COMPLEXITY MANAGEMENT AND MUTABILITY IN ARCHITECTURAL FORM CONCEPTION: FORM-BLINDNESS AND SOFTFORM

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

BY

EGEMEN BERKER KIZILCAN

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

AUGUST 2015

Approval of the Thesis:

COMPLEXITY MANAGEMENT AND MUTABILITY IN ARCHITECTURAL FORM CONCEPTION: FORM-BLINDNESS AND SOFTFORM

submitted by **EGEMEN BERKER KIZILCAN** in partial fulfillment of the requirements for the degree of **Master of Architecture in Architecture Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. T. Elvan Altan ____________________
Head of Department, **Architecture**

Prof. Dr. Zeynep Mennan ____________________
Supervisor, **Department of Architecture, METU**

Examining Committee Members:

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

Prof. Dr. Zeynep Mennan ____________________
Department of Architecture, METU

Assoc. Prof. Dr. Berin Gür ____________________
Department of Architecture, TEDU

Assoc. Prof. Dr. Fehmi Doğan ____________________
Department of Architecture, IZTECH

Assist. Prof. Dr. Başak Uçar Kırmızıgül ____________________
Department of Architecture, TEDU

Date: August 31, 2015

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: Egemen Berker Kızılcan

Signature:

ABSTRACT

COMPLEXITY MANAGEMENT AND MUTABILITY IN ARCHITECTURAL FORM CONCEPTION: FORM-BLINDNESS AND SOFTFORM

Kızılcan, Egemen Berker
M.Arch. Department of Architecture
Supervisor: Prof. Dr. Zeynep Mennan

August 2015, 119 pages

Expressions in architecture, as modes of conveying information, have the power to shape perception and thinking, as well as architectural production. Architectural conception is today induced by custom information models with the advent of technologies and methodologies associated with the computational paradigm and therefore are capable of inhabiting complex orders that arise from a multiplicity of concerns. Architecture is no longer constituted by the fixity of a single instance of form, instead possibility spaces which represent the mutability of form and which extend the solution set are being privileged for exploring the qualitative aspects or testing outcomes of design approaches. Architecture's relation with aspects of information, complexity, multiplicity and changeability in design conceptualization requires expressions that are efficient in providing for such aspects, besides establishing the dialogue between conceptualization and information. The thesis provides a discussion of the multi-faceted issues regarding the expression of design intention and thinking in the context of computational design, exploring dualities between visual and relational reasoning as well as between conceptual and perceptual structures of expression. The study presents challenges in terms of the computational architectural design paradigm regarding the management of complexity and form mutability, and to this end, develops two concepts/approaches/qualities, form-blindness and softform, that are exhibited in current architectural expressions and thinking. Form-blindness denotes a controlled level of specificity about form and the postponement of its fixation in a certain implementation, while ensuring complexity management and multiplicity. Softform, on the other hand, infers encapsulations of mutable forms that can incarnate multiple instances, to enable exploration and multiplicity in design.

Keywords: expression, possibility space, form-blindness, softness/hardness, complexity management, mutability

ÖZ

MİMARİ FORM YARATIMINDA KARMAŞIKLIK YÖNETİMİ VE DEĞİŞEBİLİRLİK: FORM-KÖRLÜĞÜ VE YUMUŞAK FORM

Kızılcan, Egemen Berker
Yüksek Lisans, Mimarlık Bölümü
Tez Yöneticisi: Prof. Dr. Zeynep Mennan

Ağustos 2015, 119 sayfa

Bilgiyi iletme yöntemi olarak anlatım, mimarlıkta algıyı ve düşünceyi, dolayısıyla üretimi biçimlendirme gücüne sahiptir. Mimarlığın bugünkü kurgusu hesaplama paradigmasıyla bağdaştırılan teknoloji ve metodolojilerin ilerlemesiyle, çoğul düşüncelerin beraber işlenmesinden ortaya çıkan karmaşık düzenli özel bilgi modelleriyle yürütülmektedir. Ayrıca mimarlıkta artık tekil bir sonuç ürünün sabitliğindense, formun nitel olanaklarını incelemek için veya bir tasarım niyetinin sonuçlarını test etmek için formun dönüşebilirliğini gözeten ve çözüm kümesini genişleten olasılık alanları öne çıkmaktadır. Mimarlığın bilgi, karmaşıklık, çoğulluk ve değişebilirlik durumlarıyla olan ilişkisi, bu durumları kavramlaştırma ve bilginin diyaloğunu kurmasının yanı sıra tasarımın kavramlaştırılmasına bu durumları sağlayan anlatımları gerektirmektedir. Bu tez, hesaplamalı mimari tasarım bağlamında, tasarım niyet(ler)inin anlatımını ve düşüncesini ilgilendiren, görsel ve ilişkisel nedenleme, ve algısal ve kavramsal yapıların anlatımları ikililiklerine dair konuları bir materyel pratik olan mimarlığın hesaplamalı tasarımdan daha çok görselliğe ve algısallığa dayandığı önsavını göz önünde bulundurarak tartışmaktadır. Mimarlığın güncel durumunda karmaşıklığın yönetilebilirliği, formu değişebilir yapmak, algısal ve kavramsal yapılar arasında ilişkilendirilebilirlik, ve görsel ve ilişkisel anlatım yöntemleri açısından çeşitli sorun ve sorunsallar sunan bu tartışmayı yürütürken, çalışma, güncel mimari tasarım paradigmasına ait anlatım ve düşünceyle bağdaşan ve tezde form-körlüğü ve yumuşak form olarak adlandırılan iki kavram/yaklaşım/nitelik geliştirmektedir. Form-körlüğü, tasarımda karmaşıklığın kontrolünü ve çoğulluğu sağlamak amacıyla formdaki özgüllüğün derecesinin kısıtlanması ve belli uygulamalara erken bir sabitlenmeyi ertelemek anlamına gelmektedir. Yumuşak form ise keşif veya çoğulluğu sağlamak için çoğul miktarda örnek cisimleştirebilecek değişebilir formların kapsanmasıdır.

Anahtar Kelimeler: ifade, olasılık alanı, form-körlüğü, yumuşaklık/sertlik, karmaşıklık yönetimi, değişebilirlik

To My Family and

To All Else Whom I Consider Dear

ACKNOWLEDGEMENTS

There is an immense amount of effort that goes into a work like this, not only mine but also of many others.

I owe my deepest gratitude to my supervisor, my mentor Prof. Dr. Zeynep Mennan for her infinite patience and meticulousness, in addition to her encouragements, suggestions and endless support. She has patiently encouraged me to achieve the better and the advanced throughout all of the stages of this study. I feel very lucky that I am guided by her knowledge and wisdom in my academic life.

I would like to thank my jury members Assoc. Prof. Dr. Berin Gür, Assoc. Prof. Dr. Fehmi Doğan, Assist. Prof. Dr. Başak Uçar Kırmızıgül, and Assist. Prof. Dr. İpek Gürsel Dino for their invaluable and sincere contributions.

I would like to thank my family, especially my mother Mine, my father Fatih, and my brother Engin, for their endless love, care, and support, not only throughout this study, but in life overall. Without their support, I feel I would not be able to come this far in life, and for that I think this is the best place to express my gratitude to them. I am more than blessed to have them in my life.

Thank you to Gizem Akdağ, for all her love and support. I feel very lucky to have you in my life.

Thank you, my dearest friend, my brother, Emrehan Berkay Çelebi. We have shared countless moments throughout our long friendship, many of them being odd and strange. As you, I also feel that there is more to come for us to face together.

Lastly, but definitely not the least I would like to thank all my friends, my colleagues, especially Seray Türkay, Hayri Dörtdivanlıoğlu, and Duygu Tüntaş, but also Ulaş Başar Özbilen, Yiğit Acar, Leyla Etyemez, Özgün Özçakır, Burcu Köken, Neris Parlak, Ensar Temizel, Aslıhan Günhan, Serdar Erişen, and Emine Çiğdem Asrav for their invaluable friendship and support.

I owe deep gratitude to whom I have mentioned their names here, together with many others whom I have not, for supporting me and for making me who I am. They are my greatest treasure in life.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

> The Modernist techniques of erasure and homogenisation no longer seem appropriate as a way of achieving integration, nor is the identification of historical, regional and linguistic types or figures of any use in achieving differentiation ‑ precisely because of their dependence on codes and systems of representation. We try to develop techniques that are capable of operating outside existing codes, to exploit the potential of a foreign operativity, to operate by migration, displacement, estrangement, not by seeking out origins or essences, developing genealogies, defining boundaries, assigning capacities or inventing languages.[1]

Architecture is a material practice; a practice which achieves architectural forms from what is immaterial as information, ideas and knowledge.[2] During the process from immateriality to functional material structures, or in different terms from design intentions to architectural forms, conceptualizations of design depend on a multiplicity of aspects in inducing, enduing, affecting, or shaping design solutions due to the indeterminate and ambiguous nature of design.[3] Consequently, this process from idea to form that is based on the dialogue of thinking and expression

---

[1] Foreign Office Architects. "Yokohama International Port Terminal", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p58-61.

[2] Pallasmaa, Juhani. "Empathic Imagination, Formal and Experiential Projection." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014. p80-85.

[3] Kostas Terzidis mentions about the multiplicity in design conceptualizations through a discussion comparing determinate formalized systems of computation and indeterminate nature of design. See: Terzidis, Kostas. "Algorithmic Complexity: Out of Nowhere." Complexity, Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008. p75-86.

exhibits a multiplicity of relationships between information and the conceptualization of design.

These relationships are especially diverse in the scope of computation where design is influenced by an extensive frame of information and methodologies, regarding the domains of mathematics[4], computation[5] and realization of form[6]. Contemporary problems require designs arising from a multiplicity of concerns and considerations.[7] Through new techniques and tools that are associated with the computational design paradigm, an increased amount of information endues design process and results. By virtue of new tools and methodologies that are efficient in managing complexity, architectural form is produced by complex orders that incorporate multi-layered concerns and intentions.[8] Consequently, computational design methodologies provide for custom relationships between information and conceptualizations by custom and complex information models.

In the context of computation in architecture, Marcos Novak had already described a shift of focus in form conception from the single finite object to a variable object presenting a range of outcomes.[9] In this sense, conception of architectural form is no longer concerned with processes of one-time linear resolution where knowledge about form gradually increases; but instead concerned with ranges in which form is

---

[4] Legendre, George, ed. Mathematics of Space. London: Wiley, 2011.

[5] Peters, Brady, and Xavier De Kestelier, eds. Computation Works: The Building of Algorithmic Thought. London: Wiley, 2013.

[6] Hensel, Michael, Achim Menges, and Micheal Weinstock, eds. Techniques and Technologies in Morphogenetic Design. London: Wiley-Academy, 2006.

[7] Hight, Christopher and Perry, Chris “Introduction to Collective Intelligence in Design”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p189-200.

[8] Mennan, Zeynep. “Mind the Gap: Reconciling Formalism and Intuitionism in Computational Design Research." Eds. Henriette Bier, and Terry Knight. Footprint, Delft Architecture Theory Journal, Vol.15. 2014: p33-42.

[9] Novak, Marcos J. "Computational Composition in Architecture." Computing in Design Education, ACADIA '88 Workshop Proceedings. Ed. P. J. Bancroft. Ann Arbor: University of Michigan, 1988. p5-30.

developed and explored.[10] This paradigm highlights the necessity for multiplicity and dynamism in form, therefore flexibility and changeability in form have become privileged properties in the last decades.

From this mentioned frame of design influenced by computational methodologies, one can infer that design conceptualization is increasingly induced by multiplicity, mutability and complexity in the contemporary architectural design. This thesis problematizes this state of design conceptualizations and their relationship with architectural form through expression and thinking in design. Namely, the thesis aims to provide an outline for conceiving possibility spaces for architectural form in this context of multiplicity, mutability and complexity by investigating modes of expression and thinking in architecture.

Although the thesis scrutinizes contemporary computational design theory, its scope extends to design theory in general. Establishing the connection between idea and form requires a special elaboration in this case of design and the diversity that is brought up by computational methodologies provides for important points of discussion regarding this topic. Therefore, dualities between visual and relational reasoning, conceptual and perceptual structures, formal and intuitive approaches, besides visual and mathematical/textual modes of expression pose new challenges to design and fall into the scope of discussion. In general, this thesis aims to provide a contemporary state of discourse on these topics and ultimately to develop two notions (that will be explained briefly later in this chapter) which are related with complexity management and providing multiplicity and mutability, as these issues stand at the very center of architectural form conception in contemporary design.

Both of the notions that are developed throughout the thesis emphasize multiplicity in design. The essential question here is why multiplicity is so important in contemporary design. There may be a number of answers to this question depending on the designer, but in the context of this study, multiplicity is understood as an

[10] Ibid.

aspect which allows for exploration in form, hence for better optimized, integrated and elaborated designs both through methodological and phenomenological means. In this sense, the two notions, form-blindness and softform, provide for multiplicity not just for the sake of multiplicity in design, but for meaningful multiplicity as they infer elaboration in defining the relationship between idea and form. Here, the formal methods of computational design provide for an interesting point of discussion.

Formal techniques that are associated with the computational design paradigm allow for changeability in form and implementation of complex orders through formalized and systematized methodologies due to the mechanical nature of the computer. Kostas Terzidis notes that:

> Both architects and engineers argue for the deployment of computational strategies for addressing, resolving, and satisfying complicated design requirements. These strategies result from a logic based on the premise that systematic, methodical, and traceable patterns of thought are capable of resolving almost any design problem. While this assumption may be true for well-defined problems, most design problems are not clearly defined. In fact, the notion of design as an abstract, ambiguous, indefinite, and unpredictable intellectual phenomenon is quite attuned to the very nature of - or perhaps lack of - a single definition of design.[11]

In the context of computational design, Terzidis compares differences in providing solutions to well-defined and ill-defined problems, and argues that in addition to formalized determinate traceable processes, design is also influenced by the indeterminate, the unpredictable, the ambiguous and the abstract.[12] He illustrates a different paradigm of form production through formations and transformations via formal processes in the context of computational design, together with the notion of expressiveness and meaning in form in the context of computational design that

[11] Terzidis 2008, op. cit., p86.

[12] Terzidis 2008, op. cit., p75-86.

is regulated by formal processes.[13] Similarly, Zeynep Mennan argues for the reconciliation between the formal and intuitive realms of design for architectural solutions that exhibit phenomenological qualities, which otherwise are abstract and lack meaning and interpretation.[14] Design within the computational paradigm therefore exhibits diversity in both methodology and outcomes.

Additionally in the context of computational design, and when concerned with the relationship of information and conceptualization, Pablo Lorenzo-Eiroa mentions the dualities between visual and relational logic besides the conceptual and perceptual structures of architectural design.[15] He highlights the importance of the dialogue between information and conceptualization of design through representational structures and states that interfaces are "spaces for differentiation" when considering visual and relational logic that are relevant to architecture with simultaneously conceptual and perceptual structures of design.[16] Hence, when considering the contemporary state of architectural design, computational design presents diversities in both design thinking and expression by the dualities of formal and intuitional, visual and relational, conceptual and perceptual.[17]

---

[13] Terzidis, Kostas. Expressive Form: A Conceptual Approach to Computational Design. London: Spon, 2003. p67.

[14] Mennan 2014, op. cit.

[15] Lorenzo-Eiroa, Pablo. "Form:in:form, On the Relationship Between Digital Signifiers and Formal Autonomy." Architecture in Formation: On the Nature of Information in Digital Architecture. Ed. Pablo Lorenzo-Eiroa and Aaron Sprecher. London: Taylor & Francis, 2013. P10-21.

[16] Ibid. p18.

[17] Mennan, Zeynep. "Non-standardization Through Non-visualization: Scripting the Dom-ino House" Game Set and Match II; On Computer Games, Advanced Geometries, and Digital Technologies. Eds. Kas Oosterhuis and Lukas Feireiss. Episode Publishers. 2006. P234-241; Mennan, Zeynep. "Des Formes Non Standard : Un 'Gestalt Switch'." ("Of non standard forms: A 'Gestalt Switch'") Architectures Non Standard. Eds. Migayrou, Fréderic and Zeynep Mennan. Paris: Editions du Centre Pompidou. 2003. P34-41; Lorenzo-Eiroa, op. cit.

Following Lorenzo-Eiroa's discussion on interfaces with information and their power in shaping design solutions[18], it can be stated that architectural form is endued by the dialogue between information and conceptualization of design that is established by expressions of design intentions. Representation of architectural ideas, or expressions provide for a widespread discussion as there is a diversity in representing information in architecture, especially in visual form as architecture is deeply related with perceptual structures.[19] Besides the orthographic set[20] which is in direct correspondence with perceptual structures or architectural form, there are modes which instead represent conceptual structures such as conceptual diagrams[21] or ideograms that are produced by Leon van Schaik, Le Corbusier or Louis Kahn[22]. These modes exhibit a different relationship between form and design in the sense that they define ideas that lead to multiple designs. Furthermore, there are custom modes of visualizing information by the virtue of tools and techniques that are associated with computation and that define custom languages through which ideas and information are expressed.[23]

When concerned with such diversity in the modes of representing information, or expressing design intention, the contemporary field of design exhibits a tension between mathematical/textual and visual expressions since architecture is essentially a physical practice which primarily relies on visual aspects and

[18] Lorenzo-Eiroa, op. cit.

[19] Cook, Peter. Drawing, the Motive Force of Architecture. Hoboken: Wiley, 2008.

[20] Türkay, Seray. "Introduction" The Orthographic Set: Making Architecture Visible. M.Arch. Thesis METU, Ankara, 2011. P1-10. Retrieved from: http://etd.lib.metu.edu.tr/upload/12613652/index.pdf

[21] Dogan, Fehmi and Nersessian, Nancy. "Conceptual diagrams: representing ideas in design", Diagrams 2002. Eds. M. Hegarty, B. Meyer & N. H. Narayanan. Berlin: Springer, 2002. P353-355.

[22] Throughout this study ideograms produced by Leon van Schaik are presented as primary examples to the mode. Van Schaik's ideograms are prominent as they strictly represent conceptual structures instead of the perceptual. See: Van Schaik, Leon. Ideograms. Melbourne: Lyon Housemuseum, 2013.

[23] Lorenzo-Eiroa, op. cit.

perceptual structures in the production of form.[24] The mathematical/textual counterpart to visual expressions (such as the code or algorithms) is promoted by both computational methodologies and the increasing complexity of architectural problems as computational techniques are more efficient in calculating this complexity.[25] The visual and mathematical/textual duality poses a tension between what is visual and what is syntactic in the sense that each mode of expression has a different correspondence to perceptual and conceptual structures of design.[26] In this regard, Mennan mentions a new interpretative regime in the code which replaces and displaces perceptual qualities with the unfamiliarity of mathematical/textual expressions.[27] The code as a mathematical/textual expression is interested in the relational, the descriptive, the definitive, the formal, the generic, the sequential, the operational, and the procedural at the expense of the visual, and the perceptual. Thus, the code as an expression provides different descriptions for form in contrast to a visual depiction.[28]

This tension between the visual and the mathematical/textual provides for an important discussion in terms of thinking and expression in architecture where the code has become one of the most operational and expressive media in producing form, or transforming design intention into form in the digital substrate. Representing ideas in the formalized and numerical logic presents discussions on topics such as visual logic and relational logic, besides problematics in displacing perceptual structures with the relational.[29]

Expressions in architecture, whether mathematical/textual or visual, describe the relationship between idea and form, the intention and its implementation. They

---

[24] Mennan 2006, op. cit.

[25] Mennan 2014, op. cit.

[26] Mennan 2006, op. cit.

[27] Mennan 2006, op. cit.

[28] Mennan 2014, op. cit.

[29] Lorenzo-Eiroa, op. cit.

define the correspondence between conceptual and perceptual structures of architectural design. Either more intuitive or more formal, visual or mathematical/textual, they describe a range of possibilities for architectural form and conditions which produce architectural form. Expressional qualities are diverse in referring to a concurrent state of information and knowledge about form and design intention. Above all, expressions lay foundations to conceive, comprehend and convey design intention. They are agents in establishing the dialogue between information and design intention, where design intention is dynamic, evolving through this dialogue.[30] They are ways to manage and increase complexity in design, where multi-layered concerns are possible to implement.

Each expression has a certain degree of knowledge about architectural form and design intention while representing different stages and structures of design. Expressions may represent conceptual structures in more abstract descriptions as in the case of Leon van Schaik's ideograms[31] or conceptual diagrams; or they may represent perceptual structures and have a direct relation with the built form as exhibited in the case of orthographic drawings. Each expression has therefore a certain degree of specificity about form and a different capability of expressiveness and abstraction. Specifically, abstraction as a method to reduce inessential specificity about information holds an important location in this discussion for the sake of complexity management and multiplicity in form. A tension in the modes of abstraction between visual and mathematical/textual modes of expressions is presented as a departure point throughout this study, as abstraction in these modes both similar and differ drastically. Moreover, they may be more observational such as the ideograms, as they may be more operational such as the code. Therefore, there is a diversity in purpose, expressional quality and degree of expressiveness.

In this frame of multiplicities and complexity in both methodology and outcomes in design, the dialogue between information and conceptualization of design

[30] Dogan and Nersessian 2002, op. cit.

[31] Van Schaik 2013, op. cit.

becomes problematic. Existence of custom and complex information models in the production of form infers the necessity for new languages that are efficient in describing and enduing such information models. This thesis problematizes descriptions of design intentions and representations of information, or in more general terms, expressions in architecture that manifest immaterial ideas into material structures.

The aim of this thesis is not to explain how to express design intentions as that is highly subjective and problem-dependent; thus it is impossible to provide a generalized framework for such issues. Instead, this thesis centers on the two qualities of expressions which are related with complexity management and form multiplicity, namely form-blindness and softform. Form-blindness as an approach or an expressional quality denotes a controlled level of specificity about form or postponing fixations to certain implementations of ideas in order to ensure complexity management and multiplicity. Softform, on the other hand, infers a virtualization of design intention into a mutable single entity or an encapsulation of a mutable form instance that can incarnate multiple instances, for the sake of exploration and multiplicity in design. These expressional qualities are reflected in terms of both knowledge about the concurrent stage of design and its ability to change and adapt. This thesis provides a discussion on qualities of expressions that are relevant in describing complex design intentions and which aid design exploration and development by introducing multiplicities.

The study explores these qualities through expressions structured in both visual and mathematical/textual form in order to establish a basis for discussion in the contemporary scope of computational design in scrutiny of three modes of expression: the code, the diagram and the ideogram. The tension between visual and mathematical/textual structures provides for a valuable discussion in terms of expressional ability in providing multiplicities and managing complexity. Correspondence and reliance to conceptual and perceptual structures in expressions are also investigated as these qualities challenge architecture through the introduction of relational logic and bring up the question of the dependency of architecture on image and visual form.

