designerly evaluation of the filtered design search spaces.

4.4. Findings and Results

1. The perception of the same / similar feature in a set of design is defined as resemblance. Resemblance in a design space can be used as an operational method to build visual structures.
2. According to the case studies, each designer built a different structure using the same design instances according to different resemblance criteria for grouping. This study highlights the subjectivity of the designer and designerly evaluation.

[259] *ibid.*

3. Within the framework of visual structures, three visual structuring behaviours are proposed and discussed as; (1) perception-based, (2) retrieval-based and (3) optimality-based.

   Perception-based structuring behaviour is subjective and based on designer's perception of visual resemblance; and structures visual resemblance relations between the design instances for small design search spaces.

   Retrieval-based structuring behaviour is objective and based on the exploitation of 3D retrieval search algorithms which is based on the detection and filtering in a design search spaces according to visual similarity.

   Optimality-based structuring behaviour is also objective and based on the filtration due to the optimality. Different from perception-based and retrieval-based structuring, optimality based structuring does not structure relationships between the design instances.

	DSS	**Structuring**	**User-Behaviour**	**Build-Up**	**Represented Individuals**
Perception-Based	Small	Hierarchical	Subjective	Manual	All/ Filtered
Retrieval-Based	Small/Large	Hierarchical	Objective	Automated	Filtered
Optimality-Based	Small/Large	NH/Sorted	Objective	Automated	Filtered

In perception-based the designer is directly in dialogue with complete set of design instances. In comparison with perception-based structuring behaviour, the designer is in dialogue with filtered and structured design search spaces. Here, in perception-based the designer is involved through the whole process of visual structuring, therefore the visual structuring process ends up with subjectivity. However, in retrieval-based and optimality based structuring, the designer does not get involved through the automated visual structuring. Therefore, subjectivity may decrease and is not centric in these behaviours. Despite the subjectivity states of such behaviours, both of them proposes refined and structured design search spaces that designer can easily interact.

Accordingly, these structuring methods provide visual guidance to the designer and facilitate the designerly interaction and evaluation after the design synthesis.

CHAPTER 5

CONCLUSION

Throughout this research, generative design systems and genetic algorithms are investigated, and visual structuring to manage the complexity of the large solution spaces is discussed. Within the scope of this research, first and foremost it is aimed to analyse and understand generative design systems, and characteristics of GDS are explained from a theoretical point of view. The complexity of computational design synthesis and the understanding of a design artefact are different from conventional understanding of design. With the change in the understanding of the design generation process, the designer's contribution and way of interacting with the 'artefact(s)' have changed. Design search space is enlarged as the generation of design solutions is automated, weakening the designer's contribution, evaluation and visual contact with the design space. As design is a subjective search and visual evaluation process, the designerly actions must be amplified for generative design strategies. As the design search space is enlarged, there is a need for methods that facilitate the navigation of the designer within the solution space.

Accordingly two main problems identified in this thesis are (1) the broad design search spaces of GDS that are populated by many design instances and (2) the elimination of designerly decisions and evaluation from the generation phase due to automated generation of GA. As a solution for these problems, visual representations (structures) are proposed as a mediator between the designer and the design artefact during the generative process. Such structures are proposed as a support for the designerly evaluation of design instances. Three structuring behaviours (perception-based, retrieval-based, and optimality-based) are proposed, discussed and compared. For future research, this thesis proposes a set-up for the implementation of the suggested methods as computational tools. With the implementation of such methods, designers will be in dialogue with the design search spaces of generative design systems visually.

As perception-based visual structuring behaviour requires a manual build-up of visual structures, it is not applicable for large design search spaces. Therefore, the large design spaces of generative design systems pose a challenge for the future use of this method for complex search spaces. Retrieval-based visual structuring behaviour can be implemented for form-centric design processes. With the implementation of such structuring methods, designer will be able to filter the design instances according to his/her reference design and will be able to evaluate similar design instances.

When these three behaviours are considered for the future research and the implementation in practising design areas, optimality-based visual structuring has a potential for the professional use regarding to the propagation in performance-oriented design and architecture. Furthermore, optimality-based visual structuring methods remain capable of mediating between the designer's formal decisions and optimization by allowing designer to evaluate the optimization and the formal properties together. Therefore, such structures will be able to accommodate form and the performance, in other words, the objectivity of optimization and the subjectivity of designerly evaluation.

Other than these methods, this thesis draws several conclusions based on the literature overview and the case study. For instance, forms emerge as intermediate states of transformation. The most distant forms emerge between the initial and the end states of the system, the system that converges to the form of the fittest design instance.
From the case study and its theoretical background based on Arnheim's visual reasoning, the repetition of certain elements recalls the plausibility of the visual resemblance. The common elements/ generic characteristics (within the scope of this thesis, visual elements such as the patterns, textures and components of design instances) in design instances structure resemblance relations and form resemblance clusters. Furthermore, as the identification of generic characteristics is observer-dependent, each designer can structure different resemblance relations between the

design instances. Each designer built different resemblance relations with different resemblance criteria based on the most distinguishing characteristic(s) that they perceive. Accordingly, this case study concludes with the emphasis on subjectivity of the designer and designerly evaluation. Even in design generation strategies that foster the automation of the design generation, designer and a designerly evaluation must be a part of the generation process which supports subjectivity and contributes to the diversity and richness of design.

As a conclusion, visual guidance is needed through the generative design processes to support the designerly evaluations and the decisions in computational design synthesis. If these visual structuring methods are utilized as computational tools, as they value the outcomes of the objective design synthesis processes as well as the formal decisions of a designer, such methods have a potential to be exploited in areas of professional design and architecture.

REFERENCES

Abraham, A., Nedjah, N., & de Macedo Mourelle, L. (2006). Evolutionary Computation: from Genetic Algorithms to Genetic Programming. *Studies in Computational Intelligence (SCI), 13,* 1–20 .

Akın, Ö. (2001). Variants in design cognition. In C. Eastman, M. Mike, & N. Wendy, *Design Knowing and Learning: Cognition in Design Education: Cognition in Design Education* (pp. 105-125). Netherlands: Elsevier.

Arnheim, R. (1969). *Visual Thinking.* California: University of California Press Berkeley and Los Angeles.

Batty, M., Xie, Y., & Sun, Z. (1999). Modeling Urban Dynamics Through GIS-based Cellular Automata. *Computers, Environment and Urban Systems, 23,* 205-233.