Both of the offered qualities argue for a liberation of architectural form from fixation into a single outcome in order to ensure the manageability of the complexity of multi-layered information and to increase correspondence to conceptual structures for better integrated, conceived and conceptualized designs. The thesis is therefore concerned with the questions of how to introduce changeability to architectural form which is produced by complex orders; how to achieve multiplicity both in form and organization; how to represent and work with ranges of form that arise from complex orders; and finally, how to provide such ranges for form which allow for exploration in which better architectural solutions are searched and found.

Through illustrations, examples and analysis in the proceeding chapters, this study scrutinizes visual and relational models of architectural design, the correspondence of conceptual and perceptual structures in design, abstraction and specificity in both form and expression, changeability and flexibility of architectural form, encapsulating and widening possibility spaces for form in the frame of expressional quality in computational design.

CHAPTER 2

THE DIALOGUE BETWEEN CONCEPTUALIZATION AND INFORMATION IN CONTEMPORARY ARCHITERCTURAL DESIGN

Architecture has a renewed relationship with information and complexity by virtue of methodologies and approaches that are associated with the computational paradigm. New and custom modes of utilizing, interacting with, interpreting and visualizing information with increased complexity are exhibited in conceptualizing design, thus producing form. Through formal techniques which provide for multiplicity and variability in form, conceiving architecture in the contemporary scope of computational design is based on designing for possibilities and variabilities that arise from complex orders of multi-layered information and intentions.[32] While computational methodologies emphasize formalism in producing form, it is argued that this process should also be involved with intuitive and subjective aspects of design, as otherwise would result with designs that lack phenomenological qualities such as meaning and interpretation.[33]

Considering multiplicity of approaches in design that are fed by subjectivity and intuition besides the existence of custom information models in producing architectural form, this situation outlines a new relationship between information and conceptualization of design. In this context, representational media in architecture, or representations of information have the power to shape conceptualizations about design as they present interfaces with information where

[32] Novak, op. cit.

[33] Mennan 2014, op. cit.

thinking and expression are established in architecture.[34] In contrast, representation of complex and custom information models is problematic in the sense that traditional modes of representation in architecture exhibit inadequacy in representing custom and complex information. Design intentions which aim to provide multiplicity, diversity and complexity to both design process and result, mandate custom modes of expression that are efficient in representing custom information models.

2.1. Conceiving/Conceptualizing Architecture in the Age of Information

> Considering the exponential capabilities offered by the information technologies, architecture has been engaged into redefining its modes of production and the nature of its expression. [···] the architectural object increasingly resembles an organism that is responsive to its own internal nature and the external conditions of its surrounding. In this hyper-mediated environment, what used to be, the collective gives way to the connective, the rigid structure to the open system, the condition of causality to non-linearity. Such an environment is generated by a wide range of information influences that render a reality in constant mutation; a reality shaped by potentialities, instabilities, and probabilities.[35]

The impacts of the Information Age are prominent, not only in the field of architecture, but in all fields including daily life. Utilization of information has become vast and diverse through new technologies and fields: It is possible to gather, access, use, and communicate information from a variety of devices, such as computers, tablets, smartphones or even smart watches independent from time and place. Information invades both social and personal levels through social

[34] Lorenzo-Eiroa, op. cit.

[35] Sprecher, Aaron. "Architecture in Formation: On the Affluence, Influence, and Confluence of Information." Architecture in Formation: On the Nature of Information in Digital Architecture. Ed. Pablo Lorenzo-Eiroa and Aaron Sprecher. London: Taylor & Francis, 2013. P25.

platforms such as Facebook or Twitter. There are new notions such as Big Data[36], and there are new fields such as data mining[37] that are based on information.

These developments which involve custom and specialized modes in the visualization of complex information [Figure 2.1] epitomize the new ways of obtaining, using and visualizing information in the context of computational methodologies. The diversity related with the gathering and dissemination of information emphasizes its relationship with complexity. Computation of complex information is now agreed to be far more elaborate than the simple linear input-process-output chain of calculation, as computation allows for complex procedures in dealing with complex information.

---

[36] Big data is a term which is used to explain data sets that are large and complex where traditional data processing applications fall inadequate.

Snijders, Christian, Uwe Matzat, and Ulf-Dietrich Reips. "“Big Data”: Big Gaps of Knowledge in the Field of Internet Science." International Journal of Internet Science, 2012. Web. 3 Sept. 2015. <http://www.ijis.net/ijis7_1/ijis7_1_editorial.pdf>.

[37] Data mining is concerned with discovering patterns in large data sets like big data, involved with methods at the intersection of artificial intelligence, machine learning, statistics and database systems.

"Data Mining Curriculum: A Proposal (Version 1.0)." *Data Mining Curriculum: A Proposal (Version 1.0)*. 2006. Web. 3 Sept. 2015. <http://www.kdd.org/sites/default/files/CURMay06.pdf>.

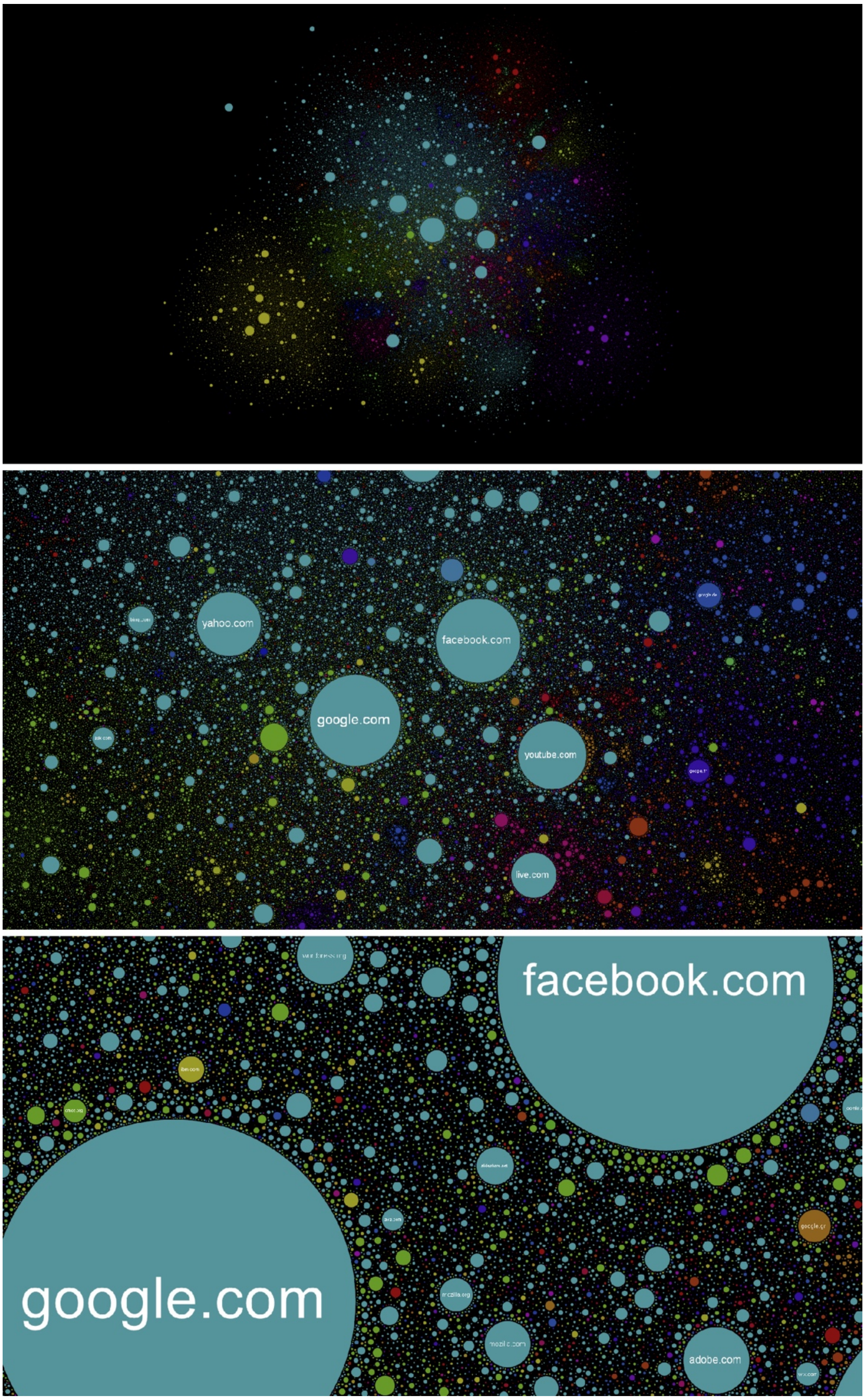


Figure 2.1 The Map of the Internet

Source "The Internet Map”. Web. 7 June 2015. <http://internet-map.net>.

Impacts of this state of information and complexity are manifest in architecture as well, exhibited by the amount of research on complexity and information in the context of architecture. There exist multiple approaches towards complexity and the understanding of complexity in architecture. Research on complexity spans the extents of the contextual, the behavioral, the social and the technical.[38] Similarly, there are multiple positions that discuss which, why and how information is relevant to architecture. There are diverse opinions about structuring, interacting with, interpreting, representing and utilizing information in architectural design.[39]

This condition of architectural design related with complexity and information is enabled by advances in the field of computation. There is a new way of conceiving architecture, one which has been initiated with the adoption of computational techniques and approaches associated with these techniques into the field of architecture. Through new techniques that are involved with computation, it is now possible to deal with greater complexity; a complexity which had been absent from or ignored in the theory and praxis of the last century.[40] In the context of the *Digital*

---

[38] There is a volume titled *Complexity, Design Strategy and World View* which illustrates the notion of complexity in architecture in the context of digitalization. The volume comprises of essays by many authors, whom illustrate different notions about complexity such as Robert Venturi (pages 13-24) and Denise Scott Brown (pages 25-36) argue about the relation of context with complexity; Andrea Gleininger (pages 37-58) discusses "The Difficult Whole"; Georg Vrachliotis (pages59-74) discusses behavioral and operationalization aspects of complexity; Kostas Terzidis (pages75-89) demonstrates algorithmic complexity; Klaus Mainzer (pages89-98) explains complexity in natural and social dynamics.

Gleiniger, Andrea, and Georg Vrachliotis, eds. Complexity Design Strategy and World View. Basel: Birkhäuser, 2008.

[39] There is a recent volume about architecture and information titled *Architecture In Formation*. Composed of six chapters about different aspects of information in architecture, this editorial provides an extensive discussion held by a number of authors. The topics of discussion are: (1) structuring information, (2) information interfaces, (3) responsive information, (4) evolutionary information, (5) extensive information-material information, (6) information affect.

Sprecher, Aaron and Lorenzo-Eiroa, Pablo. eds. Architecture in Formation: On the Nature of Information in Digital Architecture. London: Taylor & Francis, 2013.

[40] Zeynep Mennan argues that the preference for simplicity and simplification in the design of the last century was mainly based on the lack of efficient tools in managing complexity and that computational design has now obtained new methods and tools which remedy for deficiencies in the management of complexity.

Turn in Architecture[41], as Mario Carpo calls it, there is a shift to complexity driven design techniques and approaches. What this digital turn exhibits, argues Carpo, is that design is less concerned with notions that are associated with simplification such as linearity, objectification, specification, singularity, and standardization: It is rather concerned with the more complex, such as non-linearity, multiplicity, pluralism, and non-standardization. Carpo illustrates this changing paradigm in conceiving design through the theory and praxis of the last two decades.[42]

Three decades ago, on similar terms, Marcos Novak lists the causes for this new way of conceiving architecture as: (1) the finite object has been replaced by the variable object, (2) the idea of singularity has been replaced by that of pluralism, (3) diverse aspects of knowledge have been brought into contact through the common representation of equally accessible information.[43] While the first two causes infer the idea of the dynamically specifiable and modifiable object in a range of acceptable possibilities, the third cause states a change in usage of knowledge in design in a multi-layered, specialized and non-hierarchical manner. The three causes together, argues Novak, deduce an architectural object which resides in a spectrum of possibilities and dynamically modified by a multi-layered set of concerns, considerations and intentions while staying in this spectrum:

> These three ideas make it possible to create a liquid architecture, an architecture of relations in which the final built object is wrested from an infinite continuum of possible variations by the assignment of particular values to variables that govern its internal structures. The creation of systems of relationships and the assignment of specific values becomes the foreground of the architect's activity

---

Mennan 2014, op. cit.

[41] Mario Carpo illustrates this paradigm shift regarding design following the digital revolution through a set of articles that are ordered in chronological order. In doing so he outlines the period of this shift and the changing approaches towards design in a range of aspects concerning, for instance, material technologies, architectural form, information in architecture, and scripting.

Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

[42] Ibid.

[43] Novak, op. cit., p14.

> and invention. The new infinitely variable object negates the singularity of the old finite object and asserts freedom of an open system where change is celebrated.[44]

There is a shift from progressive, linear design processes where architectural form is resolved and specified over time into a final product, to non-linear, re-executable, reproducible processes where potentials of design intentions can be explored. By virtue of computational techniques, it is possible to design for a possibility space of form which represents all possible permutations of an abstract logic that produces form[45] rather than its single one-time instance. By defining a relational, information based, multi-layered model, it is possible to automate the design process, re-run it, generate countless permutations of form without tedious manual labour. Computation allows architectural form to be heuristically generated, tested, diversified, and modified.[46]

This modifiable, re-executable, heuristic mode of conceiving design which emphasizes possibility spaces and multi-layered information would, according to Mennan, infer a new relationship with complexity regarding design process and product:

> […] the computational paradigm both creates and sustains complexity. Complexity bears a non-linear relation to information transmission and processing technologies: improved means and methods used in complexity management do not reduce but rather increase the complexity of design problems.[47]

---

[44] Novak, op. cit., p14.

[45] This description of a possibility space for form is developed from Manuel De Landa's discussion on abstract machines and virtual solution spaces, and Marcos Novak's discussion on variable forms and pluralism. See: De Landa, Manuel. "Deleuze, Diagrams and the Genesis of Form". Any Magazine Issue 23, 1998. p30-34; Novak 1988, op. cit.

[46] Terzidis 2003, op. cit., p65-73.

[47] Mennan 2014, op. cit., p33.

With increased capability in managing and instrumentalizing complexity, it has become possible to process more information or take into account more factors in design conception. Computational models make it, not only possible, but also manageable to design for possibility spaces that are governed by a multi-layered context and multiplicity.

In a wider perspective, contextual information about design is only one layer that contributes to multi-layered computational models. As it can be inferred from Novak's third cause for this new way of conceiving architecture, by using these models it becomes possible to merge architectural knowledge with knowledge belonging to different fields. This provides a custom, equally accessible, multi-layered knowledge which is to be used in governing the design process and which is specific to the design problem at hand. Such modes of specialization in knowledge engage a prominent role in contributing to the qualities of complexity and multiplicity, qualities which are essential for widening the possibility space of a design solution.

2.1.1. Specialized Knowledge

Architectural knowledge belongs to multiple fields such as architecture, urban design, structural engineering, mechanical engineering to cite a few. It is shaped by different perspectives and approaches that belong to different professions and fields; for instance, how an urban planner conceives design differs from how an architect does. All of these approaches coexist in the design environment.

Van Schaik argues that the analytic procedures of the scientific revolution of the 17th century have critically undermined architectural thought: Through dissections and taxonomic classifications, wholes were divided into elucidatory and consumable parts which are stripped of their meanings and associations.[48] Hence,

---

[48] Leon van Schaik illustrates the results of standardization in knowledge in terms of creating space and *spatial intelligence.* See: Van Schaik, Leon. "Chapter Six: New Futures for Architects: New Roles for Practitioners." Spatial Intelligence: New Futures for Architecture. Chichester, England: Wiley, 2008. p164-181.

knowledge that is relevant to design has become dissociated and isolated into field based individual spheres. It could be argued that, following the scientific revolution, the categorization of knowledge belonging to different fields have resulted in reduced complexity in the domain of architecture by standardization of and lost associations between knowledge bases which induce architecture.

Currently, the necessary enfolding of these fields for providing better integrated design solutions and broader possibility spaces can be observed. Such cross-disciplinarity or interdisciplinarity is for instance apparent in the works of those who specialize or hybridize their individual knowledge by acquiring knowledge outside their primary field, such as architects who become capable to develop custom software solutions for specific design problems. Similarly, there are trans-disciplinary collaborations between individuals from different fields on a project or practice basis, which create a '*collective intelligence*'[49] with specialized knowledge. Examples to this approach would be the annual research pavilions designed and manufactured by the ITECH/ITKE programme which are collective efforts of individuals from various disciplines such as civil engineering, architecture, biology, biomimetics, robotics, urban planning.[50]

Considering multi-layering of knowledge, multiplicity in approaches and broader possibility spaces together, it can be stated that complexity in conceiving architecture emerges from two expectations; a multiplicity in concerns, conditions, considerations, and intentions and a demand for multiplicity in results. Although computational tools have the capability to sustain this complexity through formal methodologies, this brings forth the issue of complexity management:

---

[49] Hight and Perry discuss 'collective intelligence' in terms of two scales: (1)scale of the design practice, (2) scale of design technology and product. These two scales are integral to each other and are exhibited on the social, economical, political aspects as well as the technical. See: Hight and Perry, op. cit.

[50] Information about the mentioned series of annual pavilions can be found on the programme's webpage: "News « Institute for Computational Design (ICD)." News « Institute for Computational Design (ICD). Web. 8 June 2015. <http://icd.uni-stuttgart.de/>.

> Complexity management is undeniably becoming a major issue in current computational research, sustaining and promoting naturalisation and formalisation as the two main operational forms encountered in the management of this complexity. In computational design research, as in other fields, the realisation of a growing complexity contributes to an extensive use of formal languages and quantitative/computational tools that rely increasingly on the translation of complex structures into a formal, natural idiom.[51]

Following Mennan, there is a refreshed way of working on both conceptual and methodological levels of design, one which favors formal methodologies over subjectivity and intuition. On the one hand, the two expectations in multiplicity of this new way of conceiving architecture are achievable and manageable by using computational techniques. On the other hand, this way of working should not mandate an over-formalization in design. The more subjective and intuitive approaches, that are based on the mental space of the designer, should also be accounted for in order to refer to a broader possibility space that is not stripped of its meaning and other such phenomenological qualities.[52]

[51] Mennan 2014, op. cit., p33.

[52] Mennan 2014, op. cit., p33.

Figure 2.2 Heydar Aliyev Centre by Zaha Hadid Architects

Source "Zaha Hadid Architects Heydar Aliyev Centre". Web. 7 June 2015. <http://www.zaha-hadid.com/architecture/heydar-aliyev-centre/>.

2.2. Mental Space

As discussed on the previous section, there is a new way of conceiving architecture that is shaped by potentialities and multi-layered information. The possibility space for the architectural object relies on the conceptual level and manifests itself on the methodology level. On the methodology level and when concerned specifically with technique, the reductionism of the simplified architectural form, concerning its geometry, organization and manufacturing, has been overcome by the new technologies in the domain of computational design and manufacturing. There are ample amount of study that dwells on topics such as mathematics in architecture[53],

[53] The 212th profile of AD which is titled 'Mathematics of Space' illustrates architectural design in relation with mathematics and computation on a praxis basis. Topics of discussion include geometry and double-curved surfaces by Mark Burry (p80-99), computation and importance of mathematics by Fabian Scheurer and Hanno Stehling (p70-79), mathematics and architecture by Antoine Picon (p28-35). See: Legendre, op. cit.

computation in architecture[54], generation of form and material technologies for realization[55] in relation with architectural form; its geometry, its conception/actualization and its realization in the context of computational design. Mainly, through custom material or immaterial techniques it has become undemanding to aid the generation and actualization of the increasingly complex forms of architecture.[56] It is possible for any form, whether complex in geometry or not, to be resolved, optimized, manufactured and assembled with high precision with the aid of these techniques; such as in the case of Heydar Aliyev Centre by Zaha Hadid Architects [Figure 2.2].

However, such computational design methodology that concentrates solely on technique exhibits a case which favors the formalization paradigm over conceptualization in design, hence leaving spatial, perceptual, thus the overall phenomenological qualities incomplete: “Calculation leaves an incomplete space that cannot be saturated with information alone and waits to be filled with meaning and interpretation”.[57] Outlining the position of algorithms as the centerpiece in computational design methodology when considering the generation of form, Kostas Terzidis explains this situation as:

> While many algorithms have been invented and implemented for architectural design in space allocation and planning problems, their implementations in aesthetics and formal theories has been, generally, limited. Most of the theories related to form pertain mainly to subjective interpretation and perception. In contrast, algorithmic logic involves a deterministic approach to form and its

[54] Another AD profile which is titled ‘Computation Works, The Building of Algorithmic Thought’ dwells locates formal procedures and computation in providing architectural solutions. See: Peters and De Kestelier, op. cit.

[55] Hensel, Menges and Weinstock, op. cit.

[56] A contemporary discussion on materiality and materialization in architectural form and manufacturing with computational tools is presented in a conference paper. See: Kızılcan, Egemen Berker. "Non-standard Materialities Tailored for Soft and Hard Construction." *What's the Matter? Materiality and Materialism at the Age of Computation*. Ed. Maria Voyatzaki. European Network of Heads of Schools of Architecture (ENHSA), 2014. p101-110.

[57] Mennan 2014, op. cit., p40.

> shaping forces; it suggests rationality, consistency, coherency, organization, and systematization.[58]

Hence, conceptualization or conceptual qualities should not be expected from formalized systems which are not capable of 'calculating' or interpreting these qualities. The expectations of a species (human) cannot be fulfilled by another (computers) which do not have any criticality about those expectations. These phenomenological expectations can be fulfilled by the elaboration of conceptual multiplicity in design methodology: "Such qualitative user-centric spatial issues have been the domain of a less visible computation agenda that is slowly surfacing as spatial performances and user behaviors becoming increasingly important."[59]

With increased importance given to phenomenological qualities, conceptualization of and conceptual approaches to space such as notions of heterogeneous space[60], and empathic space[61] have also become prominent. The notion of 'heterogenous space' is discussed/illustrated by Christopher Hight, Micheal Hensel and Achim Menges. The starting point of authors' discussion on the heterogeneous space is on the inadequateness of the concepts of "complexity in form and multiplicity in programme" which are widely discussed in the current architectural discourse. These concepts are inadequate in providing complex, multiplicitous or heterogeneous spaces. In the authors' point of view, the progressive architectural design of the past 40 years has instead emphasized autonomous form and programmatic determination, which resulted with spaces designed without richer exploration or complexity. The concepts of 'spaces of multitude', 'spatial

---

[58] Terzidis 2003, op. cit., p65.

[59] Derix, Christian. "Introduction, The Space of People in Computation." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014. P15.