Berndt, R., Blümel, I., & Raoul, W. (2010). PROBADO3D – Towards an Automatic Multimedia Indexing Workflow for Architectural 3D Models. *In proceedings of 14th International Conference on Electronic Publishing* (pp. 79-88). Hanken School of Economics.

Brownlee, J. (2011). *Clever Algorithms: Nature-Inspired Programming Recipes.* LuLu Press.

Burns, A. M. (2010). Mathscapes- Fractal Scenery. In D. Gulick, & J. Scott, *The Beauty of Fractals: Six Different Views* (pp. 1-21). The Matchematical Association of America (Incorporated).

Burrow, A., & Woodbury, R. (1999). π-Resolution in Design Space Exploration. In G. Augenbroe, & C. Eastman, *Computers in Building* (pp. 291-308). Springer US.

Cagan, J., Campbell, M. I., Finger, S., & Tomiyama, T. (2005, September). A Framework for Computational Design Synthesis: Model and Applications. *Journal of Computing and Information Science in Engineering* (Vol.5), 171-181.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-organization in Biological Systems.* New Jersey: Princeton University Press.

Cannon, J. W., Floyd, W. J., & Parry, W. R. (2001). Finite Subdivision Rules. *Conformal Geometry and Dynamics, Vol. 5* , 153–196.

Carpo, M. (2013). The Ebb and Flow of Digital Innovation: From Form Making to Form Finding- and Beyond. *Architectural Design Special Issue: The Innovation Imparative: Architectures of Vitality,* 56-61.

Chen, D.-Y., Tian, X.-P., Shen, Y.-T., & Ming, O. (2003). On Visual Similarity Based 3D Model Retrieval. *EUROGRAPHICS, 22*(3), 223-232.

Coates, P., & Carranza, P. M. (2000). The Use of Swarm Intelligence to Generate Architectural. *In Proceedings of the 3rd Generative Art Conference Generative Art.* Milan, Italy: AleaDesign Publisher.

Coenders, J., & Bosia, D. (2006). Computational Tools for Design and Engineering of Complex Geometrical Structures. In K. Oosterhuis, & L. Feireiss, *Game Set And Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies* (pp. 271-278). Rotterdam: Episode Publishers.

Couchman, J. J., Coutinho, M. V., & Smith, J. D. (2010). Rules and Resemblance: Their Changing Balance in the Category Learning in Humans (Homo Sapiens) and Monkeys (Macaca Mulatta). *J Exp Psychol Anim Behav Process., 36*(2), 172–183.

Cross, N. (2001). Design Cognition: Results From Protocol and Other Empirical Studies of Design Activity. In C. Eastman, M. McCracken, & W. Newstetter, *Design Knowing and Learning: Cognition in Design Education: Cognition in Design Education* (pp. 79-105). Netherlands: Elsevier.

Cross, N. (2011). *Design Thinking: Understanding How Designers Think and Work.* New York: Berg Publishers.

Cross, N. (Summer 2001). Designerly Ways of Knowing: Design Discipline Versus Design Science. *Design Issues, Vol. 17*(3), 49-55.

Çelik, Z. (1986). *The Remaking of Istanbul: Portrait of an Ottoman City in the Nineteenth Century.* Seattle: University of Washington Press.

DeLanda, M. (1997). *A Thousand Years of Nonlinear History.* New York: Zone Books.

DeLanda, M. (2002). Deleuze and the Use of the Genetic Algorithm in Architecture. In N. Leach, *Designing for a Digital World.* New York: Wiley.

Dodder, R., & Dare, R. (2000, October 31). Complex Adaptive Systems and Complexity Theory: Inter-related Knowledge Domains. Cambrigde MA, USA: MIT Press.

Dorst, K. (2003). *Understanding Design.* Amsterdam: BIS Publishers.

Dorst, K., & Cross, N. (2001). Creativity in the Design Process: Co-evolution of Problem–solution. *Design Studies* (22), 425–437.

Fang, R., Godil, A., Li, X., & Wagan, A. (2008). A New Shape Benchmark for 3D Object Retrieval. *ISVC 2008 Proceedings* (pp. 381-392). Berlin Heidelberg: Springer- Verlag.

Flake, G. W. (2000). *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation.* Cambridge, MA: MIT Press.

Fleischer, M. (2003). Foundations of Swarm Intelligence: From Principles to Practice. *SWARMING: NETWORK ENABLED C4ISR.* McLean, Virginia.

Frazer, J. (1995). *An Evolutionary Architecture.* London: Architectural Association.

Fromm, J. (2005, June 13). *Types and Forms of Emergence.* Retrieved May 23, 2013, from Cornell University Library: http://arxiv.org/abs/nlin/0506028

Funkhouser, T., Kazhdan, M., Min, P., & Shilane, P. (2005, June). Shape-Based Retrieval and Analysis of 3D Models. *Communications of the ACM, 48*(6), 58-64.

G. Dino, İ. (2012). Creative Design Exploration By Parametric Generative Systems in Architecture. *Middle East Technical University Journal of the Faculty of Architecture* , 204-224.

Gavrilets, S. (2004). High-dimensional Fitness Landscapes and Speciation. In M. Pigliucci, & K. Preston, *Phenotypic Integration: Studying the Ecology and Evolution of Complex Phenotypes* (pp. 45-80). New York: Oxford University Press.

Gell-Mann, M. (1994). Complex Adaptive Systems. In G. A. Cowan, D. Pines, & D. Meltzer, *Complexity: Metaphors, Models, and Reality* (pp. 17-45). Addison-Wesley.

Gero, J. S. (1993). Towards a Model of Exploration in Computer-Aided Design. In J. S. Gero, & F. (. Sudweeks, *Formal Design Methods for CAD (pre-prints)* (pp. 271-291). IFIP, University of Sydney.

Goetsch, W. (1957). *The Ants.* Ann Arbor, MI: University of Michigan Press.

Gutowitz, H. (1990). *Cellular Automata: Theory and Experiment.* Amsterdam, Netherlands: Elsevier Science Publishers.

Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms.* Hoboken, New Jersey: John Wiley & Sons.

Hensel, M., Menges, A., & Weinstock, M. (2004, May). Emergence: Morphogenetic Design Strategies. *Architectural Design, 74*(3), 6-9.