[60] Hight, Christopher, Micheal Hensel, and Achim Menges. "En Route: Towards a Discourse on Heterogeneous Space beyond Modernist Space-Time and Post-Modernist Social Geography." Space Reader: Heterogeneous Space in Architecture. Eds. Michael Hensel, Christopher Hight, and Achim Menges. Chichester, U.K.: Wiley, 2009. p9-38.

[61] Derix, Christian and Asmund Izaki, ed. Empathic Space: The Computation of Human-centric Architecture. London: Wiley, 2014.

complexity' and 'heterogeneity' are suggested in order to fulfill the contemporary complex requirements of space in architectural design.[62]

The notion of 'empathic space' is illustrated in AD profile No. 231, *Empathic Space, the computation of Human-Centric Architecture*, dedicated to attempts and approaches on re-integrating perceptual and user-centric qualities of space to computational methodologies. A paradigm of user-centric design is illustrated which is dependent on the user's perception of space and which is promoted by computational methodology, emphasizing the necessity for user-centric, performative, interactive, responsive, effective spaces.[63] It is these qualities that have become criteria for a better space experience, concentrating on issues about space which are more on the human side of design.[64] In this frame, architecture is both concerned with technical and conceptual levels of design while integrating such qualities which, according to Hight, Hensel and Menges, would be absent in the standardized, progressive, over-formalized, and abstract design solutions of the twentieth century.[65]

Likewise, design which solely relies on the technical level could be considered as a mode of advanced computerization, or a frame of over-formalization over architectural design, a mode which is empty in terms of meaning, content and other phenomenological qualities. This frame of over-formalization infers predetermination in design. While discussing the deficiencies of a "mechanical and linear view of causality" Manuel De Landa explains in Henri Bergson's view that:

---

[62] Hight, Hensel, and Menges, op. cit.

[63] Derix and Izaki, op. cit.

[64] An interesting example to this user-centric design approach and user interaction can be seen in interface design for softwares. The interface becomes interactive with the user on many levels and changes its form depending of the type of interaction or medium. The character of the form is dynamic, adaptable, interactive and responsive to its user and conditions. This has been readily adopted by the non-physical virtual space by taking advantage of its dematerialized property, thus its liberation from permanence or stability factor of the physical space. An example to this would be the case of adaptive layouts for websites. See: Marcotte, Ethan. *Responsive Web Design*. New York: Book Apart, 2011. Print.

[65] Hight, Hensel, and Menges, op. cit.

> […] if the future is already given in the past, if the future is merely that modality of time where previously determined possibilities become realized, then true innovation is impossible. To avoid this mistake,he thought, we must struggle to model the future as open-ended, and the past and the present as pregnant not only with possibilities which become real, but virtualities which become actual.[66]

Methodology of architectural design should therefore consider the indeterminate or the intuitive when concerned with phenomenological qualities and conceptualization.

2.2.1. Intuition and Multiplicity in Design Conception

Van Schaik discusses, in a more philosophical approach, that the more subjective and intuitive parts of designers' mind such as his/her knowledge, past experiences, preferences, skills and *spatial intelligence*[67] contribute to design methodologies and results: He provides the term '*mental space*' which is used to describe how designers create ideas.[68] 'Mental space' depends on and is shaped with subjective histories of the designer.[69] In van Schaik's terms, architecture is necessarily a product of the non-static mental space; "the accumulated histories of individuals in spaces".[70] This history is time and place specific, hence depends on the subjective interpretations of the user/observer of space.

---

[66] De Landa, op. cit., p30.

[67] Van Schaik 2008, op. cit.

[68] Schaik provides this description in reference to Mark Turner's 'The Origin of Ideas' in a recent AD Profile. See: Van Schaik, Leon. "How Can Code Be Used To Address Spatiality in Architecture." Ed. Christian Derix and Asmund Izaki. Empathic Space: Computation of Human-Centric Architecture 2014: p141.

[69] Schaik conducts a wide discussion on the term 'mental space' throughout his book titled Spatial Intelligence. He argues that intuition, subjective histories and spatial intelligence of individuals build and diversify mental spaces, hence diversify designs: Van Schaik 2008, op. cit.

[70] Van Schaik, Leon. "The Disruption of the Unity of Time, Place and Architecture, and Some Precursors of Reunification." Spatial Intelligence: New Futures for Architecture. Chichester, England: Wiley, 2008. p57.

On a similar note, Novak argues that architecture should escape formal fixity and should become a means for 'pure' instead of 'applied' architecture while illustrating a set of goals for computational design.[71] In this context, he references the common elements in the behavior of creative persons as noted by Robert Sternberg such as "lack of conventionality" or "integration" and "intellectuality".[72] It is these qualities, which seem to have lost their prominence with the formalization paradigm of computation, that bring multiplicity to design approaches and solutions. Even so, the intuition of the designer affects the formal aspects of design as formalization is an intellectual act, thus depends on intuition as well. There may be multiple ways to formalize an approach but it is through intuition that the 'which' or the 'how' is being decided. This is similar to the case of orthogonal and perspective drawing; both being formalized methods of describing architectural form, the latter includes a higher degree of subjectivity.

This poses the question of how it would be possible to integrate intuitive skills of designers, such as imagination, invention, improvisation, preferences, assumptions, skills, knowledge, experience, and interpretation to a design paradigm which is dominated by formalization and rationalization; how it would be possible to integrate empirical information or physical experience to systems which generate the spatial form through formalized sequences. These intuitive skills are all parts of the mental space of the designer and are not exterior to the design process, even when formalization is most prominent in computational design.

There are attempts which aim to integrate this kind of subjectivity and intuition to computational design methodology through various methods. The common points to these attempts is that they use more elaborate approaches in design conceptualization which consider intention, subjectivity, intuition, content and

[71] Novak, op. cit., p19.

[72] As listed by Novak these elements as: (1) lack of conventionality, (2) integration and intellectuality, (3) aesthetic taste and imagination, (4) decisional skill and flexibility, (5) perspicacity, (6) drive for accomplishment and accomplishment. See: Novak, op. cit., p11.

meaning, and incorporate these aspects with methodology. Mennan presents the works of two architects, whose methodologies are based on custom workflows which simultaneously take formal and intuitive aspects of design into account.[73] She presents two approaches that are exhibited in their work: ‘messy computation’ and ‘strange feedback’. The former which is by Tom Wiscombe deals with mixing the boundaries of methodologies and creates custom workflows depending on the work, where the latter is by Roland Snooks who deals with maximizing the interaction of the formal and generative system with the intuition of the designer. Both of these approaches exhibit systems that are open to designers’ intuitive input and are not over-formalized pre-definitive, pre-declared systems.[74]

This involvement with simultaneously formal and intuitive approaches in the conception of architecture exhibits a multiplicity in approaches towards design problems which manifests itself both at the technical and conceptual level of design. Perspectives differ on approaches regarding design problems. Van Schaik illustrates these perspectives in an observational manner by creating ideograms which describe ideas, concepts, preferences, decisions and other such qualities that belong to the dialogue between design and the mental space of the designer. Ideograms exhibit that each design approach is specific to the dialogue between design and designer and is unique.[75]

It is the “subjective and non-static history of the designer” that provides a multiplicity in design approaches. Intuition, specialized knowledge, and other subjective aspects that are relevant to the act of design imply that no designer is expected to interpret, design or express in the same way as another. Hence, both technical and conceptual levels in design are bound to these intuitive factors no matter how much design relies on formalization of design procedures. Therefore, multiplicity in approaches and potentiality in architecture can be said to rely on both the formalizable/objective and the intuitive/subjective.

---

[73] Mennan 2014, op. cit.

[74] Mennan 2014, op. cit.

[75] For a compilation of such ideograms see: Van Schaik 2013, op. cit.

2.3. Representation of Information

In search for a solution designers make a conceptualization about design, or an abstract design intention which reveals a range of possible solutions regarding the domain of design.[76] In order to convey their intention to human or non-human parties whom take part in the building or generation of form, it is essential for designers to express their abstract design logic in a structured and interpretable form that is efficient in describing the possibility space of design . In this manner, Manuel De Landa argues about the existence of diagrammatized problem solving machines led by visual knowledge. He states that the visual aspect of diagrams are emphasized and he exemplifies it as "[···] the ability of geometric representations to rapidly convey to a problem-solver some of the crucial aspects defining a particular problem, and hence, to suggest possible solutions."[77] The description of possible solutions represents how information is used to provide for design intention and how this intention will generate architectural form; a representation about conceptualization and information regarding the domain of design.

As argued earlier, it is the extended field of specialized knowledge and its renewed relationship with complexity that provides the multiplicity factor to design. It is through this complexity that design embodies all the potential of its domain and achieves a broader possibility space. On the other hand, complexity poses a problematic to the conceptualization of design which is its representation.

> Complexity is a term used to describe the length of a system, or the amount of time required to create a system. [···] Randomness is a term used to describe a lack of an identifiable pattern, purpose, or objective. [···] So randomness is characterized as the maximum of complexity as the opposite of regularity and simplicity.[78]

---

[76] De Landa, op. cit.

[77] De Landa, op. cit., p30.

[78] Terzidis 2008, op. cit.

What occurs as random in the complex corresponds to that which is unrepresentable with the traditional modes of representation. It is hard, if not rarely possible, to create precise descriptions of the complex, especially in the case of custom and non-standard information, with traditional modes of representation:

> Discussions of architecture and complexity, hence, become a play with the unrepresentable. Contemporary information technologies confront architectural-theoretical discourses with developments that call for an expanded theoretical instrumentarium. It remains unclear which architectural language might best be used to approach the concept of complexity associated with information technologies.[79]

In the context of custom and complex information, the question is about finding the kind of representational media that is efficient in describing complex information and possibility spaces.

There are infinitely many ways to represent information, especially when interested in subjectivity and disinterested with convention. From the traditional models; from a simple hand gesture to sketches, models, or drawings to the more custom; such as the diagram and the code, there are many media and methods for representing information or design intention.

The mode for the representation of information is chosen in accordance with the mode of conveying. For example, the two-dimensional character of the orthogonal set falls inadequate for representing the complex double-curved surfaces with high three-dimensional qualities, thus requires an appropriate mode of representation in conveying the three dimensional qualities. Similarly, what conceptualizations of possibility spaces needs in this context of complexity are complex information compositions or, more specifically, custom models for representing information and logic to manage and instrumentalize complexity.

---

[79] Vrachliotis, Georg. "Popper's Mosquito Swarm: Architecture, Cybernetics, and the Operationalization of Complexity." Complexity Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008. P72.

Dialogue between information and conceptualization through representational media has the power to conceive and shape possibility spaces. In this manner, Lorenzo-Eiroa explains in terms of software interfaces:

> Software interfaces and codes constitute implicit frames where artistic expression begins. If the mediums of representation have such a power to regulate the work, then interfaces are spaces of differentiation. As such, interfaces can activate a performative aspect in the work, triggering a formal generative capacity.[80]

In order to be able to convey this custom model of design intention, designers use various modes, which have different characteristics in terms of abstraction, complexity and representation. In this matter, expressions (of intention) are characterized with different qualities that ensure extensibility, complexity, existence of custom information, but more importantly, the manageability of complexity and that of broad possibility spaces.

This chapter presented three issues so far that are concerned with the contemporary scope of architecture in the context of computational design. The first one is the shift to design for possibility spaces that are induced by multi-layered, custom information models, instead of one-time linear resolution of architectural form. The information models used to this end have an inherent complexity that exists as a result of multiple layers of information acting simultaneously in shaping design. The second issue is the existence of a multiplicity in approaches in the conceptualization of design which are fed by subjectivity and intuition in addition to formal and technical aspects of methodologies in design. The third issue lies in the intersection of the preceding two issues and is concerned with the dialogue between information and conceptualization by means of representation, or more specifically, expressions of design intention. There are custom modes of expressions which are characterized by qualities which mainly ensure the

[80] Lorenzo-Eiroa, op. cit., p18.

manageability of complexity and efficiency in conceiving and describing possibility spaces. Such expressions have a prominent role in establishing the dialogue between information and conceptualization of design.

The next chapter provides for a discussion on different modes of expression in terms of diversity and independence in the mode of structuring information. A tension between visual and mathematical/textual expressions are presented, as this duality brings out useful discussions in the context of computational design. Furthermore, expressiveness and relevance to conceptual and perceptual structures of design in terms of both visual and mathematical/textual modes will be investigated.

CHAPTER 3

EXPRESSING DESIGN INTENTION

There is a diversity in modes of expression in architecture where the character regarding the description of architectural form differs in terms of conceptual and perceptual structures. Namely, some of the expressions, such as the orthographic set, are dominant in describing perceptual qualities instead of the conceptual, thus are isomorphic with the form that they describe. Such expressions have a direct relationship with the concretized final form.

This isomorphic correspondence of form to its representation is absent in expressions such as conceptual diagrams, where instead of the specifics of final form, conceptual structures that are relevant to the production of form are conveyed. Conceptual diagrams where conceptualization about design are visualized; or the code where the development of geometry is provided in a relational model expressed in a mathematical/textual syntax are examples of non-isomorphic expressions. Such expressions are not static notations where what is to be built is described, but instead a description or a regulation for a dynamic possibility space for form. These expressions represent relational models with an emphasis on virtuality[81] where relational models denote information models which are shaped by internal relations.

The emphasis in architectural conception has moved from the single object to variable object[82], and consequently from the single product to the process which provides for variability and multiplicity. Architecture has recently become more

[81] De Landa, op. cit.

[82] Novak, op. cit.

concerned with possibility spaces that are constituted by multiplicity and less in single form instances. In this respect, expressions which are isomorphic and thus have a direct relationship to a final form reduce the possibility space to single defined entities. In contrast, expressions that describe the conditions of form production, or conceptual structures seem today more relevant.

Specifically, three modes of expressions provide an important discussion based on reasoning and conceptualization of design in terms of the dualities of visual and mathematical/textual, formal and intuitive, graphical and computational, conceptual and perceptual. These modes are listed as the code, conceptual diagrams and Leon van Schaik's ideograms[83]. Each of these modes of expressions have different characteristics in terms of expressiveness and in establishing the dialogue between information and conceptualization while producing form.

3.1. Structuring Expressions

> Architectural ideas are not usually born as clear and final forms; they arise as diffuse images, often as formless bodily feelings, and are eventually developed and concretized in successive sketches and models, refined and specified in working drawings, turned into material existence through purposefully functioning utilitarian structures in the context of life.[84]

Architecture is a practice which aims to achieve materialized form from immaterial ideas and intentions. Architectural form is either progressively specified during the realization process or selected from a possibility space which includes a family of instances of permissible/satisfactory forms. Media such as sketches, models, drawings induce the process of realizing architectural form from what is immaterial such as information, intentions, ideas and knowledge. During the design process, it is possible to express the concurrent state of knowledge about form and design with

[83] Van Schaik 2013, op. cit.

[84] Pallasmaa, op. cit., p83.

each of these types of media. Media that are used to express information have different characteristics regarding the representation of information.

As previously argued, the dialogue between the conceptualization of design and information through the expression of design intention is prominent during the architectural design process. Whether as an explanation to other parties or to better understand/observe themselves, designers use different types of expressions to describe the specifics about architectural form or its generation. These expressions are embodiments of the abstract design intention which are governed by multi-layered information and the necessity for possibility spaces. In this sense, the representational qualities of expressions have a prominent place in design conception.

Pablo Lorenzo-Eiroa illustrates the relationship between information and language both in terms of linguistics and in terms of the necessity for the representation of information: He argues there is no information without respective representation.[85] It is the representation of information that carries meaning to an interpreter.

> A vectorial line drawn in the computer screen is not a line. It is rather a series of computed codes that simulate a three-dimensional beam of light projected into a two-dimensional screen. The image of this line is therefore a representation of an external binary calculation from its means of constitution.[86]

This relationship between information and its representation is also exemplified in computational processes: Each computation refers to a state and the representation

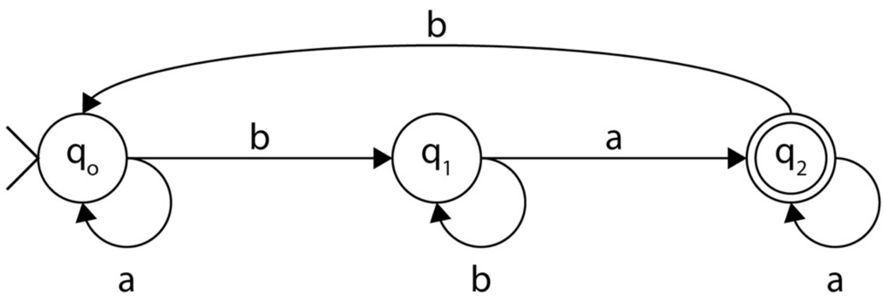
[85] Lorenzo-Eiroa provides a discussion on linguistics by mentioning authors such as Jacques Derrida, Ferdinand De Saussure, Noah Chomsky to cite a few. See: Lorenzo-Eiroa, op. cit., p13.

[86] Lorenzo-Eiroa, op. cit., p13.

of that state.[87] The representation of a state refers to a concurrent condition of information of a computational process [Figure 3.1]. Computation is a matter of the representability of information. It is possible to represent a binary code as a vectorial line through structuring and simulation processes. The relevance of representation here is that of conveying meaning. A binary code, which is a series of 0's and 1's, has an abstract character and is not able convey meaning without representation. Such representation requires an agency or language which would establish the relation between information and meaning and make the interpretation of information possible.

[87] Sudkamp provides an outline of *finite automata* and *abstract machines* that is based on the theory of computer science where *abstract machines* are implementation independent descriptions of computation processes and where *finite automata* are types of abstract machines which are used to determine acceptability of input strings. *Finite automata* are referred as finite as they have finite number of states and processes. Each state represents a state of information about the computation process. See: Sudkamp, Thomas A. "Finite Automata" Languages and Machines: An Introduction to the Theory of Computer Science. Reading, Massachusetts: Addison-Wesley, 1988. P155-196.

State Diagram



Traces For State of Computations

i) abaa

<table>
<tr><th>Computation</th><th>Path</th></tr>
<tr><td>[q_o, abaa]</td><td>q_o</td></tr>
<tr><td>|- [q_o, baa]</td><td>q_1</td></tr>
<tr><td>|- [q_1, aa]</td><td>q_2</td></tr>
<tr><td>|- [q_2, a]</td><td>q_2</td></tr>
<tr><td>|- [q_2, λ]</td><td>q_2</td></tr>
</table>

ii) bbbabb

<table>
<tr><th>Computation</th><th>Path</th></tr>
<tr><td>[q_o, bbbabb]</td><td>q_1</td></tr>
<tr><td>|- [q_1, bbabb]</td><td>q_1</td></tr>
<tr><td>|- [q_1, babb]</td><td>q_1</td></tr>
<tr><td>|- [q_1, abb]</td><td>q_2</td></tr>
<tr><td>|- [q_2, bb]</td><td>q_0</td></tr>
<tr><td>|- [q_0, b]</td><td>q_1</td></tr>
</table>

i) bababa

<table>
<tr><th>Computation</th><th>Path</th></tr>
<tr><td>[q_o, bababa]</td><td>q_1</td></tr>
<tr><td>|- [q_1, ababa]</td><td>q_2</td></tr>
<tr><td>|- [q_2, baba]</td><td>q_0</td></tr>
<tr><td>|- [q_1, aba]</td><td>q_0</td></tr>
<tr><td>|- [q_0, ba]</td><td>q_1</td></tr>
<tr><td>|- [q_1, a]</td><td>q_2</td></tr>
<tr><td>|- [q_2, λ]</td><td>q_2</td></tr>
</table>

ii) bbbaa

<table>
<tr><th>Computation</th><th>Path</th></tr>
<tr><td>[q_o, bbbaa]</td><td>q_1</td></tr>
<tr><td>|- [q_1, bbaa]</td><td>q_1</td></tr>
<tr><td>|- [q_1, baa]</td><td>q_1</td></tr>
<tr><td>|- [q_1, aa]</td><td>q_2</td></tr>
<tr><td>|- [q_2, a]</td><td>q_2</td></tr>
<tr><td>|- [q_2, λ]</td><td>q_2</td></tr>
</table>

Figure 3.1 State Diagram of a Sample Deterministic Finite Automaton and Traces of Computations of Sample String

Produced by the author.

Architecture is in possession of languages or media which are able to convey design information and logic in a meaningful manner, or in more specific terms, which are efficient in expressing design intention. Common examples to such media are sketches, models or drawings as listed by Juhani Pallasmaa in his discussion of the ways in which ideas turn into "utilitarian structures"[88]. Hence, there are multiple ways to eliminate abstraction from information and ensure interpretability and meaning. When concerned with conveying custom information model, there is a diversity of expressions which have different qualities and characteristics in terms of representing multi-layered information and possibility spaces.

3.1.1. Diversity in Modes of Expression

Expression is a generic term which encapsulates any type of interpretable media or act that describes design intention, or in different terms, the descriptions of conditions that produce architectural form. These media, structure design intention for instrumentalization through design process, thus establishing the dialogue between information and conceptualization.

However, this dialogue is not solely constituted by the structure of representation. Thomas Sudkamp illustrates this important distinction between syntax (structure) and semantics (meaning) by mentioning "sentence structures": As he explains, sentences can have the same constituent structure where some can provide sensible meanings where others cannot.[89] Structuring of information does not ensure the expression of meaning and interpretability, therefore meaning and structure are independent from each other however related only in terms of expressions.

---

[88] Pallasmaa, op. cit., p83.

[89] Sudkamp provides a set of rules for generating strings of sentences therefore providing a partial syntax for sentence generation in English. He applies rules to a limited vocabulary to generate sentences. It is possible to generate sentences that have the same constituent structure such as "John eats slowly." or "The car eats slowly." where the former can provide a sensible meaning whereas the latter cannot. See: Sudkamp, op. cit., p55-86.

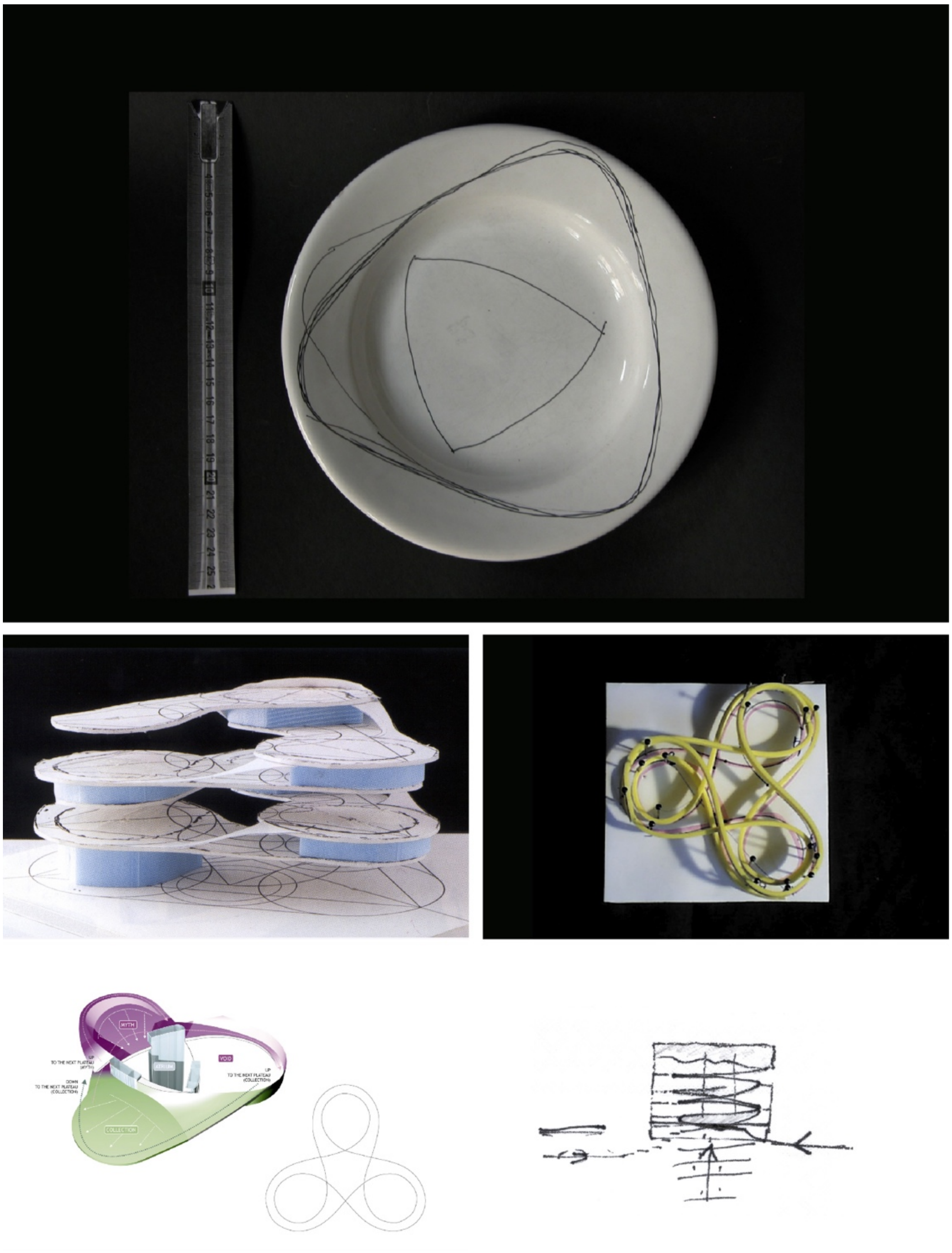


Figure 3.2 Different Modes of Expressing Architectural Information - Mercedes Benz Museum by UNStudio

Source: Images taken from UNStudio website, put together by author.

"Mercedes-Benz Museum by UNStudio." Mercedes-Benz Museum » UNStudio. Web. 12 July 2015. <http://www.unstudio.com/projects/mercedes-benz-museum>.

In this manner, it is possible to consider a simple hand gesture[90] or even markings on a plate [Figure 3.2] as an expression of design intention, since such acts or media have the expressive capability to describe form. Designers express design intention by different modes of representational structures which have different characteristics in terms of conveying information, convenience, formalization, intuition, specificity, complexity and level of abstraction to cite a few. These are characteristics of expressions which regulate their structure and expressive capability.

> For every medium there is a language, for every expression there are motives, codes, contexts which differ for the maker, the user, the critic. There is no firm ground, and precisely in that inherent absence of a firm ground we find delight.[91]

The diversity in representational structure spans from the more common modes of expression such as the sketch, which is more intuitive and subjective, to the more conceptual and visual such as FOA's diagram of the Yokohama Terminal where the governing logic of the design is represented via simple linear flows [Figure 3.3], to the more abstract and observational such as Leon van Schaik's ideograms [Figure 3.4], and to the relatively new mathematical/textual and formal type such as the code.

Being an earlier example of usage of conceptual diagrams in design, FOA's diagram holds a prominent location. The importance of the diagram is increasing today, arguably due to the increased relevance of immaterial design systems of

[90] The importance of body language or hand gestures in describing design is highlighted by Coop Himmelb(l)au, an architectural practice which is based in Vienna. See: "The Dissipation of Our Bodies in the City." Coop Himmelb(l)au. Web. 8 Aug. 2015. <http://www.coop-himmelblau.at/architecture/philosophy/the-dissipation-of-our-bodies-in-the-city>.

[91] Novak, op. cit., p12.

computational design and their relationship with information.[92] Computation provides for customized information interfaces, or different types of representing, mapping or visualizing information and knowledge, hence increasing interpretability (by humans) of otherwise complex and unintelligible information.[93] Diagrams act as virtualizing, actualizing and materializing agencies in the immaterial space of computation.[94]

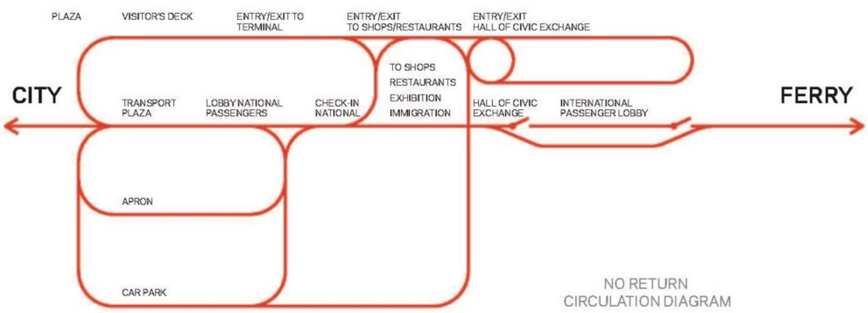


Figure 3.3 Yokohama International Passenger Terminal No Return Circulation Diagram by Foreign Office Architects (FOA)

Source "Gallery - AD Classics: Yokohama International Passenger Terminal / Foreign Office Architects (FOA) - 17." ArchDaily. Web. 12 July 2015. <http://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa/542078f1c07a8086fc00000a_ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa-_yoko_circulation_diagram-png/>.

The code's relevance as a mode of expression is promoted by the computational design paradigm. The code is different from visual expressions in terms of

[92] There is increasing literature on the diagram since Peter Eisenmann's Diagram Diaries and the 23rd issue of Any Magazine. Futhermore, Mark Garcia illustrates the contemporary location of diagrams in relevance to computational systems, and furthermore, proposes futures for their usage. See: Eisenman, Peter. *Diagram Diaries*. New York, NY: Universe Pub., 1999; Any Magazine Issue 23, *Diagram Work: Data Mechanics for a Topological Age* 1998; Garcia, Mark. "Epilogue (A Beginning of Other Diagrams of Architecture and the Futures of the Diagrams of Architecture)." *The Diagrams of Architecture*. Mark Garcia, ed. Chichester: Wiley, 2010. p310-315.

[93] Ibid.

[94] Ibid.

representation and description. It is a formal digital substrate of form generation and at the same time an expression which describes the geometric conditions of form. Moreover, it is free in terms of interactions and expressiveness as the code exceeds the pre-programmed limited graphical user interfaces of software, but is itself only limited in terms of a programming language and analytical-geometrical operations.[95] Code, or algorithm, in Kostas Terzidis's definition:

> […] is a computational procedure for addressing a problem in a finite number of steps. It involves deduction, induction, abstraction, generalization and structured logic. It is the systematic extraction of logical principles and the development of a generic solution plan.[96]

The code is a formal relational complex system where form is generated, tested and evaluated heuristically. The numerical and mechanical nature of the computer highlights the necessity for a formal logic in the production of form, therefore the code rises questions such as "how can architectural meaning be generated and even shaped by a technology whose operations are non-semantic in nature?"[97]. Through the code, form is expressed in quantitive and relational means, whereas manipulations, evaluations and combinations of these processes correspond to qualitative processes.

Despite its formal nature, intuition is embedded within the code where it becomes possible to provide solutions to complex problems belonging to domains such as the social or the cultural. The computational leap is immense for precise calculations of such complex problems, therefore providing expressive descriptions of such problems in terms of formal mathematical systems that require formalized

---

[95] Lorenzo-Eiroa, op. cit.

[96] Terzidis, op. cit., p65.

[97] Vrachliotis, op. cit., p72.

conceptual structures about problem solutions.[98] There are multiple, if not infinite ways in mathematical systems to reach a definitive result, but how that result is achieved is a matter of conceptualization, therefore of subjective approaches.[99]

Similarly, there is structure, therefore formalization, in intuitive, conceptual and visual media, for example in diagrams [Figure 3.3] or in van Schaik's ideograms [Figure 3.4], where structuring information through notational systems increases the interpretability of the subjective and intuitive mode of expression. In this sense, when considering the code, conceptual diagrams and ideograms, expressions can be considered as both intuitive structures and structured intuition, namely, formal structures which are induced by intuition as exhibited in the code, and intuition which is structured into a formal mode, as in the case of ideograms or conceptual diagrams.

[98] Klaus Mainzer provides a discussion on complexity of social and natural problems through notions of dynamicism, multiplicity, self-organization and open-systems. See: Mainzer, Klaus. "Strategies For Shaping Complexity in Nature, Society, and Architecture", Complexity Design Strategy and World View. Gleiniger, Andrea, and Georg Vrachliotis, eds. Basel: Birkhäuser, 2008. p89-98.

[99] This case is exemplified with sorting algorithms which sort a list of items or array by special logic. There are a number of sorting algorithms where each have a different technique for sorting a shuffled list but achieves the same sorted list in the end of execution. This diversity is based on requirements of performance in sorting lists that are shuffled differently. For an interactive representation comparing different sorting algorithms for different lists see: "Sorting Algorithm Animations." Sorting Algorithm Animations. Web. 12 July 2015. <http://www.sorting-algorithms.com>. Finally, for a detailed animation for sorting algorithms see: "15 Sorting Algorithms in 6 Minutes." YouTube. YouTube. Web. 12 July 2015. <https://www.youtube.com/watch?v=kPRA0W1kECg>.

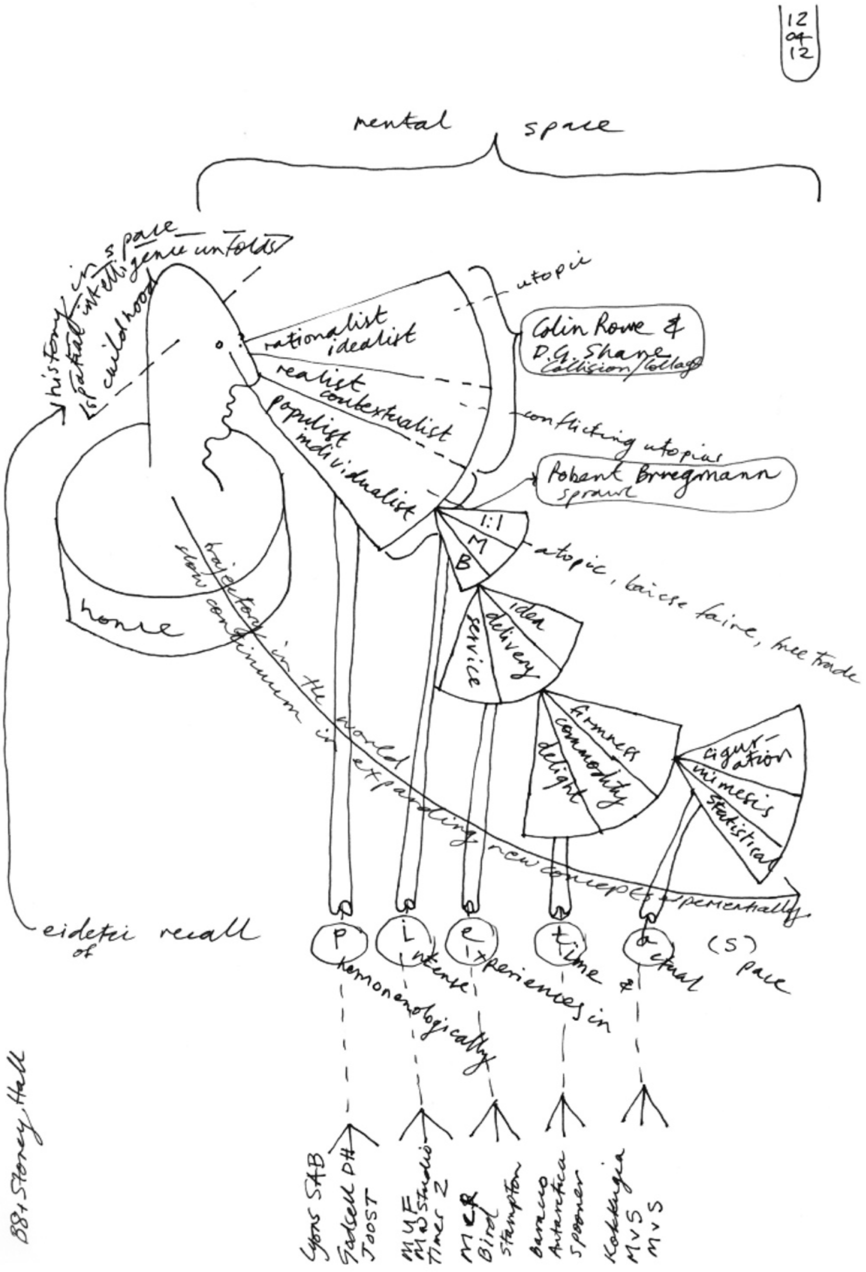


Figure 3.4 Differentiation in Vital Practive, Ideogram by Leon van Schaik

Source Van Schaik, Leon. Ideograms. Melbourne: Lyon Housemuseum, 2013. p67.

3.1.2. Independence of the Mode of Structuring from Information and Expressiveness

The diversity in representational modes shows that expressions do not have common structural properties, or in more specific terms, a common syntax but that each mode has different 'resistances' in terms of expressiveness. The idea of resistance, as given in Van Schaik's description:

> […] is the term used by artists to describe the qualities of a medium in which they work: a good medium offers resistance to their immediate impulses and, in doing so, causes a dialogue between artist and medium that the best amongst them welcome.[100]

There are an indefinite number of factors concerning the diversity of expressions, hence their resistances, such as different levels of specificity and abstraction, different notational systems, different methods in conceiving, different constraints and different levels of control over the conveying of information or expressiveness. Malcolm McCullough argues that code as a formal, constraining, and therefore, 'resistant' medium, would not hamper the creativity of designers but instead promote richer results:

> Any expressive medium has its idioms, types and genres, and the better established of those are often where the richest expressions occur.[101]

It is possible to express intention in a specific representational mode if that mode has properties such as the ones listed by McCullough, i.e. idioms, types and genres that are efficient in expressing such intention. Each mode of expression has its limitations and advantages. Consequently, with increased power in expression, the complexity in the construction of an expression increases.[102] There are different modes of expression where certain things are easier and less complicated to express.

---

[100] Van Schaik 2008, op. cit., p166-167.

[101] McCullough, Malcolm "20 Years of Scripted Space", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p183.

[102] Authors Micheal Huth and Mark Ryan argues on the need for a richer language and the power of expressibility by a discussion on propositional logic, first-order logic (predicate logic) and second-order logic. See: Huth, Michael, and Mark Ryan. "Predicate Logic." Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge: Cambridge UP, 2000. P93-107.

This situation can be exemplified by formal systems of logic. Propositional logic only operates with logical connectives such as conjunction (and), disjunction (or), negation (not), and implication (if).[103] First-order logic operates with, on top of logical connectives that are used in propositional logic, with predicates, functions and quantification of variables with the universal quantifier (for all) and the existential quantifier (there exists).[104] Higher-order logic, on top of operations of lower orders of logic, operates with quantification of relations, functions and sets of variables.[105] Each higher order of logic is more complex in terms of expression but more expressive than lower orders [Figure 3.5]. As things become more expressive, they become harder to represent, and in the case of computation, harder to compute.

Direct conversion between these modes of expression without change in meaning or conveyed information is not possible since each mode has different characteristics regarding the representation of information. Through conversion, the represented information is restructured into a different form, therefore resulting in a change in meaning and interpretation; while some information is obscured, others become more visible.

> In architecture, the methods of projection also serve a subliminal purpose. While axonometric views are considered exact, precise, accurate, and measurable, perspective views are empirical, observable, factual and expressive. Perspective projection is about the depicted object's identity and characteristics.[106]

Kostas Terzidis discusses the convention-based methods of axonometric and perspective views, and their difference in quality in the depiction of space. Both axonometric and perspective projections are formalized techniques but the latter is concerned with subjective qualities such as perception where the former is not.

[103] Ibid. p31-53.

[104] Ibid. p93-171.

[105] Ibid.

[106] Terzidis 2003, op. cit., p57.

Purpose of and emphasis on information changes with the mode of expression. Hence it is especially difficult to convert two contrasting types of expressions into each other, for example, to translate a conceptual diagram into a code.

Propositional Logic

Statement: No tie, no suit, no service.

Atomic sentences:

p: tie worn
q: suit worn
r: service given

Proposition:

$(\neg p \vee \neg p) \rightarrow \neg p$

First-order Logic (Predicate Logic)

Statement: Everyone is loved by someone.

Function:

Love(y,x): y loves x.

Proposition:

$\forall x \exists y \; Love(y,x)$

Figure 3.5 Comparison of Propositional Logic and First-order Logic (Predicate Logic) in terms of Expressive Power and Syntax

Produced by the author.

The relationship between information and its representation depends on multiple factors. As illustrated here, these factors include the independence of information from representational structures, the diversity in structuring information for interpretability, the conceptualization in expression, different resistances in expressive media, and the complexity of expression and expressiveness. These are prominent and effective factors in establishing the dialogue between information and the conceptualization of design.

3.2. Visual versus Relational in their relation to the Conceptual and the Perceptual

Through the promotion of computational methodologies, architectural conception is endued by an increased amount of information. The new way of conceiving architectural form relies on custom relational information models that are efficient in addressing complex problems and incorporating multi-layered concerns. On the one hand, this puts the emphasis on such relational models which are based on information protocols instead of visual structures that represent them. On the other hand, however, this interest in relational models brings out a disjunction of the visual from the relational in architecture which is a discipline having a distinct relationship with the visual and the perceptual.

> Media communications have advanced a sensibility and education based on the understanding of a visual logic that was highly beneficial to architecture - a visual arts discipline based on formal logic. Media has separated visual appeal and affection from the underlying protocols engineered to manipulate mass behavior. Therefore the visual is no longer a paradigm for reference, as underlying codes have now become referential.[107]

To illustrate this disjunction, Pablo Lorenzo-Eiroa mentions the independence of the software interfaces from the underlying relational information layers, and as a consequence of this, a disjunction from the visual and perceptual structures is observed.[108] For instance, in websites and web based services, it is possible to provide multiple interfaces for different devices by maintaining the independence of information from its representation. Each device obtains different perceptual structures in terms of representations of information and different modes of interactions that are suitable for the device without affecting the business-logic and information layers of the software [Figure 3.6].

---

107 Lorenzo-Eiroa, op. cit., p11.

108 Lorenzo-Eiroa, op. cit., p11.

Figure 3.6 Different Interfaces of Facebook on Desktop, Tablet and Smartphone Screens

Produced by the author from screenshots of social media Facebook on different devices and applications.

As argued earlier and similar to software interfaces, expressions are independent in terms of the representation of information and therefore there is a multiplicity of modes in representing design intention. Expressions refer to a state of information and design logic, hence represent a state of architectural form, either the inherent relations which make form (the conceptual structure, or a representation of the conceived form itself) or the perceptual structure through different interpretable interfaces. The importance of this differentiation of the relation between information and its representation is that expressions are agents in rationalizing and reasoning with design intention where different modes of expression provide different design possibilities. Reasoning about situations requires constructing formal and logical arguments about such situations[109] and expressions are what constitutes reasoning in design.

---

[109] In the context of computer science, authors Huth and Ryan, discuss the aim of logic as developing formal arguments to validate and defend them rigorously, or to be able to execute them on a deterministic machine. Huth and Ryan, op. cit., p1.

There are two modes of reasoning in architecture, as named by Lorenzo-Eiroa; visual logic and relational logic, each relating to perceptual and conceptual structures differently.[110] Similarly to Pablo Lorenzo-Eiroa's assertion and in the context of computation, Zeynep Mennan mentions contrasting forms of reasoning as stated by Bruno Bachimont; that of computational and graphical reason. The formal computational reasoning is conveyed by the unfamiliar numerical inscriptions of computational rationality that lack material or visual associations. Mennan argues that such modes of formal inscription and reasoning are preoccupied with form, except for its content and meaning, or in wider terms, with perceptual and phenomenological qualities. Mennan further mentions that different ways in inscribing or thinking, such as the computational and the graphical have the power to implement different meanings, concepts or results that are suitable to their nature.[111]

When considering the two authors' discussion, it is possible to state that, the dialogue between conceptualization and information through expressions is based both on visual and relational reasoning. Furthermore, as noted earlier, relational logic, a mode of reasoning not essentially involved with visual structures, poses a challenge to architecture which is essentially visual or perceptual in nature.

In this context of the duality in reasoning and the disjunction between conceptual and perceptual structures of design, there are two relevant modes for structuring expressions; the visual and the mathematical/textual. These modes are widely

---

[110] Lorenzo-Eiroa, op. cit.

[111] Mennan 2014, op. cit.

discussed in terms of their characteristics or conditions in design processes[112] but it is essential to discuss some key remarks about the relationship of these modes of structuring with the relational and visual modes of reasoning in the context of design within the computational media.

Visual structuring of expressions is based on the reliance to images and perceptual structures in architecture[113] both through drawings that are directly related to architectural form or through conceptualizing agents such as diagrams and ideograms. Visualization of information, therefore, has a distinct importance in architecture in providing possibility spaces and realizing architectural form. In addition to visual structuring, with the advent of computational methodologies in architecture, the mathematical/textual mode of structuring has become equally relevant due to its efficiencies in managing complexity and providing custom information models that are constituted by multi-layered concerns and potentialities. This is especially the case for diagrams as there is an increased amount of discourse on the subject in the last three decades.[114] It is possible to argue that this relatively new interest in diagrams is parallel to the advances in computational research and the necessity for a visualizing agency for the immateriality of the computational medium.