Hensel, M., Menges, A., & Weinstock, M. (2010). *Emergent Technologies and Design: Towards a Biological Paradigm for Architecture.* Oxon: Routledge.

Herr, C. M., & Kvan, T. (2005). Using Cellular Automata to Generate High-Density Building Form. In B. Martens, & A. Brown, *Computer Aided Architectural Design Futures 2005* (pp. 249-258). Springer Netherlands.

Holland, J. (1998). *Emergence: From Chaos to Order.* Oxford: Oxford University Press.

Holland, J. (2011). Constrained Generating Procedures. In A. Menges, & S. Ahlquist, *Computational Design thinking* (pp. 131-142). Chichester, UK: John Wiley & Sons.

Holland, J. (Winter, 1992). Complex Adaptive Systems. *Daedalus, A New Era in Computation*. MIT Press.

Hornby, A. S. (2005). Resemblance. *Oxford Advanced Learner's Dictionary of Current English*. Oxford: Oxford University Press.

Johson, S. (2001). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software.* New York: Scribner.

Jones, G. (2002, April 15). Genetic and Evolutionary Algorithms. *Encyclopedia of Computational Chemistry*.

Kalay, Y. E. (2004). *Arhitecture's New Media: Principles, Theories, and Methods of Computer-Aided Design.* Cambridge, Massachusetts: The MIT Press.

Kauffman, S. (1995). *At Home in the Universe: The Search for Laws of Self-Organization and Complexity.* New York: Oxford University Press.

Kennedy, J., & Eberhart, R. C. (2001). *Swarm Intelligence.* San Diego, CA: Academic Press.

Knight, T. (1993). Color Grammars: The Representation of Form and Color in Designs. *Leonardo, Vol. 26*(No. 2), 117-124.

Knight, T. (1999). *Applications in Architectural Design, and Education and Practice.* Cambridge, MA: Massachusetts Institute of Technology.

Knight, T. (1999). *Applications in Architectural Design, and Education and Practice* . NSF/MIT, Department of Architecture. Cambridge, MA: MIT.

Knight, T. (2001). Either/Or And. *Proceedings 3rd International Space Syntax Symposium Atlanta.* Atlanta: University of Michigan.

Kolarevic, B. (2000). Digital Morphogenesis and Computational Architectures. *Proceedigns of the 4th Confernece of Congreso Iberoamericano de Grafica Digital*, (pp. 98-103). Rio da Janeiro.

Kolarevic, B. (2003). *Architecture in the Digital Age: Design and Manufacturing.* New York: Spon Press.

Koutamanis, A. (2000). Digital Architectural Visualization. *Automation in Construction, 9*, 347-360.

Krish, S. (2011). A Practical Generative Design method. *Computer-Aided Design*(43), 88–100.

Kroenke, D. M., & Auer, D. (2009). *Database Processing: Fundamentals, Design, and Implementation* (11th Ed. ed.). New Jersey: Prentice Hall.

Leach, N. (2009). Digital Morphogenesis. *Architectural Design, 79*(1), 32-37.

Les, Z., & Les, M. (2008). *Shape Understanding System.* Berlin: Springer-Verlag.

Lidwell, W., Holden, K., & Butler, J. (2003). *Universal Principles of Design.* Beverly, MA: Rockport Publishing.

Lim, S., Prats, M. P., Jowers, I., Chase, S., Garner, S., & McKay, A. (2008). Shape Exploration in Design: Formalising and Supporting a Transformational Process. *International Journal of Architectural Computing, 6*(4), 415-433.

Manson, S. M. (2001). Simplifying Complexity: A Review of Complexity Theory. *Geoforum*(32), 405-414.

Marcuse, P. (2007). The Grid as City Plan: New York City and Laissez-Faire Planning in the Nineteenth Century. *Planning Perpectives*, 287-310.

Mayr, E. ((1976) 1994). Typological versus Population Thinking. In E. Sober, *Conceptual Issues in Evolutionary Biology: An Anthology* (pp. 156-160). Cambridge, MA: The MIT Press.

McCandlish, D. M. (2011, June). Visualizing Fitness Landscapes. *65*(6), pp. 1544-1558.

McCormack, J. (2004). Generative Modelling With Timed L-Systems. In J. S. Gero, *Design Computing and Cognition '04* (pp. 157-175). Dordrecht, Netherlands: Kluwer Academic Publishers.

Menges, A., & Ahlquist, S. (2011). Introduction. In A. Menges, & S. Ahlquist, *Computational Design Thinking* (pp. 10-29). Chichester, UK: John Wiley & Sons.

Mitchell, M. (1996). Genetic Algorithms: An Overview. *Complexity*, *1*(1), 31-39.

Mitchell, M. (2006). Complex Systems: Network Thinking . *Science Direct*, 1195-1210.

Mitchell, M. (2009). *Complexity: A Guided Tour*. New York: Oxford University Press.

Mitchell, W. (1990). The Logic of Architecture. In L. Hee Won, *The Computation of Design Vocabulary: Prototyping, Variation, and Composition* .

Mitchell, W. J., Liggett, R. S., & Tan, M. (1989). Top-Down Knowledge-Based Design. *CAAD futures Digital Proceedings* (pp. 137-148). Boston: The MIT Press.

Ohbuchi, R., Nakazawa, M., & Takei, T. (November 7, 2003). Retrieving 3D Shapes Based On Their Appearance . *MIR''03* (pp. 39-46). Berkeley, California, USA.: ACM.

Oracle Corporation. (2013, 11 26). *Oracle-Define an entity*. Retrieved from Oracle: http://docs.oracle.com/html/E24270_01/Content/Data%20model/Define_an_entity.htm

Oxford University Press. (2013). *Emergence*. Retrieved May 16, 2013, from Oxford Dictionaries: http://oxforddictionaries.com/definition/english/emergence?q=emergence

Oxman, R. (1997, October). Design by Re-representation: a Model of Visual Reasoning in Design. *Design Studies, Vol.18*(Issue 4), 329-347.

Oxman, R. (2002). The Thinking Eye: Visual Re-Cognition in Design Emergence . *Design Studies, 23*(2), 135-64.

Oxman, R. E., & Oxman, R. M. (1992). Refinement and Adaptation: Two Paradigms of Form Generation in CAAD. *CAAD futures Digital Proceedings 1991* (pp. 313-328). Braunschweig/Wiesbaden: Friedrich Vieweg & Sohn Verlagsgesellschaft mbH.