Comparison of these two modes is important in the context of conceptual and perceptual structures since the former has been associated with architecture, its image and its visual-perceptual structures, and the latter is promoted by new

---

[112] Zeynep Mennan provides a discussion on the duality of visual and mathematical/textual expressions. The unfamiliar mathematical/textual expression exhibits a replacement or displacement of the visual and perceptual. There are also books and articles which illustrate the location of visual and mathematical/textual expressions. For Mennan's discussion on non-visualization see: Mennan 2006, op. cit. For visual structuring see: Garcia, Mark, ed. The Diagrams of Architecture. Chichester: Wiley, 2010; Van Berkel, Ben and Bos, Caroline. "Diagrams - Interactive Instruments in Operation". Any Magazine Issue 23, 1998. p19-23; De Landa, op. cit. And for mathematical/textual structuring see: Terzidis 2003, op. cit. p65-73; Legendre, op. cit.; McCullough, op. cit. p182-187.

[113] Seray Türkay provides a discussion on the relationship between image and architecture through visual representational devices in architecture, specifically the orthography. See: Türkay, op. cit.

[114] Garcia, Mark, ed. The Diagrams of Architecture. Chichester: Wiley, 2010

methodologies and the new way of conceiving architecture that relies on relational information models. These modes of structuring expressions present differences and similarities in terms of conceptual and perceptual structures, visual and relational logic, possibility spaces, and expressiveness.

3.2.1. Visual Structuring of Expressions

Kostas Terzidis refers to Peter Eisenman's concept of an "architectural diagram as an explanatory, analytical, generative or representational device" to state that this concept is directly dependent on human interpretability.[115]

> This human-centric approach is implicit within the sphere of subjective phenomena and personal interpretations. Within that realm, any logic that deals with the evaluation or production of form must be, by default, both understandable and open to interpretation.[116]

Information, or more precisely, abstract conceptualizations of design require structuring to become perceivable and open to interpretation, without which they remain abstract and invisible. Architecture, a profession which relies on image and perceptual structures[117], presents a diversity in the structuring of expressions, especially in visual form. Drawings such as the orthographic set which are directly related to built form, intuitive sketches which are quick and subjective, and diagrams which present conceptual approaches and relations rather than to-be-built form are common examples of such visual structures. In terms of the visual

[115] Terzidis argues that algorithmic processes are involved with human perception or interpretation but are rather considered as extensions of human mind through exploration and codifications. He further argues that diagrams are logically structured materializations of both machine's extensibility and human mind. See: Terzidis 2003, op. cit., p65-73.

[116] See: Terzidis 2003, op. cit., p70.

[117] Evans, Robin. "Architectural Projection." Architecture and Its Image: Four Centuries of Architectural Representation : Works from the Collection of the Canadian Centre for Architecture. Ed. Eve Blau and Edward Kaufman. Montreal: Centre Canadien D'Architecture/Canadian Centre for Architecture, Cambridge, Mass; Distributed by,the MIT Press. 1989. P21.

structuring of information and visualizing design intention, modes of structuring expressions differ in their focus on conceptual and perceptual aspects of design.

Visual expressions which refer to conceptual structures instead of the perceptual are informal, dynamic, decentralized, dematerialized, dislocated and timeless.[118] Such expressions are associated with abstraction from the perceptual qualities of form, but instead involved with diagrammatical reasoning which provide for its production. Manuel De Landa mentions the power of visual knowledge in providing diagrammatized problem solutions where crucial aspects of conceptual structures are represented in geometric form.[119] This form of diagrammatic expressions are involved with the representation of abstract conceptualization, or design intention: A visualization of design intention in abstract structure.

> Conceptual diagrams are abstract representations that embed the core of a conceptualization of a problem solution. They are concise, yet powerful aids in problem solving in that they provide high-level commitments constraining solutions. In architecture, they embed the core of a design solution encapsulating its *generic* characteristics and constraints and conveying the form of possible *specific* solutions.[120]

As discussed by Doğan and Nersessian, conceptual diagrams are a mode of visual reasoning which aid design by specifying fundamentals of the conceptualization level of design and deterring from perceptual structures such as commitments to a final architectural form and therefore allow exploratory reasoning.[121] Authors further discuss that such conceptual diagrams "are not ambiguous in the way sketches are in that they fix meaning and define a set of related solutions."[122] Such diagrams are not representations of explanations about static, predetermined and

---

[118] Ben van Berkel and Caroline Bos provide a discussion of conceptual diagrams. See: Van Berkel and Bos, op. cit.

[119] De Landa, op. cit., p30.

[120] Dogan and Nersessian 2002. op. cit., p353.

[121] Dogan and Nersessian 2002. op. cit., p353-355.

[122] Dogan and Nersessian 2002. op. cit., p353.

finalized concepts but are more dynamic definitions of relations, spatial conditions, intensities and entities about design.

Conceptual diagrams are discursive devices: They establish the dialogue between the abstract design intention and information through image.[123] They are not fixed in meaning, thus they allow for multiple interpretations.[124] The power of visual reasoning lies in this hermeneutical nature and abstraction is a substantial aspect of such reasoning. In their concluding remark about diagrams and their interactivity in terms of interpretation, van Berkel and Bos mention them as Deleuze's abstract machines which oscillate between the real and the abstract:

> The abstract machine in motion is a discursive instrument: it is both a product and a generator of dialogical actions which serve to bring forth new, unplanned, interacting meanings.[125]

Authors argue that such diagrams are about the "real that is yet to come".[126]

Expressions that describe conceptual structures are concerned with fundamentals about design instead of specifics of form and perceptual structures and therefore apply a degree of abstraction to form. Sudkamp argues that the analysis for the working conditions or conceptual structural basis of a system requires the separation of fundamentals of design from the implementation details. Additionally in this discussion, he provides a description for abstract machines: "The

[123] Doğan and Nersessian claim that external representations used in design process are more than tools for conveying information, instead they allow for interactions with forming and evolving ideas in dynamic thinking. Authors illustrate this aspect of visual media, or namely conceptual diagrams, through the case of Daniel Libeskind's Jewish Museum. See: Dogan, Fehmi and Nersessian, Nancy. "Conceptual Diagrams in Creative Architectural Practice: The case of Daniel Libeskind's Jewish Museum", arq: architecture research quarterly, 2013.

[124] Van Berkel and Bos, op. cit., p19-23.

[125] Van Berkel and Bos, op. cit., p23.

[126] Van Berkel and Bos, op. cit., p21.

implementation independent description is often referred to as an abstract machine."[127]

Non-isomorphic expressions, expressions which do not resemble the form that they produce, are similar to abstract machines that are defined by Sudkamp as they provide specifics about the conceptualization of design. Implementation details refer to perceptual structures and are absent in abstract machines. Therefore, non-isomorphic expressions require a dematerialization of form into conceptual structures:

> An abstract machine in itself is not physical or corporeal, any more than it is semiotic; it is diagrammatic (it knows noting of the distinctions between the artificial and the natural either). It operates by matter, not by substance; by function, not by form. Substances and forms are of expression "or" of content. But functions are not yet "semiotically" formed, and matters are not yet "physically" formed. The abstract machine is pure Matter-Function–a diagram independent of the forms and substances, expressions and contents it will distribute.[128]

Deleuze and Guattari provide a similar description of the abstract machine that is independent of matter and form.[129] As explained by De Landa, this type of

[127] Sudkamp provides an analysis of a newspaper vending machine by its working states where current state of the machine changes every time it receives input, or in terms of the vending machine, coins. Examples to state descriptions are: "needs 30 cents", "needs 25 cents", "needs 20 cents" and so on. The vending machine reaches the "needs 0 cents" state if enough coins are inserted and dispenses a newspaper. This description of the states are abstract, and provides the fundamental working principles of a system which dispenses newspapers. It is possible to embody this logic by any type of real dispensing machine. Sudkamp, op. cit., p157.

[128] Deleuze, Gilles and Guattari, Félix. "587B.C. - A.D.70: On Several Regimes of Signs." A Thousand Plateaus: Capitalism and Schizophrenia. Trans. Brian Massumi. Minneapolis: University of Minnesota Press, 1987. P141.

[129] Ibid. p111-148.

diagrammatized thinking is involved with detachment from reality by abstraction in the form virtuality or actuality.[130]

Increased specificity and details in either one of the perceptual and conceptual structures of design narrow the possibility space about conceptualization, or the "diagrammatic space of energetic possibilities" as de Landa puts it[131], to a single entity. Doğan and Nersessian emphasize the importance of abstraction from specificity, where less specified representations of information highlight the most common features of the design scheme.[132] The abstract diagram is about conceptualization of design, which describes how form will be conceived and providing for potentialities that are independent from any specific and concretized form. On the other hand, drawings such as the orthographical set, describe how the form will be in reality. They are representations of the finalized form but not of the design intention.

Therefore, abstract visual definitions are utilized for rationalizing, formalizing and reasoning with relations inherent to form where concrete visual definitions exist for representing the finalized concrete architectural form. In both modes of visual structuring, different scales and aspects about conceptual and perceptual qualities of form are described. The important aspect which relates to the scale of the perceptual or conceptual degree of the expression used can be argued to be the level of abstraction over form. Abstraction allows to reason with logic in the context of complex information and multi-layered information models.

3.2.2. Mathematical/Textual Structuring of Expressions

Computational methodologies have provided ways to solve complex problems with multi-layered information models through formalization of conceptualizations

---

[130] De Landa discusses the relationship of the virtual and real through diagrammatization and the notion of "abstract oscillating machine" which shifts in between actual and real. See: De Landa, op. cit., p30-34.

[131] De Landa, op. cit., p30.

[132] Dogan, Fehmi and Nersessian, Nancy. "Generic Abstraction in Design Creativity: the Case of Staatsgalerie by James Sterling", Design Studies No.31(3), 2010. p207-236.

about design. The mechanical and digital substrate of computers requires formal arguments to input, operate on information and provide results.[133] It is the deterministic nature of the simple input-process-output chain in computational methodologies which mandates the use of formalized, unambiguous, decidable and determinable logic.

Sudkamp states that "[a] mathematical structure consists of functions and relations on a set or sets and distinguished elements from the set" while mentioning preliminary mathematical notions and concepts for computation.[134] In terms of programming, "functions and relations" translate to control structures where what he calls as the "set or sets" translates to data structures. Programming is essentially a combination of the control of thought and control of information.

The code, as an expression, is constituted by control structures such as defining, looping, sequencing, and conditionals, and simple (holds a single value) or complex (holds multiple values) data structures where more complex data structures can be built from simpler ones. From 0 to 3 dimensions and more, the production of form by way of the code has a generative methodology or a methodology which is involved with formation through these structures.

Challenges of a paradigm based on control through formal logic in the computational medium manifests itself on the mathematical/textual structuring of expressions in terms of interpretability. Expressions which privilege machine interpretability renders themselves as abstract and less interpretable to humans, whereas expressions which are ambiguous and discursive such as ideograms or diagrams have a higher degree of interpretability and bring out a tension in the computational medium. One may argue that this would impact the computational medium negatively when approaching design problems more intuitively. But it should also be considered that formal logic and determinability provides for a

[133] Huth and Ryan, op. cit., p1.

[134] Sudkamp, op. cit., p28.

"traceable creativity"[135] in design, an aspect which enables the recreation of design processes.

Another similar issue in this context is the gap between the more intuitive modes of expressions that are mostly visual (conceptual diagrams or ideograms) and the formal ones that are mostly mathematical/textual (the code). It is only possible to bridge the gap between these contrasting types of expressions if there are commonalities between underlying logic, design intention and descriptions that are represented in the expression. Otherwise a conversion between these contrasting types requires another mediating agency in providing common points between formal and intuitive approaches in design.

Similar to the visual structuring of expressions, the perceptual and conceptual aspects of design intention have a different relevance in mathematical/textual structuring. It seems important to discuss this difference of relevancy through the distinction between computerization and computation.

> Computerization is about automation, mechanization, digitization, and conversion. Generally, it involves with the digitization of entities or processes that are preconceived, predetermined, and well defined.[136]

Computers and computational methods provide ways to manipulate and generate form by infinite variation without manual labor. It is possible to define a generation sequence for form as long as the geometrical conditions about form are formalized. Computerization is involved with automating the generation of realized, concretized form, independently from a conceptual basis, thus providing for a narrow possibility space. Reproducibility of the process is a beneficial aspect about mechanization or digitalization that is exhibited in computerized methodologies but a systematization of form generation over a predetermined and preconceived basis

[135] Terzidis 2003, op. cit.

[136] Terzidis 2003, op. cit., p67.

is a method for generating concretized form with similar variations, thus provides as narrow possibility spaces as modes of drawing such as the orthographic which have direct relationship with the built form.

> In contrast, computation is about rationalization, reasoning, logic, algorithm, deduction, induction, extrapolation, exploration and estimation. In its manifold implications, it involves problem solving, mental structures, cognition, simulation, and rule-based intelligence, to name but a few.[137]

As argued by Kostas Terzidis, computation is distinct from computerization and is involved with a different paradigm; one which is constituted by the formalized conceptualization of solutions instead of a predetermined logic.[138]

It is possible to solve a problem by a formal system only if the problem is decidable. Therefore, the code is induced by formalizable and quantifiable problems: Concepts such as “the closest distance” and “the furthest point” are examples to quantifiable aspects of geometry which are referred to when providing solutions to problems via computational methodologies. Such aspects are comparable and calculable by quantifiable means. The reason for the emphasis on quantifiability is that such aspects are operational without ambiguity by way of the determinate nature of numbers, and are therefore, decidable.

Quantifiability brings convenience in providing solutions by using formal systems but computation does not mandate a deterministic quantifiability. Computation is more than the reduction of complex conceptual intentions to quantities as the design of an algorithm is more involved with conceptualization about providing a solution: Providing an algorithm requires conceptualization and design intention for solving a problem.

---

[137] Terzidis 2003, op. cit., p67.

[138] Terzidis 2003, op. cit., p65-73.

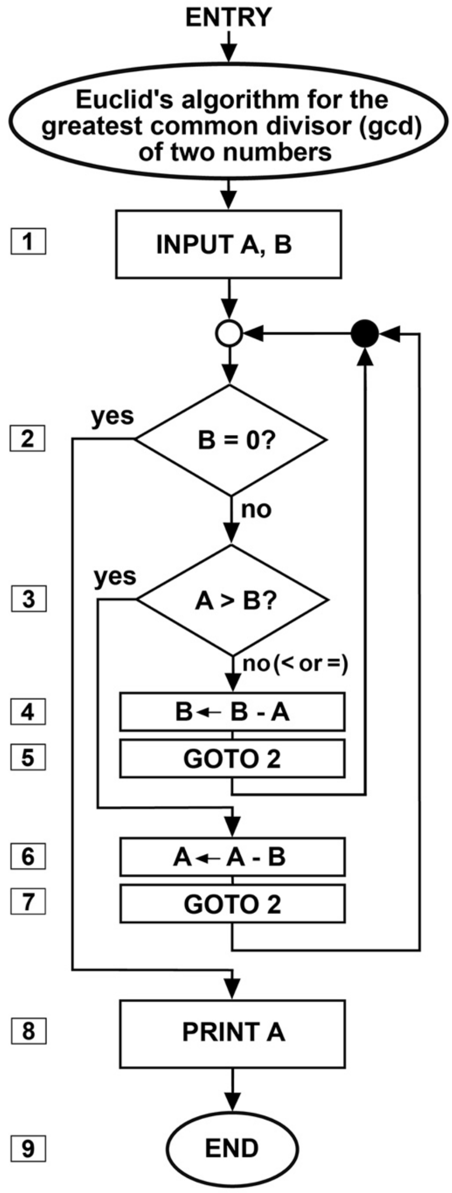


Figure 3.7 Flow Chart of an Algorithm for Calculating the Greatest Common Divisor (Euclid's Algorithm)

Source: "Algorithm." Wikipedia. Wikimedia Foundation. Web. 13 July 2015. <https://en.wikipedia.org/wiki/Algorithm>.

An algorithm is a set of finite number of sequenced instructions to solve a problem in terms of calculation, processing data and automating reasoning. It defines the fundamentals about the problem solution [Figure 3.7]. It is possible to express algorithms in different forms such as flowcharts, pseudo code, natural languages or programming languages. This being the case, there is a distinction between code and algorithms.

Code is more specified than algorithms and is more involved with the substrate or machine that it is executed on. Programs are designs arising from algorithms and control the behavior of the machine that generates solutions. Each line in the code is written in order to express or output, in a way similar to visual structuring where each mark in a drawing serves a purpose. The code puts the emphasis on the relations that make form instead of its implications as form while providing form through its execution. It is not an abstract machine, but it is both the implementation and the description of a solution.

The code, therefore, is dependent on the substrate on which the code is being executed, therefore the expression of an algorithm changes depending on the substrate. The substrate provides interfaces for manipulating geometry or processing information. Furthermore, different programming languages provide different services: It is different to provide solutions with, for instance Grasshopper[139] [Figure 3.8] or Python[140] [Figure 3.9]. Such technologies can be said to have different 'resistances'[141]. The abstract definition of an algorithm does not have such involvements with a substrate for execution. In this case, the kind of possibility space conceived with an algorithm is related to its substrate, hence the code.

There is a similar relation with the degrees in the expression of perceptual and conceptual aspects of design intention in the code and the ones with drawings and conceptual diagrams. Creating a formal description of possibility spaces for form

---

[139] Grasshopper is a software plug-in built for Rhinoceros modelling software and is used to compose scripts with visual syntax that parameterize geometry. See: "Grasshopper." Algorithmic Modeling for Rhino. Web. 12 July 2015. <http://www.grasshopper3d.com>. And see: "Rhinoceros." Rhinoceros. Web. 12 July 2015. <http://www.rhino3d.com>.

[140] Python is a programming language which has gained relevance recently as it is supported by popular modelling software such as Rhinoceros and Grasshopper. See: "Welcome to Python.org." Python.org. Web. 12 July 2015. <https://www.python.org>. And see: "McNeel Wiki." Python Scripting for Rhino and Grasshopper. Web. 12 July 2015. <http://wiki.mcneel.com/developer/python>.

[141] Van Schaik 2008, op. cit., p166-167.

is based on conceptualization, similar to conceptual diagrams where fundamental relations of design intention are described instead of the predeterminate logic exhibited in computerized methodologies. The perceptual does not exist until the code is executed, but when the code is executed, form exists as in a concretized state in terms of geometrical representation, similar to the orthographic set. The code is defined by the conceptual and it produces the perceptual. It is concerned with both actualization and realization, or conceptual and perceptual aspects at the same time, therefore incarnating wide possibility spaces and variability.

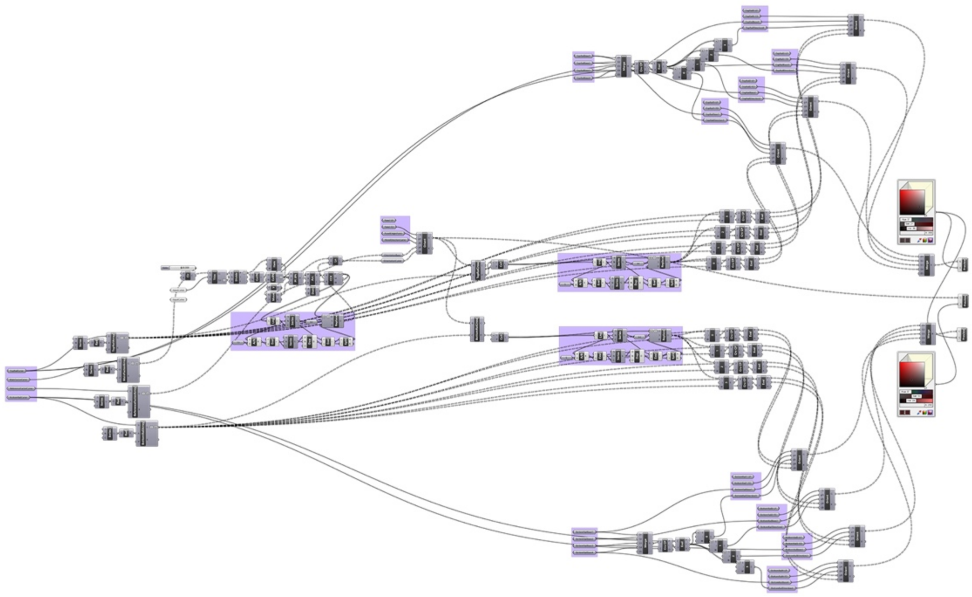


Figure 3.8 Sample Grasshopper Algorithm Showing its Canvas Based Data-flow Syntax

Produced by the author.

```python
# Turns a number into one (eventually)
def collatz(n, hist = []):
    # Apply conditions
    if(n % 2 == 0):
        n = n/2
    else:
        n = 3*n+1

    # Put into list, unecessary but fun to record history
    hist.append(n)

    # Apply conditions if not one
    if(n == 1):
        #Return history
        return hist
    else:
        #Recursive call for collatz
        return collatz(n, hist)





# Read user input
num = int(raw_input('Enter a number: '))

# Call collatz
history = collatz(num, [num])
# Output stuff
print 'Turned ' + str(num) +  ' into 1 in ' + str(len(history)) + ' steps.'
print history
```

Figure 3.9 Screenshot of Sample Python Code Showing a Definition of a Function Based on Testing the Collatz Conjecture and Execution of that Function

Produced by the author as homework assignment for the course titled "Programming and Logic" given by Prof. Dr. Cem Bozşahin and held in Fall term of 2014 in Middle East Technical University.

3.2.3. Correspondence of Perceptual and Conceptual Structures

Instead of replacing visual logic for a new relational logic, an alternative axis must depart from understanding of critical relationships across perceptual structures and deeper conceptual structures. Late post-structuralist tendencies have progressively hidden conceptual structures in favor of perceptual structures rather than focusing on syntactical organizational problems that investigate alternative displacements of disciplinary fundamentals. Disciplinary fundamentals of architecture, including both representational structures and syntactical structures that organize space, must be acknowledged and displaced.[142]

[142] Lorenzo-Eiroa, op. cit., p11.

Pablo Lorenzo-Eiroa mentions the critical relationship between conceptual and perceptual structures. He argues that relational and visual logic have to coexist in design, instead of the relational being replaced by the visual: In the context of expressions, the question is, to what extent architecture is dependent on perceptual structures and in what ways do the perceptual and conceptual structures relate in this relational model of conceiving architecture.[143]

As discussed earlier in terms of software interfaces, it is possible to change the representation of information independently from the underlying working mechanisms and data flows in softwares. Representations or views are regarded as interfaces of the software with its user and can be changed independently from its information-based structures. Software design is involved with interactions, transactions, or in more generic terms, a relational structure which are regulated by procedures and information. It is possible to define structures for control and information without the necessity for a representation or a perceptual structure.

It is not essential of softwares to inhabit visual structures therefore their use is independent from, but only related to their visual representations and form. In contrast, architecture heavily relies on perceptual structures and image:

> Architecture finds itself in a unique situation: it is the only discipline that, by definition, combines concept and experience, image and use, image and structure. Philosophers can write, mathematicians can develop virtual spaces, but architects are the only ones who are the prisoners of that hybrid art, where the image hardly ever exists without combined activity.[144]

When it is considered that both software and architecture can be constituted by the code, or in broader terms, by relational structures, the problematic presented by

---

[143] Lorenzo-Eiroa, op. cit., p10-21.

[144] Tschumi, Bernard. "Responding to the Question of Complexity." Complexity: Architecture, Art, Philosophy. Journal of Philosophy and the Visual Arts. Ed. Andrew E. Benjamin. No. 6. London: Academy Group, 1995. p82.

such relational information models is that of the degree of dependency on perceptual structures and form in architectural use: Is it possible in architecture that a full relational model be provided which is dependent on conceptual structures, and independent from the perceptual?

Architecture relies on image and form in combining image and use or combining conceptual and perceptual structures. Architectural information is based on physical interactions instead of the abstract and digital interactions or transactions as exhibited in softwares. Perceptual structures are as prominent in architecture as conceptualizations or use cases exist with combined image, namely, there is no realization of form without visualization or perceptual structures. Even so, the code written for architectural solutions mandates implications of geometry, thus of image and perceptual structures. Furthermore, the importance of visualization is emphasized in computational problems that are not essentially visual.[145] Architectural solutions involve the duality of visual and relational reasoning. Therefore, both conceptual or perceptual structures are related with different relevance across modes of expressions as discursive devices, whether visual or mathematical/textual.[146]

This chapter presented issues in the structuring of information by representational structures as a way to establish the dialogue between conceptualization and information in the process of conceiving architectural form. There is a diversity of expressions which have different characteristics in terms of expressiveness. It is also argued that the mode of structuring is independent from information, but instead it is an aspect which contributes to meaning, interpretability and instrumentalization of design intention.

[145] Mike Bostock, a developer, mentions the importance of visualization of algorithms through illustrations and animations. See: Bostock, Mike. "Visualizing Algorithms." Visualizing Algorithms. Web. 10 Aug. 2015. <http://bost.ocks.org/mike/algorithms/>.

[146] Lorenzo-Eiroa, op. cit., p10-21.

Differences and similarities are discussed in terms of representing conceptual and perceptual structures across visual and mathematical/textual expressions, and in this context, correspondence between perceptual and conceptual structures in design are investigated. In this sense, it is argued that a direct relation to form is involved with specificity in perceptual structures, therefore fixates form to a final state. This situation is exhibited in both visual and textual/mathematical modes of expression.

Non-isomorphic expressions, or expressions which refer to conceptual structures instead of the perceptual are involved with the search for fundamentals of design and are disinterested in their realization as architectural form. They describe the conditions of solutions instead of solution forms. Specificity or predetermination of form as in the case of the orthographic set or computerized methods are deterred to provide for wide possibility spaces. Instead specificity in the conditions which produce form is preferred. Hence, non-isomorphic expressions are concerned with the inherent relationships in form, or in other terms, the conditions which make form, instead of its specifics in the form of geometry.

Abstraction from details and specificity of form is the basis for non-isomorphic expressions which describe the fundamentals about the design of conditions. An abstract machine which is independent from implementation details can be converted into any concrete structure with any type of implementation, still providing for the necessary mechanism to work.[147] This paradigm on the design of conditions instead of its concretized implications provides for broad possibility spaces that are constituted by multi-layered information and design intention.

Architecture is in possession of custom languages which are efficient in representing the complexity of custom information models, introducing multiplicity to form or providing for conceptual and perceptual structures of design. Further discussion in the next chapter, is based on the questions of what specific properties do such languages internalize in common, or what properties help the management

[147] De Landa, op. cit., p30-34.

of complexity of custom information models while providing for broader possibility spaces in expressing design intention.

CHAPTER 4

NOTIONS OF FORM-BLINDNESS AND SOFTFORM

Contemporary architectural problems are endued by an increased amount of information where increase in relevant information brings out a problematic in terms of management of complexity and workability. As Mennan states, architecture is now in possession of tools that are efficient in managing complexity of the contemporary architectural problems at the methodology level.[148] By methodological advances, it is possible to provide complex information models that are induced by multi-layered concerns besides introducing variation to form. But, conception of such models in design exhibits the problematic of managing complexity and introducing multiplicity to form, not only at methodological levels, but also at the design conceptualization level.[149] For the conceptualization level, expressions present an important discussion in defining the boundaries of possibility spaces.

4.1. Broadening Possibility Spaces: Form-blindness

Expressions are agents in establishing the dialogue between information and design conceptualization. The relevance of complexity and multiplicity in this context is based on many aspects concerning design both as product and process, but mainly on aspects that are relevant to architectural form, relations inherent to design, and information: Expressions are descriptions of design intentions that are induced by this frame of complexity. It requires elaborative approaches to create descriptions

[148] Mennan 2014, op. cit., p33-42.

[149] Kostas Terzidis mentions the increased ability to address complex problems by using computational methodologies based on extension of human intellect by calculation capabilities of digital systems. See: Terzidis 2008, op. cit., p75-86.

of the complex, ensure workability and manage complexity, and in this case, an approach on abstraction is exhibited in expressions which aim to broaden possibility spaces and manage the complexity of the design problem.

4.1.1. Abstraction

There are multiple ways to understand abstraction both in terms of design and expressions. For instance, Andrea Gleininer argues that abstraction is a means to eliminate complexity and provide “strategies for simplifying systematization” in Modernism.[150] Similarly for Mennan, design in the last century is concerned with simplicity and simplification both in terms of architectural form and phenomenological qualities such as meaning.[151] This paradigm which is based on an elimination of complexity through abstraction is interested in formalization, rationalization, systematization, and therefore, results in an irreversible reduction and simplification in both form and organization.

> A model, by definition, is always an abstraction of reality. Building a model means reducing the infinite complexity of the real world to a level where it can be described with manageable effort. What is obvious in the workshop of a model builder sometimes gets forgotten when almost infinite digital storage space is at hand: a perfect model does not contain as much information possible, but as little as necessary to describe the properties of an object unambiguously. Any extra bit would be meaningless for the given purpose and only impede comprehensibility.[152]

---

[150] Andrea Gleiniger provides a discussion on notions of complexity and abstraction in the contemporary context by providing an understanding of these notions in contemporary context, and in the contexts of Modernism and Post-Modernism. Gleiniger states a re-establishment of complexity in the contemporary architectural design and mentions the contrasting elimination of complexity through abstraction in Modernism. Gleiniger claims Robert Venturi’s “Complexity and Contradiction in Architecture” as a point of departure. See: Gleiniger, Andrea. ""The Difficult Whole," or the (Re)discovery of Complexity in Architecture." Complexity, Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008. P37-57.

[151] Mennan 2014, op. cit., p33-42.

[152] Scheurer, Fabian, and Hanno Stehling. "Lost in Parameter Space?" Mathematics of Space. Ed. George Legendre. London: Wiley, 2011. P72.

There is a distinction between the elimination of complexity and elaborate abstraction over complexity. Abstraction in the case of expressions which are pertinent to custom information models is not based on a paradigm where meaning and context are segregated, and where architectural form and organization are simplified. In this context of expressions, the notion of abstraction relies on the balance between the elimination of inessential specificity in information and form, an aspect which narrows down possibility spaces and changeability, while providing complexity, relativity, multiplicity, extensibility and variability to the design process and results. Despite the fact that approaches or techniques may differ for abstractions in either visual or mathematical/textual expressions, essentially its definition in this context is essentially the same regardless of the medium: An elaborate separation of fundamentals from implementation details.

Abstraction from specificity renders expressions more comprehensible and workable where only essentials about the subject are described, as in the case of abstract machines[153]; thus providing an extensible and manageable basis where any redundant information that hinders design is being hidden or eliminated.[154] An example to this situation would be the bull lithographs by Pablo Picasso [Figure 4.1]. These lithographs display a range of depictions of bulls from actual portrayals which are characterized by renderings and details which emphasize identity, to abstract representations that are defined by a limited number of elements which provide for a more generic definition. It is possible to see the distinguishing aspects of a bull in all of these depictions where abstract representations are closer to a "shortest possible description"[155] of a bull.

---

[153] Sudkamp, op. cit., p157.

[154] Dogan and Nersessian 2010, op. cit.

[155] Authors mention the definition of 'Kolmogorov complexity' or 'descriptive complexity' in the context of information theory as "the complexity of an object is defined by the length of the shortest possible description". See: Scheurer and Stehling, op. cit. Additionally, for further information about Kolmogorov complexity see:

"Algorithmic Information Theory." - Scholarpedia. Web. 29 July 2015. <http://www.scholarpedia.org/article/Algorithmic_information_theory#Algorithmic_.22Kolmogorov.22_Complexity_.28AC.29>.

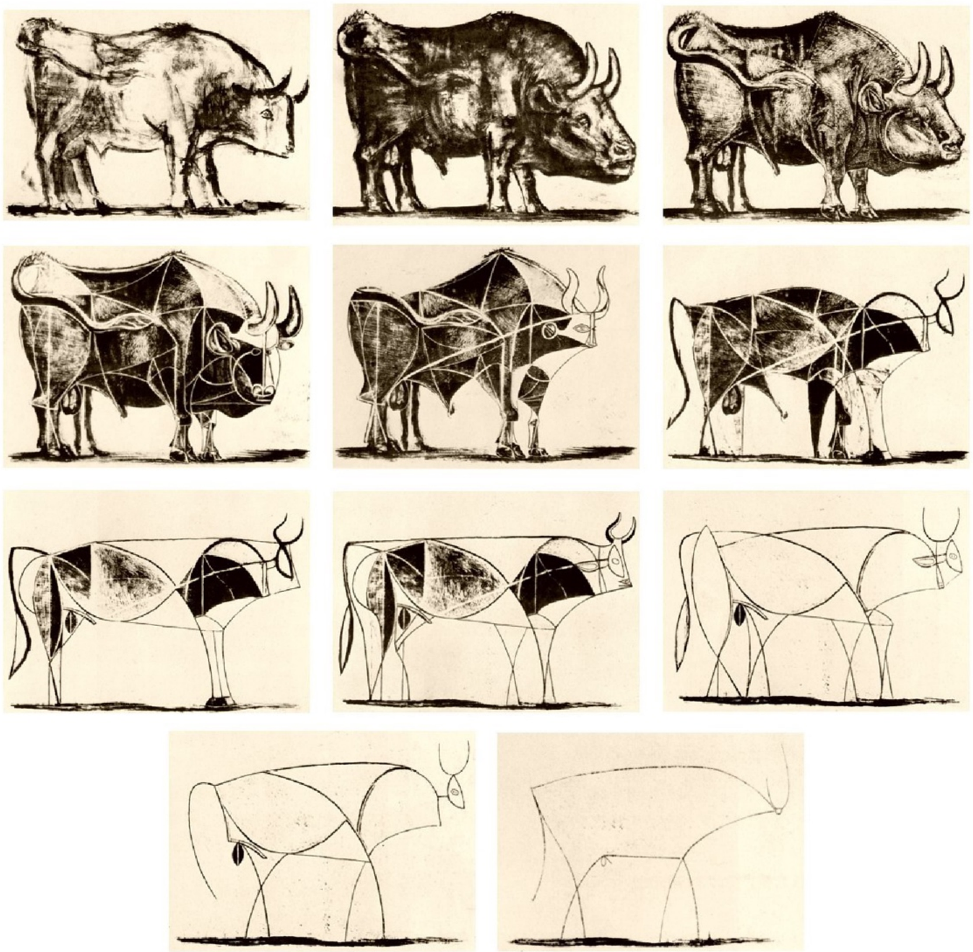


Figure 4.1 Different Depictions of a Bull by Pablo Picasso

Produced by author with images of lithographs by Pablo Picasso taken from: "Pablo Picasso - Bull: A Master Class in Abstraction." Web. 21 July 2015. <http://artyfactory.com/art_appreciation/animals_in_art/pablo_picasso.htm>.

Another relevant aspect of abstraction in the context of expressions is of emphasis: Abstract expressions emphasize different aspects about a description.[156] This situation is exhibited in the differentiation of modes of structuring expressions that are discussed in the previous chapter, where each mode of structuring has different qualities in terms of emphasis, expressiveness and conveying of information. Kostas Terzidis notes that:

[156] Dogan and Nersessian 2010, op. cit.

> Another model of depiction is that of abstraction: black-and-while line drawings convey a clear and, sharp, and sterile impression of the depicted form, whereas blueprints are understood as working drawings. In contrast, rendered drawings convey materiality, completeness, substance, and effects. The problem with rendered views is that form is not always conceived as made out of matter. In fact, form is rather an abstract entity that possesses certain geometric characteristics. […] The attachment of material qualities constrains the behavior of form and restricts the designer's imagination. In contrast, the lack of materiality liberates the form from its constraints and introduces behaviors closer to intuition rather than perception.[157]

Similar to what is seen in the case of Picasso's bulls, Kostas Terzidis illustrates a visual mode of abstraction that is based on the elimination of visual specificity in the description of form. All of the types of depictions that are mentioned by Terzidis have a different purpose in the design process where the more specific is the more constraining and restraining for designers and where visual or perceptual specificity emphasize perceptual structures rather than the conceptual.[158] Another example to this case would be a comparison between a model and a conceptual drawing where the former describes the conditions which produce form, and where the latter is a physical abstraction of form [Figure 4.2].

[157] Terzidis 2003, op. cit., p57-64.

[158] Terzidis 2003, op. cit., p57-64.

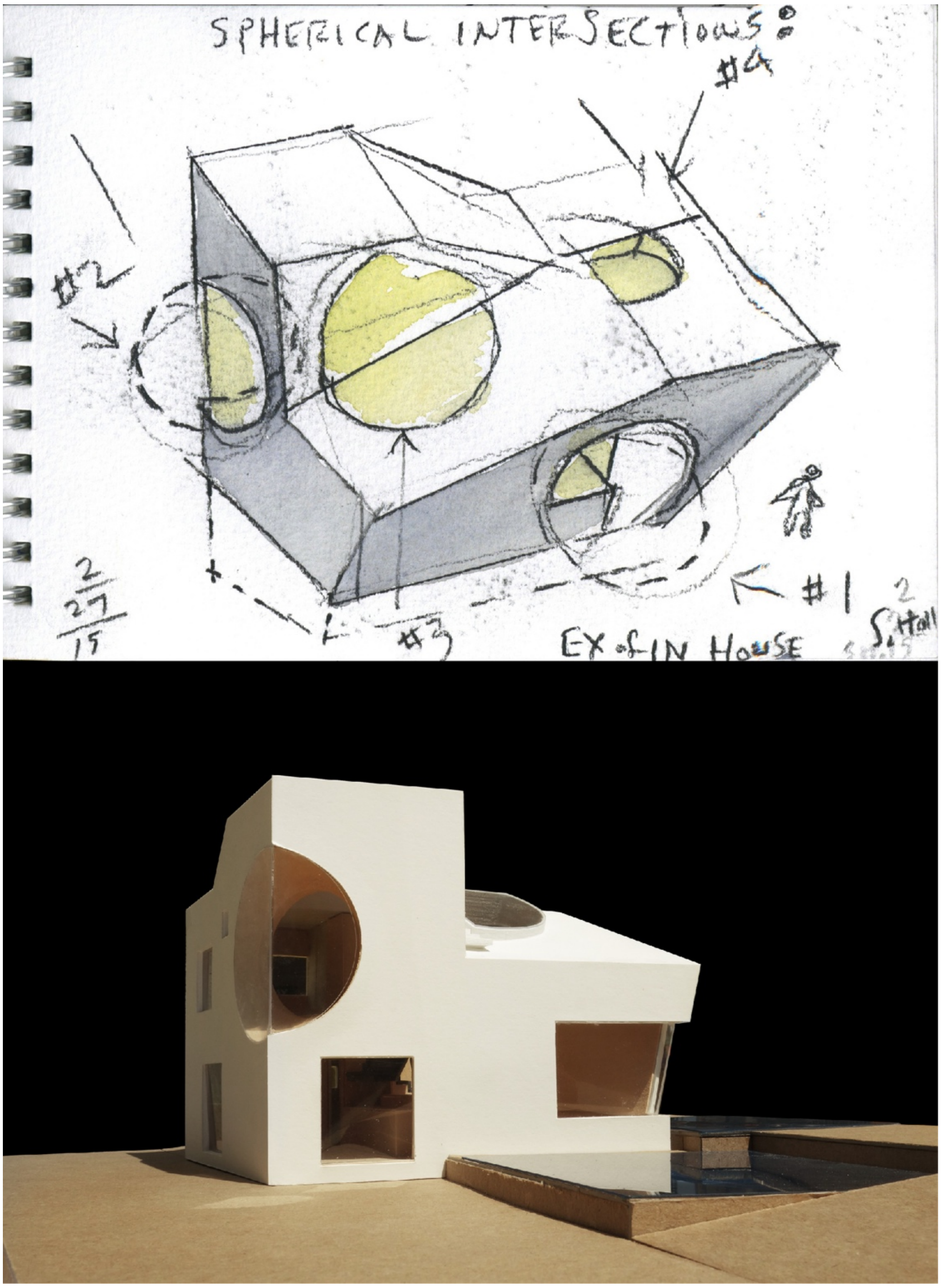


Figure 4.2 Comparison of a Conceptual Drawing and the Model of "Ex of In" House by Steven Holl Architects

Produced by author with images taken from: "Steven Holl Architects Breaks Ground on the “Ex of In" House in New York." ArchDaily. 21 July 2015. Web. 21 July 2015. <http://www.archdaily.com/770583/steven-holl-breaks-ground-on-ex-in-house-in-new-york>.

In comparison to the visual, relational models have a different mode of managing complexity through abstraction. Modern software development is involved with principles that constitute best practice in terms of scale and extensibility.[159] These principles dwell on practices such as the encapsulation of a similar functionality and information into distinguishable entities[160]; decoupling or loosely coupling of such entities in software, which ensures interchangeability and extensibility[161]; information hiding from, whether local or global, other entities to mention a few[162].[163] Such practices are means of controlling and increasing the level of complexity where the number of relations and interactions between entities in software are limited and where inessential information is hidden from any entity which does not need it.

Depending on the programmer, there may be different layers of abstraction in software architecture, where each abstraction layer hides the implementation details of a particular set of functionality.[164] These layers of abstraction are diverse on a

[159] There are five basic principles of object-oriented programming and design named by Robert C. Martin, an influential software engineer, when applied together constitutes maintainability and extensibility of a software. These five principles are named as single responsibility principle, open/closed principle, Liskov substitution principle, interface segregation principle and dependency inversion principle. For further information and illustrations about these principles see: "S.O.L.I.D: The First 5 Principles of Object Oriented Design." Scotch. Web. 29 July 2015. <https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>.

[160] Microsoft, ” Chapter 2: Key Principles of Software Architecture." Web. 11 Aug. 2015. <https://msdn.microsoft.com/en-us/library/ee658124.aspx>.

[161] Microsoft, ” Chapter 2: Key Principles of Software Architecture." Web. 11 Aug. 2015. <https://msdn.microsoft.com/en-us/library/ee658124.aspx>.

[162] "Introduction to Object Oriented Programming Concepts (OOP) and More." CodeProject. Web. 11 Aug. 2015. <http://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep#Encapsulation>.

[163] These practices are based on the principles of the paradigm of object-oriented programming. Object-oriented programming is a methodology of programming based on structuring of the software architecture based on the concept of ‘objects’ which contain data and possess procedures that modify these data. A programming paradigm which focuses on the relations and interactions of different types of objects. See: "Introduction to Object Oriented Programming Concepts (OOP) and More." CodeProject. Web. 11 Aug. 2015. <http://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep#OOP>.

[164] For a description of the notion of abstraction in computer science see: Ullman, Jeffrey D. "Computer Science: The Mechanization of Abstraction." *Fundamental Concepts of Programming Systems*. Reading, Mass.: Addison-Wesley Pub., 1976. P1-23.

scale from the more relational to the more operational; from relation-based systematic conceptual schemas to the more numerical, operational, geometrical definitions, and finally to the compiled and concrete binary string that is executed by an electronic machine. In the more abstract levels of the software, it is possible to make the code 'meaningful' in terms of expression. The code takes on a more narrative and humanly intelligible form[165] in contrast to a more mathematical and operational syntax; a case which is possible to observe in the encapsulation of functionalities [Figure 4.3 and Figure 4.4]. Each level of abstraction limits explicit interactivity of entities through controlled interfaces and hides implicit mechanics, thus controlling complexity of entities and their interactions. This is best practice for software architecture where specificity and complexity of operational layers of the software are hidden and the focus is on the relations inherent to the software as a relational structure. This case in software development is similar to that of visual expressions as well when considering Doğan and Nersessian's discussion on "generic abstraction" -which is used in model-based reasoning- is prominent for generating and integrating constraints, and furthermore delimiting interfaces.[166]

[165] Code readability and expressiveness, or meaning is often linked to the notion of 'syntactic sugar'. See: Landin, Peter J. "The Mechanical Evaluation of Expressions." *The Computer Journal* (1964): 308-20. Web. 3 Sept. 2015. <http://www.cs.cmu.edu/~crary/819-f09/Landin64.pdf>.

[166] Dogan and Nersessian 2010, op. cit.

```python
# Summation without encapsulation of functionality
a = 5
b = 10

print(a + b) # prints the total of two variables

# This methodology, requires manual coding of summation for each time
# It reads as "Print the result of a plus b"


# Definition of a summation function
def sum(a, b):
    return a + b

print(sum(13, 37))

# This methodology prints the result of given two numbers
# to the screen through the definition of sum() function

# It reads as "Print the sum of 13 and 37."
```

Figure 4.3 Encapsulation of Functionality for Summation in Python Programming Language

Produced by the author.

```python
# import the rhinoscript library; a library for manipulating
# geometry through the Rhinoceros modelling software
# which provides an abstraction layer to Rhinoceros'
# mathematical/operational geometry manipulation functionality
import rhinoscript as rs

# Definition of three points by x,y,z coordinates
a = [0, 0, 0]
b = [10, 0, 0]
c = [0, 10, 0]

# Manually creates a triangle using three pre-defined points
line1 = rs.AddLine(a, b)
line2 = rs.AddLine(b, c)
line3 = rs.AddLine(c, a)

triangle = rs.JoinCurves([line1, line2, line3])


# Definition of a triangle creation function
def createTriangle(d, e, f):
    l1 = rs.AddLine(d, e)
    l2 = rs.AddLine(e, f)
    l3 = rs.AddLine(f, d)

    return triangle = rs.JoinCurves([l1, l2, l3])

# This function encapsulates the operation creating a triangle
# and can be used as below

createTriangle([0, 0, 0], [-10, 0, 0], [0, 10, 0])
# Creates a triangle symmetrical to the first one
```

Figure 4.4 Encapsulation of Functionality for Drawing a Triangle in Python Programming Language

Produced by the author.