Prusinkiewicz, P., & Lindenmayer, A. (1990). *Algorithmic Beauty of Plants.* New York: Springer- Verlag.

Rittel, H., & Webber, M. (1973). Dilemmas in General Theory of Planning. *Policy Sciences, Volume 4*(Issue 2), 155-169.

Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic Algortihms. In E. K. Burke, & G. (. Kendall, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 96-127). New York, NY: Springer Science+ Business Media LLC.

Schön, D. A. (1983). *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books Inc.

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

Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards Integrated Performance-Driven Generative Design Tools. *Automation in Construction*(14), 253-264.

Shiffman, D. (2013, 12 22). *Chapter-8 Fractals*. Retrieved from The Nature of Code: http://natureofcode.com/book/chapter-8-fractals/

Simon, H. .. (1973). The Structure of Ill-structured Problems. *Artificial Intelligence*(4), 181-201.

Simon, H. A. (1956). Rational Choice and the Structure of the Environment. *Psychological Review, Vol: 63*(No: 2), 129–138.

Simon, H. A. (1970). Style in Design. *EDRA 2: Proceedings of the 2nd Annual Environmental Design Research Association Conference.* Pittsburgh: Carnegie-Mellon University.

Sober, E. (1980, September). Evolution, Population Thinking, and Essentialism. *Philosophy of Science,, Vol. 47*(No. 3), 350-383.

Sober, E. (2006). *Conceptual Issues in Evolutionary Biology*. Cambridge, MA: The MIT Press.

Soddu, C. (2005). Generative Art in Visionary Variations. *Art+Math=X Proceedings.* University of Colorado Boulder.

Soddu, C. (2006). Generative Design. A swimmer in a natural sea frame. *Generative Art International Conference.* Milan.

Spiller, N. (2008). *Digital Architecture Now: A Global Survey of Emerging Talent.* London, UK: Thames & Hudson.

Stiny, G. (1984). Computing with Meaning and Form in Architecture. *Journal of Architectural Education, 39*(1), 7-19.

Stiny, G. (1993). Emergence and Continuity in Shape Grammars. *CAAD Futures '93* (pp. 37-54). Pittsburgh: Elsevier Science Publishers B.V.

Stiny, G., & Gips, J. (1972). Shape Grammars and the Generative Specification of Painting and Sculpture. *nformation Processing, 1460-1465. 71,* pp. 125-135. Amsterdam: North- Holland.

Stouffs, R. (2006). Design spaces: The explicit representation of spaces of alternatives. *AIE EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 20*(2), 61-62.

Terzidis, K. (2011). Algorithmic Form. In A. Menges, & S. Ahlquist, *Computational Design Thinking* (pp. 94-102). London: Wiley.

Thompson, D. W. (1917). *On Growth and Form.* Cambridge University Press.

Trummer, P. (2011). Associative Design: From Type to Population. In A. Menges, & S. Ahlquist, *Computational Design Thinking* (pp. 179-197). Chichester, UK: John Wiley & Sons.

Visser, W. (2009). Design: One, But in Different Forms. *Design Studies, 30*(3), 187-223.

Weinstock, M. (2010). *The Architecture of Emergence.* West Sussex: John Wiley & Sons.

Wessel, R., Blümel, I., & Reinhard, K. (2009). A 3D Shape Benchmark for Retrieval and Automatic Classification of Architectural Data. *Eurographics Workshop on 3D Object Retrieval*.

Wolfram, S. (1984). Cellular Automata as Models of Complexity. *Nature, 311*(5985), 419–424.

Wolfram, S. (1994). *Cellular Automata and Complexity: Collected Papers.* Westview Press.

Woodburry, R. F. (1990). Searching for Designs : Paradigm and Progress. *School of Architecture*, Paper 62.

Woodburry, R. F., & Datta, S. (2004). Feature Nodes: An Interaction Construct For Sharing Initiative in Design Exploration. In J. S. Gero, *Design Computing and Cognition '04* (pp. 23-36). Dordrecht, Netherlands: Kluwer Academic Publishers.

Woodbury, R., Datta, S., & Burrow, A. (2000). Erasure in Design space Exploration. (J. S. Gero, Ed.) *Artificial Intelligence in Design*, 521-543.

Wright, S. (1932). The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. *Proceedings of the VI International Congress of Genetics*, (pp. 356-366).

Wu, H.-C., Sun, C.-T., & Lee, S.-S. (2004). Visualization of Evolutionary Computation Processes From a Population Perspective. *Intelligent Data Analysis , 8*, 543-561.

Zaman, L., Ofria, C., & Lenski, E. R. (2012). Finger-Painting Fitness Landscapes: An Interactive Tool for Exploring Complex Evolutionary Dynamics. *Artificial Life 13*, 499-505.