Hiding or eliminating specificity entails a shift in the character of the expression in a scale from the perceptual to the conceptual. Similar to conceptual diagrams that are used in architectural design, diagrams which visualize interactions and relations between entities in a software[167] are utilized to conceive, comprehend and optimize the relational structure of the software [Figure 4.5, Figure 4.6 and Figure 4.7]. These diagrams are involved with defining 'meaningful' entities and relations, thus providing a comprehensible conceptual structure for the software to manage its inherent complexity.

[167] An example to notational systems that are used in the diagrams that model software entities and behavior is UML (Unified Modelling Language). See: "Unified Modeling Language™ (UML®) Resource Page." Unified Modeling Language (UML). Web. 26 July 2015. <http://www.uml.org>.

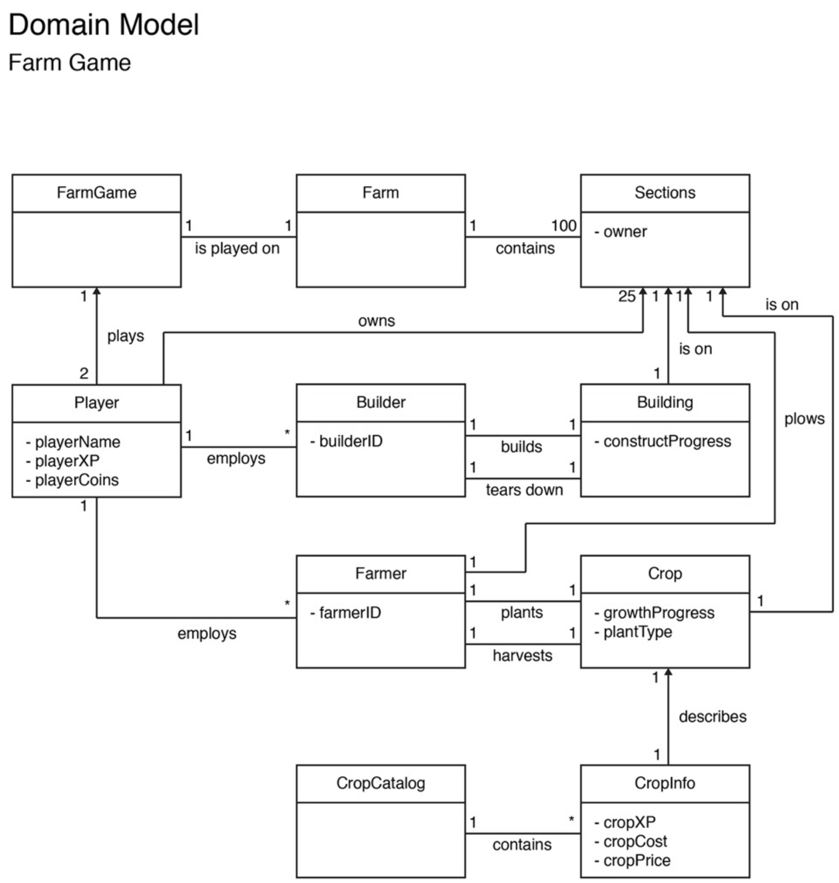


Figure 4.5 Domain Model for the Farm Game Showing Different Objects with Their Attributes and Methods

Domain Models are conceptual models used to model the entities, their properties and their in-between relations in a software. Its description is based on the information or data layers of software. Produced by author as part of an assignment of developing a game for the course 'Object-oriented Analysis and Design' taken in 2013-2014 Fall Term and given by Assoc. Prof. Dr. Altan Koçyiğit

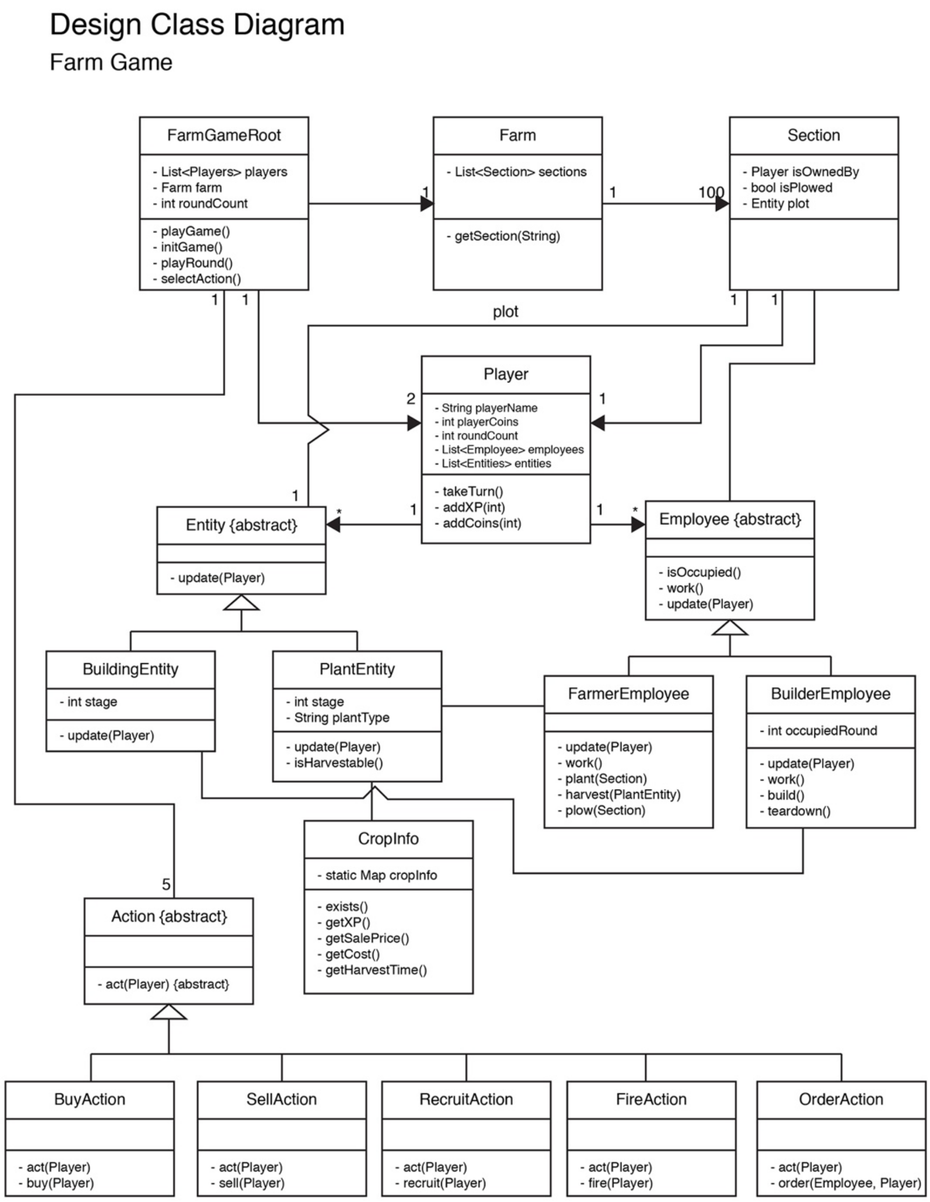


Figure 4.6 Design Class Diagram for the Farm Game Showing Relations of Different Objects in the Design of Farm Game

Design class diagrams are similar to domain models but instead of describing information or data layers, they are used to model the logic layers of softwares. Produced by author as part of an assignment of developing a game for the course ‘Object-oriented Analysis and Design’ taken in 2013-2014 Fall Term and given by Assoc. Prof. Dr. Altan Koçyiğit

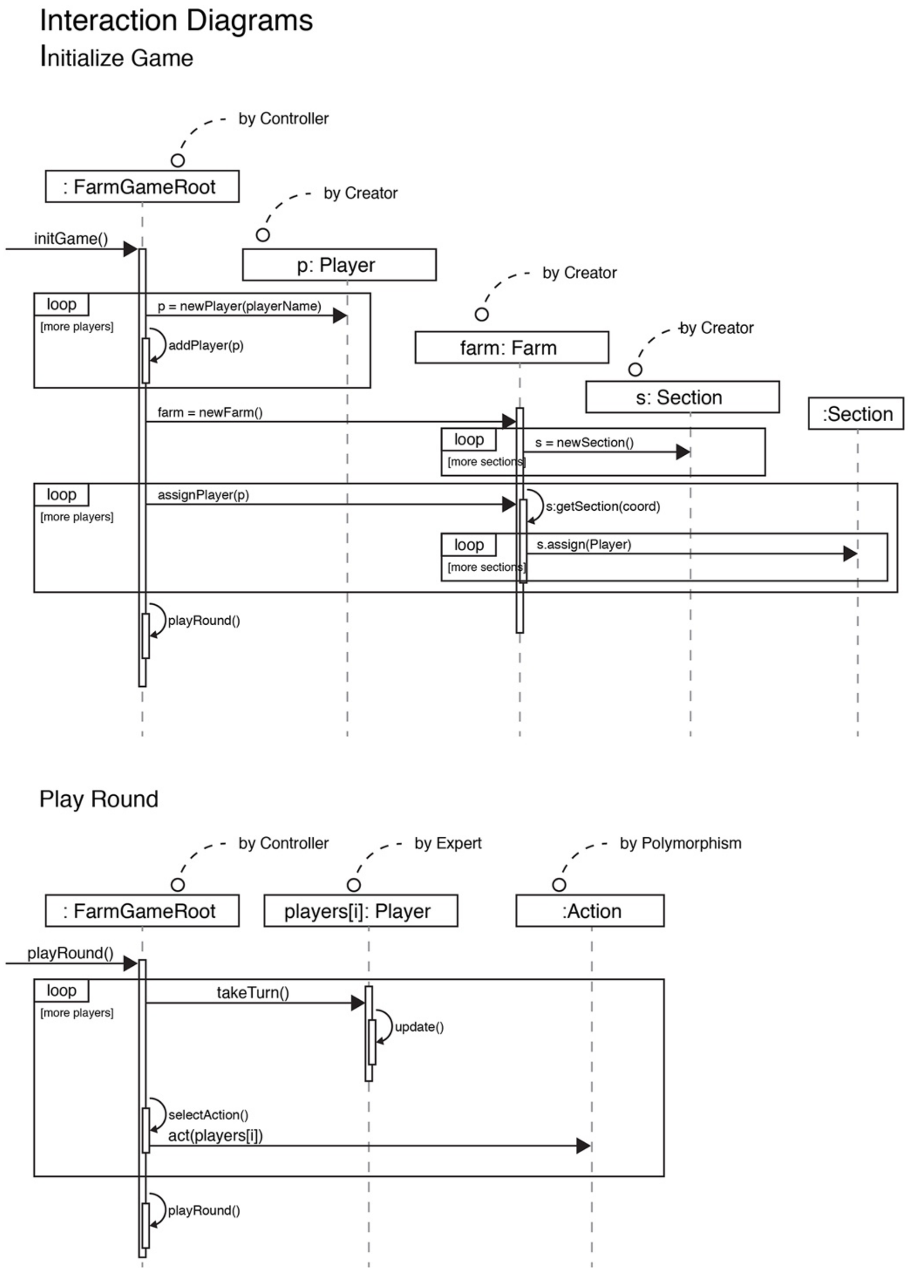


Figure 4.7 Interaction Diagrams for the Farm Game Showing Design Patterns and Interactions Between Entities

Interaction diagrams are used to describe or visualize the specifics of a process in the software. Produced by author as part of an assignment of developing a game for the course 'Object-oriented Analysis and Design' taken in 2013-2014 Fall Term and given by Assoc. Prof. Dr. Altan Koçyiğit

Such diagrams can again be considered similar to conceptual diagrams or to Van Schaik's Ideograms previously referred [Figure 4.8], as a way to express conceptual structures. Elimination of specificity infers a focus on relational and conceptual structures that are devoid of implementation details as exhibited in the case of non-isomorphic expressions. For Marcos Novak, there is a "transition into a new era of manipulating knowledge using high level abstractions and nested relations, rather than particular objects".[168] Instead of focusing on a single implementation of a model as the final object, a relational model of design is composed of multiple objects, events, relations and reference systems[169] and presents abstract conceptual structures which are independent from perceptual structures: Abstract descriptions of design where relational and conceptual structures are represented instead of their implementation, or more specifically, of a single concrete instance of architectural form. Ideograms that are produced by Leon van Schaik are examples of such abstract expressions where implications for form disappear, but instead conceptual descriptions are depicted [Figure 4.8].

[168] Novak, op. cit., p15.

[169] Marcos Novak lists a range of designed artifacts which are named as designata, and together which produce a working model of designata: objects, events, relations and reference systems. In the context of this study these designata are understood as primitives for relational models and structures. See: Novak, op. cit., p15.

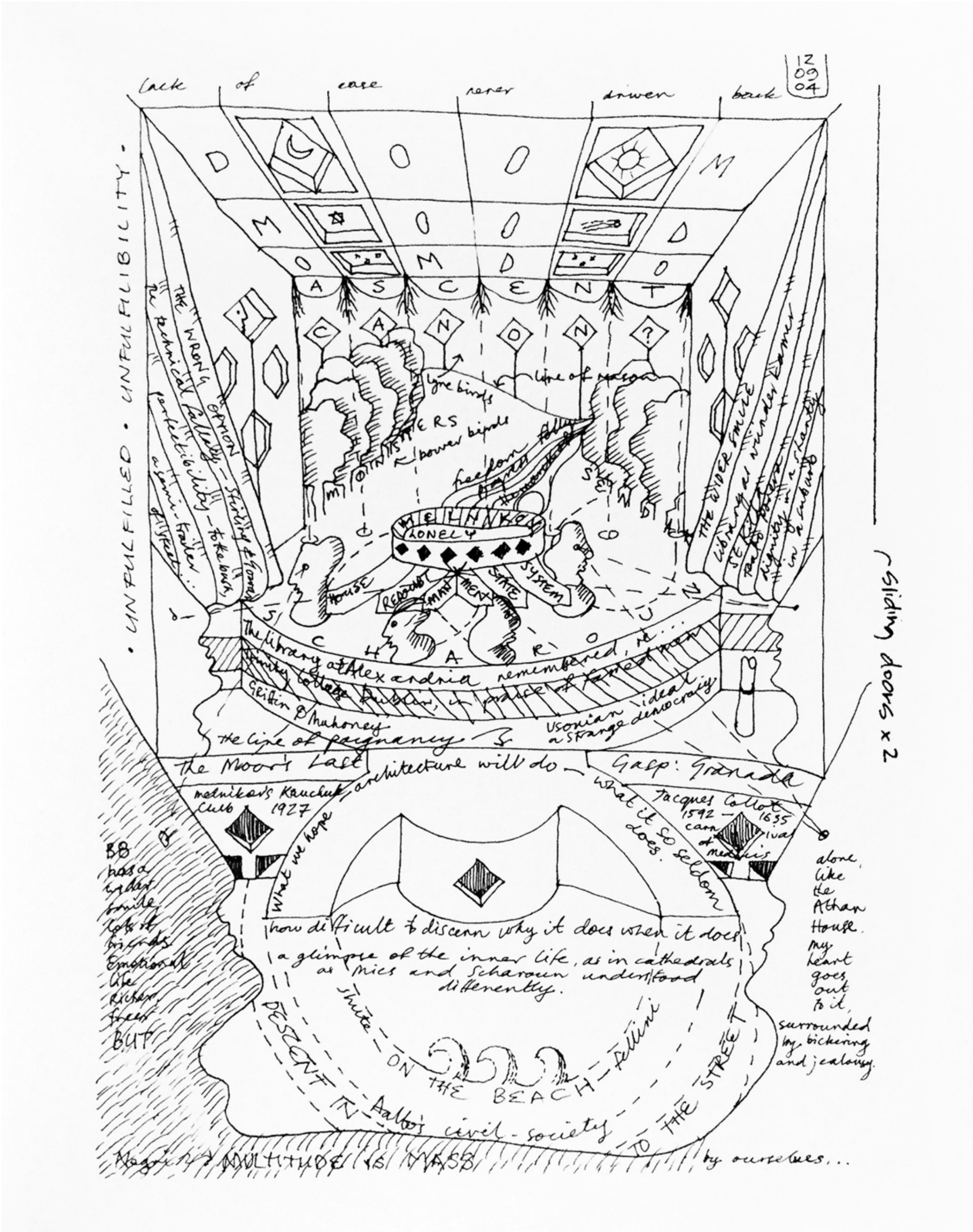


Figure 4.8 Edmond & Corrigan Ideogram by Leon van Schaik

Source Van Schaik, Leon. Ideograms. Melbourne: Lyon Housemuseum, 2013. p31.

Whether structured in visual or mathematical syntax, a degree of abstraction is necessary in expressions both to broaden possibility spaces and encourage the utilization of complex multi-layered considerations in design. Namely, abstraction from inessential specificity, especially in terms of implementation as form, presents a freedom for architectural form, visualization and perceptual structures, besides a level of manageable complexity in the relational structures of design. Hence, an independence from perceptual structures or architectural form provides the basis for extending the conceptual structures. This situation can be named as a 'blindness' towards form, one which renders important in liberating both perceptual and conceptual structures and providing manageability, workability, comprehensibility and extensibility to design conceptualizations.

4.1.2. Form-blindness and Degree of Specificity

Form-blindness[170] is defined here as a quality which is observed in expressions that are used to manage the complexity of both visual and relational structures through abstraction over inessential specificity (mainly about form). It indicates an approach for the control over information and specificity about design, to the double aim of liberating conceptual structures from the perceptual and manage or extend complexity. In its basic implication, form-blindness can be understood as the relative degree of knowledge about form desired to be maintained at a given time during the design process.

It is essential here to mention once more Juhani Pallasmaa's description of the architectural design process: Architectural ideas arise from diffuse images and immaterial ideas, which are refined and specified over time to become functioning utilitarian structures.[171] In this progressive approach of architectural design, less of the form is specified at the beginning of the design process. Architectural form is resolved progressively until it becomes fully specified and made into built form. During this process of form resolution, form-blindness is reduced progressively while the possibility space narrows down and form loses the flexibility to change. On each progression level as the image of form becomes more specific, the level of abstraction over form is reduced where the scale of form-blindness shifts from the more abstract and conceptual, to the more specific and perceptual.

[170] Two factors that are observed in expressions have led to the development of the notion of form-blindness: Changing degrees of specificity about form as exhibited in different modes of expressions (especially the visual), and the best practice in software development which are used to control or limit interfaces of interactions and information to establish extensibility and maintainability. Therefore form-blindness is based characteristics that are exhibited on both mathematical/textual and visual modes of expression.

[171] Pallasmaa, op. cit.

Figure 4.9 Sketch of BMW Welt

Source "BMW Welt." Coop Himmelb(l)au. Web. 23 July 2015. <http://www.coop-himmelblau.at/architecture/projects/bmw-welt/>.

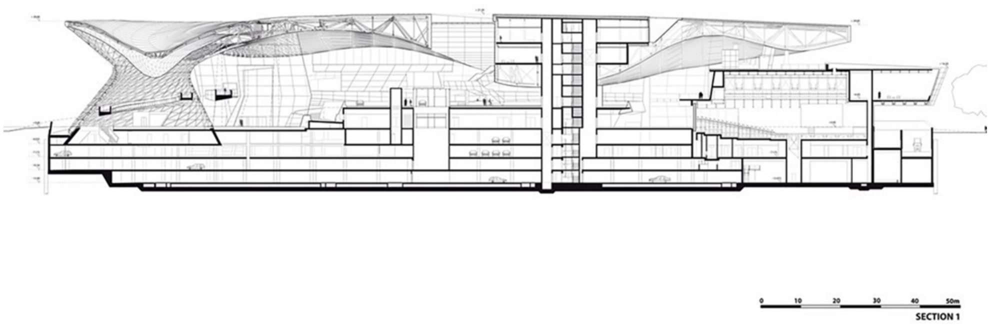


Figure 4.10 Longitudinal Section of BMW Welt

Source "BMW Welt." OpenBuildings. Web. 23 July 2015. <http://openbuildings.com/buildings/bmw-welt-profile-2506/media>.

It can be argued that every expression is constituted by a degree of form-blindness; one which refers to specificity, level of detail and determinedness regarding form [Figure 4.9 and Figure 4.10]. The degree of blindness is decided at the stage of design, depending on factors mainly pertinent to the aspects regarding changeability of form and specificity of form. Absolute blindness is only possible in expressions which describe only conceptual structures of design, such as conceptual diagrams or ideograms; but expressions that describe perceptual structures of design exhibit a lesser degree of blindness.

Form-blindness is a dissociation from inessential specifics, or in simpler terms, from unnecessary knowledge in order to increase the workability, changeability and flexibility of form and its internal conditions, as full intricacy of perceptual structures would undermine descriptions of multi-layered information models due to increased complexity. Consequently, some modes of expression are more form-blind, such as conceptual diagrams which, by their abstract character display conceptual structures and are devoid of specificity concerning implementation or realization of form.

The visual and mathematical/textual structuring of expressions present different characteristics of abstraction, and consequently, different methods for eliminating specificity. For visual structuring, form-blindness is related both to the level of detail in expression and the degree of correspondence with perceptual structures. For the more conceptual types of visual expressions, such as conceptual diagrams, the degree of form-blindness is higher in contrast to expressions which are in direct correspondence with the built form such as sections or plans. Furthermore, when considering isomorphic expressions as plans or sections, form-blindness decreases over form in detail or construction drawings where full scale details are depicted. From diagrams to sketches, from sketches to working drawings and from working drawings to detailed construction drawings, there is a scale in visual expressions where description of form becomes more specified and concretized, and where form-blindness is gradually reduced. Each mode of visual expression has a different level of specificity and determination about form and design intention where the more detailed corresponds to the less ‘blind’.

In contrast to a visual resolution of form, a process which is regulated by top-down gradual specification of design intention as a visual whole, relational models and mathematical/textual expressions have a different mode of form-blindness. The code is a medium which manipulates geometries through operations; a medium that defines sequences of form-generation from building blocks into relational structures. Therefore, definitions and sequences of geometrical operations become more relevant for relational models rather than perceptual structures. The code, as a system for form production may be understood as a mode of collective decision making: The conceptual and relational structures of the mathematical/textual expression define logics of form generation from geometrical operations where intentions of the designer exist in multiple locales and levels. Code is form-blind in perceptual terms since there is no depiction of form in a visual form, but only a definition of the conditions which produce form.

Therefore, mathematical/textual expressions describe a bottom-up sequence of formation that arise from geometrical operations and their relations, in contrast to top-down resolution and specification of form as exhibited in visual expressions. Therefore, what is becoming more specified in mathematical/textual expressions, in contrast to visual expressions, are relations that are inherent to the logic of formation instead of perceptual structures of geometry: Form-blindness decreases in code as relations become more specified.