APPENDIX A

THE ITEM LIST FOR SODDU'S CHAIR BREEDING EXPERIMENT

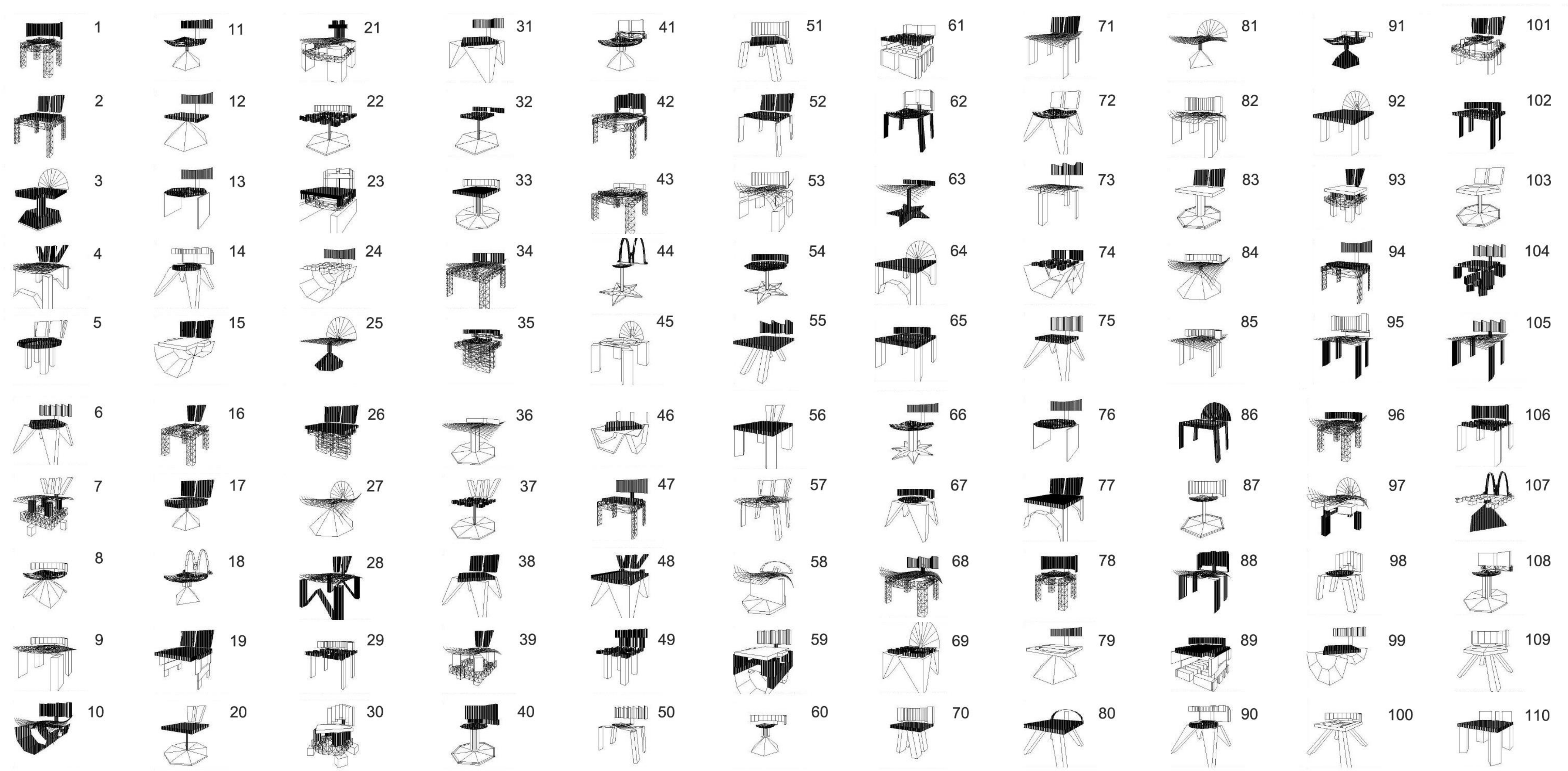

Figure 47 The Item List for Soddu's Chair Experiment

APPENDIX B

RESEMBLANCE CLUSTERS OF DESIGNER A

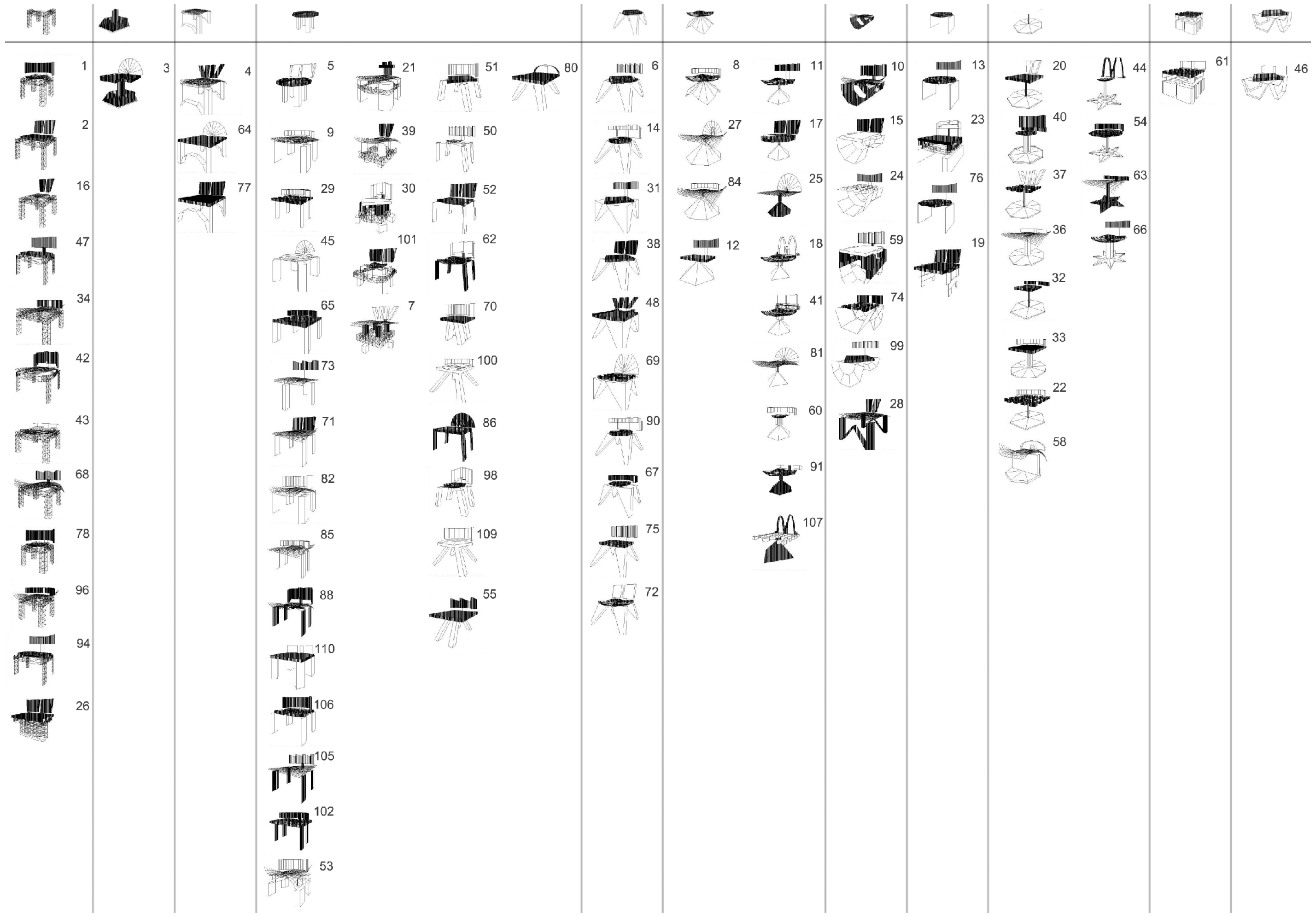

Figure 48 Resemblance Clusters of Designer A

APPENDIX C

VISUAL STRUCTURE OF DESIGNER A

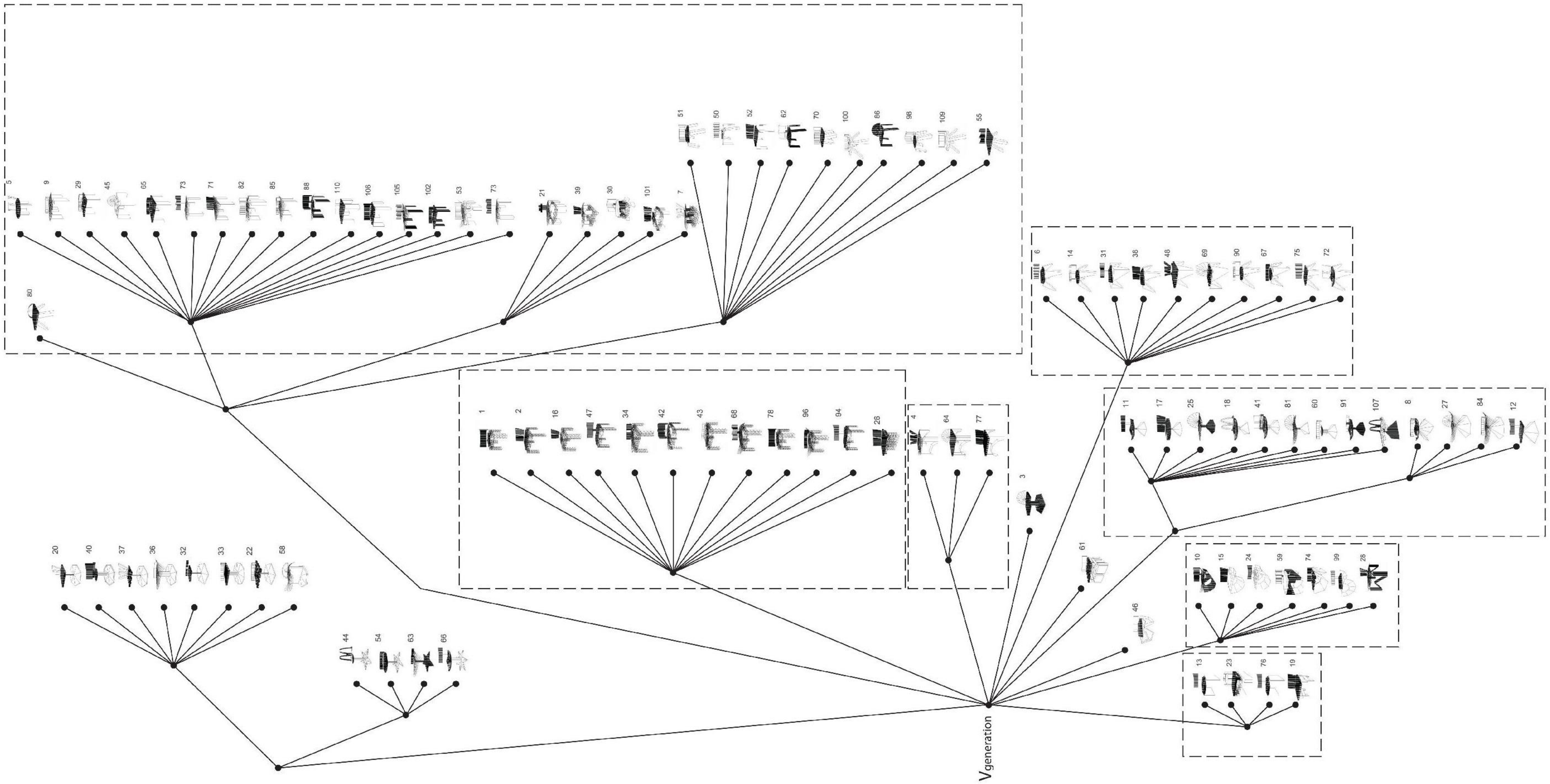

Figure 49 **Visual Structure of Designer A**

APPENDIX D

RESEMBLANCE CLUSTERS OF DESIGNER B

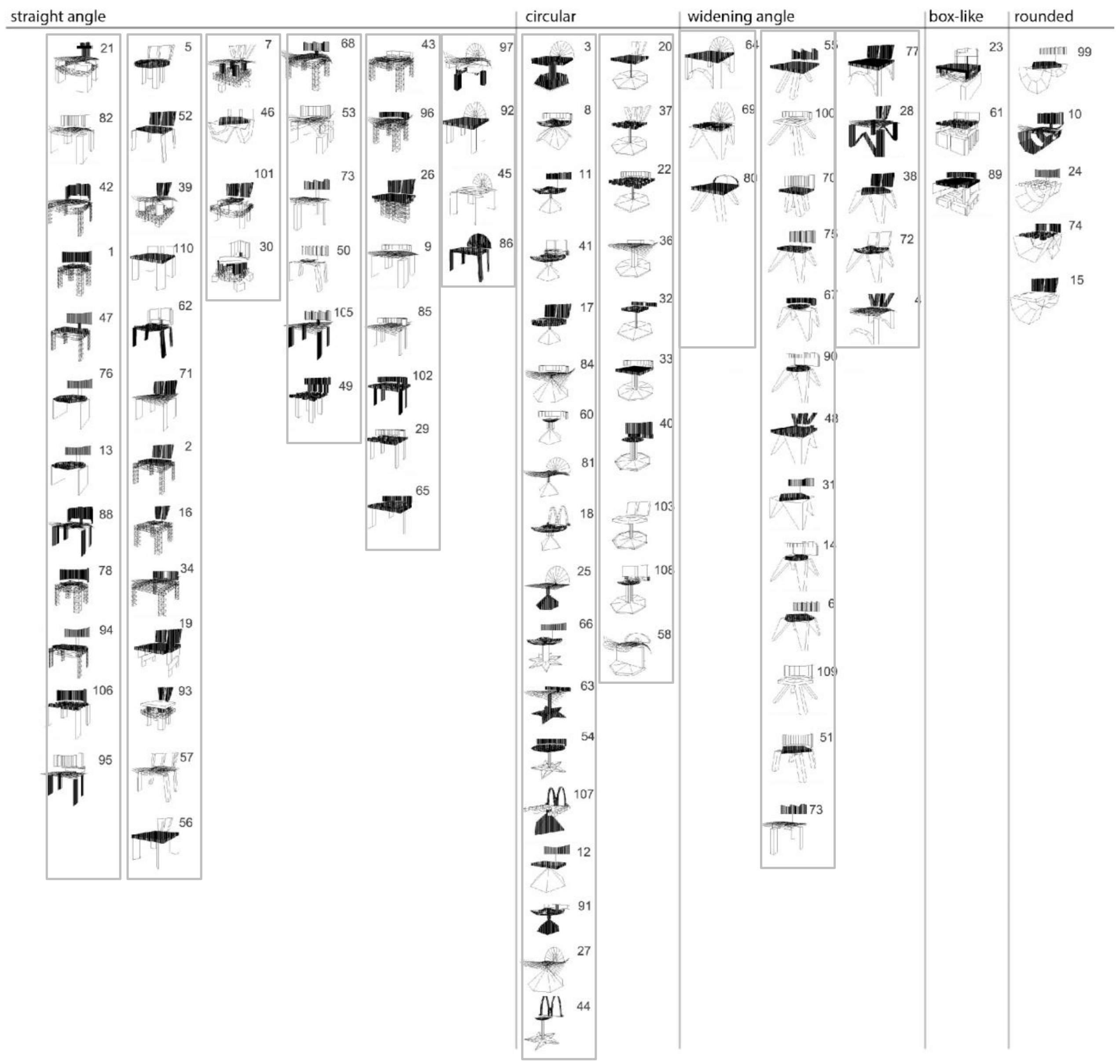

Figure 50 Resemblance Clusters of Designer B

APPENDIX E

VISUAL STRUCTURE OF DESIGNER B

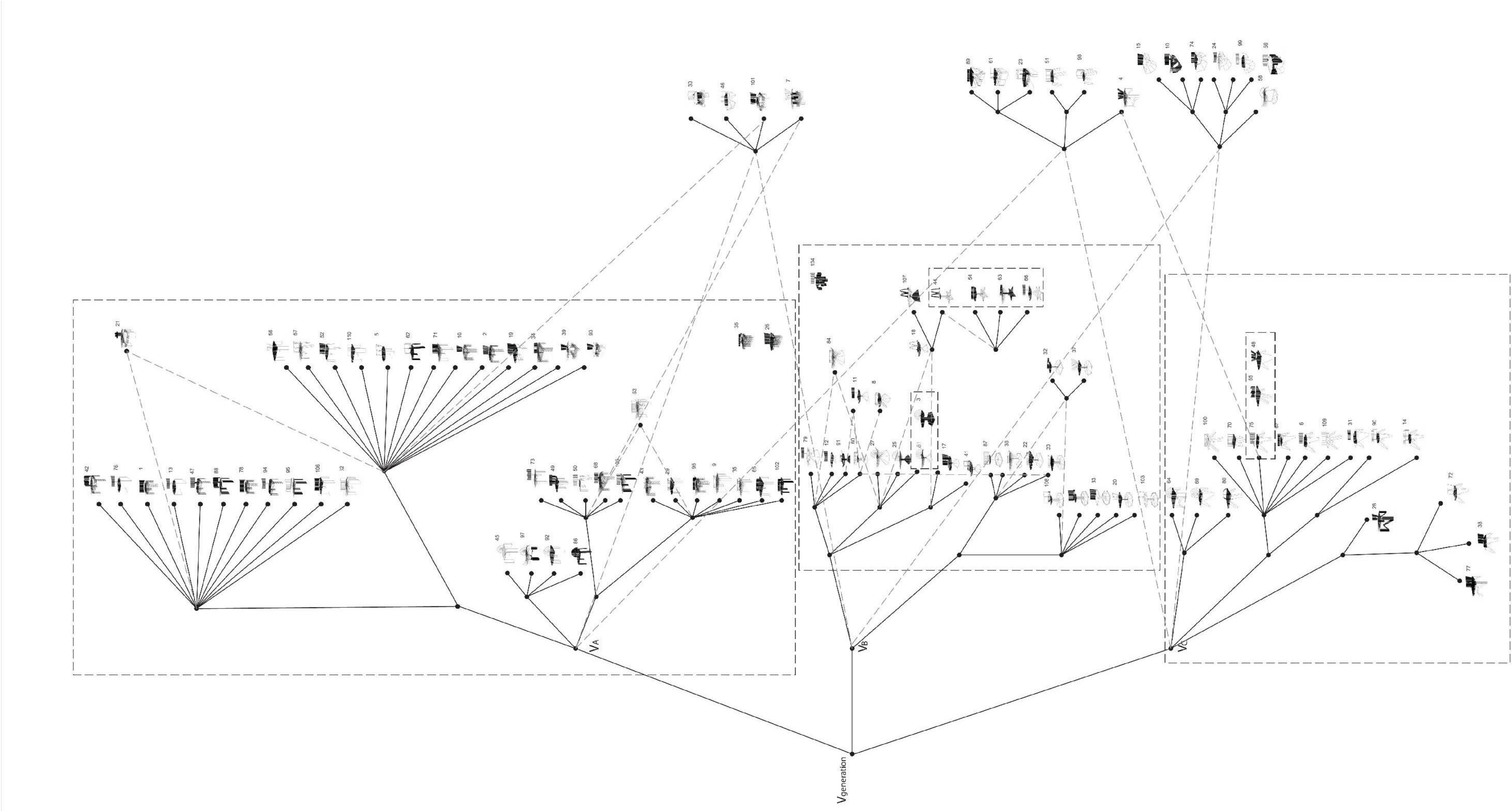

Figure 51 Visual Structure of Designer B

APPENDIX F

RESEMBLANCE CLUSTERS OF DESIGNER C

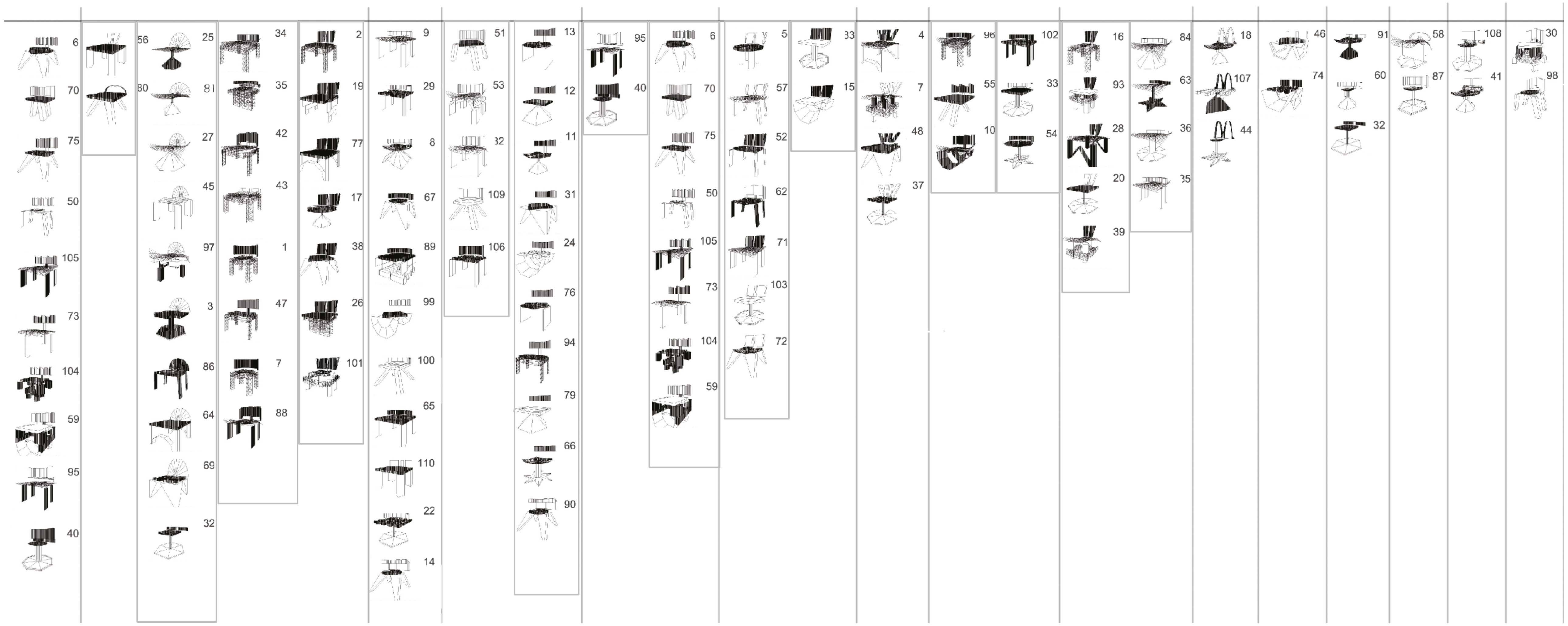

Figure 52 Resemblance Clusters of Designer C

APPENDIX G

VISUAL STRUCTURE OF DESIGNER C

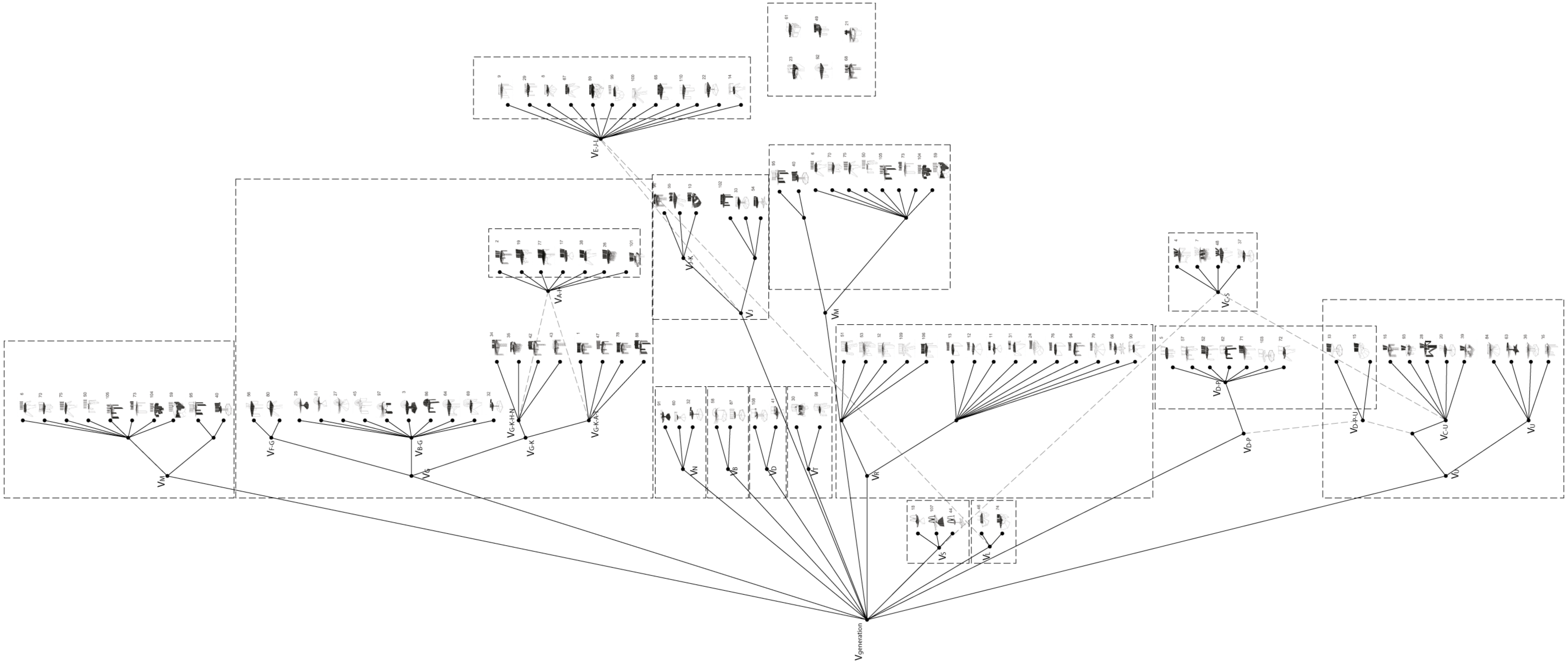

Figure 53 Visual Structure of Designer C