Either for visual and mathematical/textual expressions, form-blindness shifts the emphasis of expression to the inherent relations of form instead of the realized form itself. Therefore, form-blindness necessarily means that the designer should be less occupied with the results of the relational model but rather with the relational model itself. The most abstract form of form-blindness implies a paradigm of design without seeing: A mode of abstraction from form as geometry, thus of perceptual structures, emphasizing the comprehensibility and manageability of the conditions that produce such structures.

Another beneficial aspect of form-blindness is that, an abstract relational structure may be succeeded with a multiplicity of perceptual structures that follows its rules, while this is a matter of correspondence of perceptual structures to conceptual structures. This correspondence is exemplified between the conceptual diagram describing the circulation conditions of the Yokohama International Passenger Terminal by Foreign Office Architects [Figure 3.3] and the final built form of the Terminal [Figure 4.11 and Figure 4.12]. There is no resemblance between the diagram and the built structure in terms of form but the logic described in the diagram is applied to the design of the building, therefore built structure is liberated from any constraints while adhering to a conceptual structure. In this sense, Manuel De Landa mentions Gilles Deleuze's definition of the process of "divergent actualization": The ability of topological forms (and other abstract machines) to give rise to many different physical instantiations.[172]

Figure 4.11 Yokohama International Passenger Terminal Aerial Photograph by Foreign Office Architects (FOA)

Source "Gallery - AD Classics: Yokohama International Passenger Terminal / Foreign Office Architects (FOA) - 1." ArchDaily. Web. 23 July 2015. <http://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa/54349375c07a80110e000037>.

[172] De Landa, op. cit., p30-34.

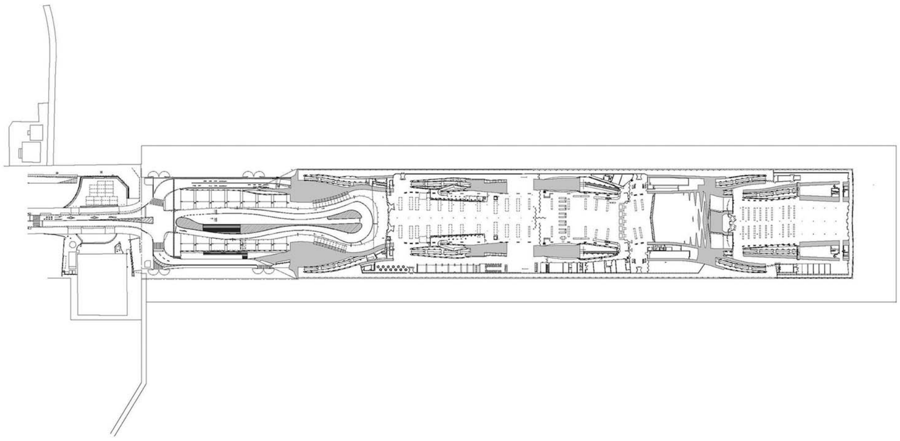


Figure 4.12 Yokohama International Passenger Terminal Plan Drawing by Foreign Office Architects

Source "Gallery - AD Classics: Yokohama International Passenger Terminal / Foreign Office Architects (FOA) - 18." ArchDaily. Web. 23 July 2015. <http://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa/542078d8c07a800de5000008>.

Form-blindness enables multiplicity and proliferation of considerations in design intention by shifting the focus from an aim for realized results to multi-layered information models which produce results, and consequently encourages designs arising from custom and intricate logic with increased complexity. Hence, from-blind relational models have an important place where the logic of producing architectural form is extended and diversified. In this case, the complexity of the model arises from the relational aspects and conceptual structures instead of the visual aspects and perceptual structures. Form-blind approaches through expressions provide conceptualizations of design which achieve broader possibility spaces for form by liberating perceptual structures from the conceptual, increase comprehensibility of design intention, and ensure manageability of complex relational models.

4.2. Encapsulating Possibility Spaces: Softform

The notion of form-blindness puts the emphasis on complexity management in design. Form-blindness as an approach provides for complex information models that incorporate a multiplicity of concerns and better integration of such concerns.

In contrast to benefits of increased complexity in design, describing a complex construct/system with exact means brings out a problematic in terms of defining the boundaries of form mutability when considering expressions. The contemporary context of architecture, necessitates ways to engage in diversity and coherence with complex models. In this sense, Greg Lynn mentions the necessity of formal flexibility as a means for embodying complexity in a single unified entity.[173]

> A plexus is a multilinear network of interweavings, intertwinings and intrications; for instance, of nerves and blood vessels. The complications of a plexus - what could best be called complexity - arise from its irreducibility to any single organization. A plexus describes a multiplicity of local connections within a single continuous system that remains open to new motions and fluctuations. Thus, a plexial event cannot occur at any discrete point. A multiply plexed system - a complex - cannot be reduced to mathematical exactitude, it must be described with rigorous probability.[174]

Lynn mentions two dominant approaches in incorporating differentiation; ‘conflict and contradiction’ or ‘unity and reconstruction’. Furthermore, in expressing the inadequacy of the prior two approaches, he proposes what is to be the third alternative in incorporating differentiation and complexity as ‘smoothness’. For Lynn, it is necessary to provide descriptions of the complex with a probabilistic approach: An approach which enables embodiment of multitudes of different concerns, a heterogeneity, in unified entities through descriptions which are characterized by ‘smoothness’ and ‘flexibility’.[175] Lynn’s argument on ‘smoothness’ is based on the notion of ‘smooth’ that is developed by Deleuze and Guattari.[176]

---

[173] Lynn, Greg. “Architectural Curvilinearity: The Folded, the Pliant and the Supple”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p29-44.

[174] Ibid. p36-37.

[175] Ibid. p29-44.

[176] Greg Lynn provides Deleuze and Guattari’s description of smoothness as the ‘continuous variation’ and the ‘continuous development of form’. See: Lynn, op. cit., p30.

The notion of 'smooth' describes the "continuous variation" and "continuous development of form" through integration of diverse and distinct elements with different intensities; where integration of these elements are not constituted by connections but rather negotiations.[177] This negotiation factor is similar to the notion of 'looseness' as developed by Stan Allen.[178] 'Looseness' denotes integrity and internal coherence in a network of relations that are capable of accommodating differences through robust flexibilities or negotiations in complex systems: "Permeable boundaries, flexible internal relationships, multiple pathways and fluid hierarchies are the formal properties of such systems."[179]

In the contemporary context of complexity in architecture, computational methodologies have the capability to provide flexibility and variation to architectural form while maintaining complex orders.[180] Bernard Cache's Objectile[181] exemplifies this capability through his work where the aim is "to maximize the flexibility and variability available within the mode of production"[182]. Description of forms through the intention of the maximization of flexibility and variation represents broad possibility spaces that are regulated by complex orders.

[177] This description is developed through Greg Lynn's discussion on the notion of 'smoothness' that is argued by Deleuze and Guattari. Further discussion on the notion of 'negotiation' in terms of self-organization is presented through local-ness and global-ness later in this section. See: Lynn, op. cit.; Deleuze and Guattari, op. cit., p474-500.

[178] Allen, Stan. "From Object to Field", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p63-79.

[179] Ibid. p77.

[180] Mennan 2014, op. cit.

[181] Objectile is the name of the practice that is led by Bernard Cache. As explained by Bernard Cache, Objectile's aim is to "develop procedures, both software and hardware, that will make digital architecture a reality at an affordable cost for small architectural practices and average consumer". Objectile's work is particularly associated with Deleuze's illustration of "topology, the fold and planes of immanence". See: Cache, Bernard. "Philibert De L'Orme Pavilion: Towards an Associative Architecture.", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p152-157; Perella, Stephen. "Bernard Cache/Objectile: Topological Architecture and the Ambiguous Sign.", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p149-151.

[182] Perella, op. cit., p149.

Consequently, such forms exhibit abilities to respond and adapt to a multitude of conditions. Instead of exactitude and determination in form, flexibility, adaptability and variation become favored factors both in terms of perceptual and conceptual structures, or visual and relational models. On top of 'smoothness' and flexibility, which Greg Lynn characterizes with notions of suppleness and pliancy[183], this paradigm of a probabilistic approach to form infers the notion of 'softness' of form; describing the degree of flexibility and variability. Furthermore, 'softness' is a term that refers to the state of indetermination of probability of form and the ability of form to change without the need of extensive effort at a given stage of design. Similar to the condition of specificity that is discussed in the context of form-blindness, inessential specificity reduces form's capacity for flexibility in incorporating different concerns and providing variation.

In a linear resolution of form that is exhibited in a progression from diagrams to sketch and sketches to drawings, form loses its capability to change over time: Softness of form increases with form-blindness. In each iteration of the resolution, form eventually reaches to a single outcome, by losing its "energetic possibilities".[184] In this case of linear form resolution, form 'hardens' and becomes resistant to change. However, it is possible to overcome this deficiency by the aid of non-linear computational methodologies where the resolution of form can be automated and reproduced.

Accordingly to the duality of softness and hardness, an entity which is constituted by the indeterministic and probabilistic characteristic that is described by 'softness' is defined as 'softform': A singular virtual entity which embodies all the possible outcomes of the description of form and the design intention, in a single entity

---

[183] Lynn, op. cit.

[184] De Landa, op. cit., p30.

which can transform whenever and however necessary.[185] Its softness is associated with its degree of multiplicity and mutability in form. Namely, the term softform denotes virtual forms or virtualizations of design intentions with different degrees of multiplicities, which are non-concrete and not finalized and which are able to transform through regulation of multi-layered concerns, thus complex orders.

Figure 4.13 Dynamism of a Dog on a Leash (1912) by Giacomo Balla

Source "Great Works: Dynamism of A Dog on a Leash (1912) Giacomo Balla." The Independent. Independent Digital News and Media. Web. 25 July 2015. <http://www.independent.co.uk/arts-entertainment/art/great-works/great-works-dynamism-of-a-dog-on-a-leash-1912-giacomo-balla-1781174.html>.

[185] Contemporary issues about physicality and implementation (hardness), and software-like mutability (softness) are discussed in the context of computational design and manufacturing in a conference paper by the author. The paper illustrated that softness of software-like behavior is penetrating the material implementations of form in physical space, and similarly, materiality or physicality is entering the digital. Contemporary architectural design and its realization favors both changeability and flexibility in the physical space and specificity about the realization of form in the digital. This discussion has later led to the development of the notion of softform. See: Kızılcan, op. cit.

A softform is not a satisfactory, progressively achieved, static, final product, but rather a formal means of defining boundaries of variability and mutability for dynamic conditions. Instead, it is a virtual entity which describes what can be real, and it provides a range of permutations of form in an allowable range.[186] It is where best possible solutions that fit to the multi-layered concerns of the complex orders are searched and found. Therefore, softforms internalize the factor of indeterminacy regarding changeability of form: Rather than definitions that highlight determination and specificity about geometrical conditions of form, outlines of changeability based on internal relations are described. The painting titled "Dynamicism of Dog on A Leash" by Giacomo Balla [Figure 4.13] is an example of the visualization of softform entities. The body of the dog is subject to temporal change in terms of its form, in the way that is defined by its anatomical boundaries. Its form can change while the inherent relations that are defined by its anatomy are preserved. Its form may be in any shape and its limbs may be anywhere anytime in the boundary that is portrayed in the painting, where the painting does not include the information about the specifics of the dog's form, or where its limbs specifically are. Softforms defer specification and determination in geometric form and therefore they are results of definitions which describe or redefine the limits of control over form.

Softforms further embody all instances regarding the conditions and relations which produce form, including both the satisfactory and the unsatisfactory. They represent, not static and concrete *progress* products, but *process* products where form is heuristically developed through successive processes until satisfactory results are encountered.[187] According to Kostas Terzidis, this sort of heuristic development to eliminate uncertainties requires the ability of decision-making and

---

[186] De Landa, op. cit.

[187] AD profile No. 231 presents a number of studies on the relation of space and heuristic development of form through computational processes which allow variation and exploration, under the section 'Heuristic Generation'. See: Derix, Christian, ed. Empathic Space: The Computation of Human-centric Architecture. London: Wiley, 2014. p24-53.

intuition upon what is satisfactory and what is not.[188] For further discussion in this sense, it is essential to introduce, the example of the "Dada Engine":

> […] the Dada Engine is a computer algorithm that produces random text based on recursive rearrangement of elements in a grammar. The resulting text, while allegedly based on random processes, is readable, occasionally makes sense, and is sometimes surprisingly intelligent.[189]

The Dada Engine illustrates that a virtual entity that represents variations through rules and relations does not necessitate that every instance or permutation that are included in the possibility space are meaningful or satisfactory. In this sense softforms embody all of the logic of production of form whether satisfactory or not, as long as these instances adhere to the inherent formal rules and relations.

Softforms allow exploration of outcomes of decision-making through inherent indefiniteness and indetermination. This situation about softforms are based on undecided properties about form, or in different terms, a lack of intentionality about design. 'Softness' of the softform is related to this lack of intention on situations, where the 'softer' is the more undecided and explorative. Increased intentionality provides elimination of unsatisfactory instances and eventually resulting in specific, concretized and hardened final form.

4.2.1. Virtualization and Encapsulation

The dialogue between information and conceptualization through expressions provides for an important point of discussion in the definitions or descriptions of changeabilities for form. Expressions are understood as descriptions of a possible range of outcomes and agents in the virtualization of design intention themselves governed by inherent relations and topologies. For further discussion, it may be useful here to mention John Frazer's argument on virtualization and virtual worlds:

---

[188] Terzidis 2008, op. cit.

[189] Terzidis 2008, op. cit., p77.

> Virtual worlds should not be seen as an alternative to the real world or a substitute, but as an extra dimension which allows us a new freedom of movement in the natural world. In other words, the transcendence of physicality in the virtual world allows us to extend our mode of operation in the physical world. A new means of travel, a new form of communication, a new way of operating, a new medium for expression.[190]

According to Frazer, virtual worlds provide extra dimensions to understand, create and optimize relationships[191], that are capable in providing softforms, hence describing dynamic conditions of changeability through topologies and relations. Virtualization of design intention through expressions allows the exploration, creation, understanding, optimization and operation on such relations. Additionally, according to Manuel De Landa, approaches in conceiving relations through virtualization is based on problem-posing thinking.[192] De Landa provides his discussion through Gilles Deleuze's arguments on diagrams and diagrammatic reasoning:

> Deleuze proposes that thinking consists not in problem-solving (as most treatments on diagrams and diagrammatic reasoning suggest), but on the contrary, that given the real (through virtual) existence of problems in the world itself, true thinking consists is problem-posing that is, in framing the right problems rather than solving them. It is only through skillful problem-posing that we can begin to think diagrammatically.[193]

De Landa and Deleuze's arguments are especially relevant in terms of expressions that represent conceptual structures, and infer the necessity of elaborately defined relations.

---

[190] Frazer, John. "The Architectural Relevance of Cyberspace", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013. p49.

[191] Ibid. p49-52.

[192] De Landa, op. cit., p30-34.

[193] De Landa, op. cit., p34.

> On the one hand, the different topologies allowed different ways for analyzing and manipulating the elements, but on the other each topology also restricted the formal result; for example, rectangular grids allowed only orthogonal results. A huge part of the work as a designer of these programs was to find the best topologies to solve the problems and ways to overcome the restrictions.[194]

Similar to the previous discussion, Markus Braach emphasizes the need for “best possible relations for less constraining design models”.[195] As illustrated by a number of authors, defining boundaries of possibility spaces is based on a wide discussion in terms of both virtualization and complexity. The essential problematic in the case of softforms is about providing best topologies, provisioning about changeabilities or mutabilities of form, hence elaborately determining boundaries and constraints to this end, namely the encapsulation of the broadest possibility space. More specifically for expressions, the issue is about finding ways to capture mutabilities in either visual or mathematical/textual form to ensure flexibility in different conditions and variability to allow exploration. In this sense, encapsulations for softforms are described by defining changeabilities and flexibilities regarding generalizations and specificities. Such definitions differ in descriptions by mathematical/textual or visual expressions, but both types of descriptions require an approach that takes into account form changeability, thus of conceiving mutabilities in the possibility space.

4.2.2. Conceiving Changeability and Flexibilities as Encapsulation of Possibilities

Following the discussion on virtualization and encapsulation, conceiving mutabilities for form depends on definitions or extractions of formal patterns which are efficient in the instantiation or the internal organization of form: A formal logic which consists of either or both aspects of visual specificity, that is, the

[194] Braach, Markus. “Solutions You Cannot Draw." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014. p48.

[195] Ibid. p46-53.

implementation of form and the conceptual or relational aspects that constitute general and abstract structures which produce form.

Regarding extraction of formal patterns, a duality of interpretation and implementation are mentioned by Derix and Izaki in terms of both experiencing spaces and designing spaces with computational design methodologies, where experiencing space is an event which depends on both physical and mental responsiveness to the environment.[196]

> Representing body-knowledge requires us to 'read' people and spaces. Representing their behaviour as patterns involves the 'writing' of abstractions. To analyze and generate within a design system based on non-standard experiential patterns, reading as well as writing computer algorithms is necessary.[197]

According to Derix and Izaki, computational design requires interpretations as well as implementations.[198] In terms of expressions the two positions of interpretation/observation and implementation/making are taken together in reading/encoding and writing/decoding behaviors for the production of form, or in broader terms, encapsulating mutabilities for form. Hence, such encapsulation is a matter of generalization and specification, or interpretation and implementation respectively. In this sense, there are two positions regarding the encapsulation of mutabilities where one is that of reading/observation/abstraction/ interpretation/encoding and the other, that of writing/making/specifying/ implementing/decoding.

The first position, that of interpretation/observation, is based on distinguishing recurring generalizable patterns of actions and relations between entities that induce

---

[196] Derix and Izaki highlight the importance of these two positions in generating spaces that are responsive to their users. See: Derix, Christian and Izaki, Asmund. "New Curricula: Syntonic Systems." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014. p122-129.

[197] Ibid. p122.

[198] Ibid. p122-129.

or produce form in order to provide for formalized or generalized descriptions encapsulating permutations. In the frame of computational design, computation has the power to implement complex structures with high specification while ensuring permutation and changeability. In contrast, computers lack human abilities such as causality and intuition to create interpretations about situations.[199] Therefore, expressions are bound to human interpretation in conceiving conceptual structures which are represented with the more abstract forms of expressions such as Leon Van Schaik's Ideograms [Figure 4.8] or conceptual diagrams [Figure 3.3]. Through such conceptual representations about either design problems or solutions, interpretations about complex situations[200] are formalized and re-used in design conception.[201]

The second position, that of implementation/making is based on creating physical solutions to abstract and conceptual structures of design. Form is diversified, specified and instantiated by this position within an encapsulated, thus relatively limited boundary of mutability. Juhani Pallasmaa expresses the prominence of "empathic imagination" in producing material/physical answers to immaterial/conceptual structures.[202] By the notion of "empathic imagination" Pallasmaa describes a correspondence between immaterial and material, or

---

[199] Kostas Terzidis mentions the lack of causality and abilities in decision making in computational systems. See: Terzidis 2008, op. cit., p75-86.

[200] Additional theoretical discussion on observations or interpretations of complex phenomena is presented in Georg Vrachliotis' text on operationalization of complexity. Vrachliotis provides a discussion that is based on Karl Popper's situated logic and Gestalt psychology, thus illustrates part and whole relationship in complex structures both in terms of local and global complexity. See: Vrachliotis, op. cit., p59-72.

[201] Izaki and Helme illustrate usage of computational systems in calculating, visualizing and stimulating human-centric architectural conditions through a series of works. Authors provide works which are induced by formal systems that 'encode' user experiences and behaviors into design intention through computational methodologies. See: Izaki, Asmund and Helme, Lucy. "Encoding User Experiences." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014. P114-121.

[202] The notion of "empathic imagination" which is presented by Juhani Pallasmaa may be further extended to the notion of "Einfühlung" -a feeling into form- in aesthetics, as referred and illustrated by Robert Vischer. See: Pallasmaa, op. cit., p80-85; Vischer, Robert, Harry Francis Mallgrave, and Eleftherios Ikonomou. Empathy, Form, and Space: Problems in German Aesthetics, 1873-1893. Santa Monica, CA: Getty Center for the History of Art and the Humanities, 1994.

conceptual and physical, hence the designer's ability to project solutions to conceptual problems.[203]

Definitions for encapsulations are provided mainly by the elimination of specificity that refers to distinct instances of the possibility space, and consequently focusing on generic characteristics about these instances. In this sense, generalization that ensures changeability, specificity and flexibility are essential for encapsulations in which permutations can occur; and therefore has the power to provide specific and concretized instantiations of generic intentions. In this context, encapsulations are bound to generic characteristics and constraints of form, and describe possible specific solutions through generic characteristics. Moreover, the higher is the degree of generalization, the higher is the degree of multiplicity.

Softforms represent all possible allowable designs that adhere to encapsulations or to formal definitions. This situation about softforms is exemplified in what is called as *regular expressions* [Figure 4.14] which provide the basis for formal languages and an important preliminary for computation; Regular expressions define patterns which match or generate a certain set of strings or sentences from a predefined alphabet.[204] These expressions have the ability to provide sets of strings with finite or infinite variations, where finiteness of the result set is dependent on the definition in the expression, without the need to represent each of the strings, but only by providing a generalized description: A formal logic where variations of predefined generalized descriptions are conceived.

[203] Pallasmaa, op. cit., p80-85.

[204] Sudkamp, op. cit., p37-54.

Expression: a*ba*ba*

Description:

All of the strings in the alphabet of {a, b} that contains exactly 2 b's.

Result:

{ bb, baab, ababa, abbaaaa, aabbaa, aaabb, aaaabaaaab, ... }

Expression: a*(ba*ba*)*

Description:

All of the strings in the alphaber of {a,b} that contains even number of b's including empty string.

Result:

{ ε, bb, baab, ababa, abbaaaa, aabbaa, aaabb, aaaabaaaab, abbbb, bbaabb, abbbba, abababa, bababababa, bbaababaaa, ... }

Figure 4.14 Sample Regular Sets and Expressions

Produced by the author according to Thomas Sudkamp's regular expression illustrations. Sudkamp, Thomas A. "Languages" Languages and Machines: An Introduction to the Theory of Computer Science. Reading, Massachusetts: Addison-Wesley, 1988. P37-54.

Another important topic for discussion in the context of mathematical/textual expressions is provided by *lambda calculus*; a formal system in mathematical logic for expression of computations.[205] The importance of lambda calculus is that it provides the basis for sequential procedural programming languages, or in simpler terms, functional programming languages[206] which compute in iterative steps of reduction from procedural abstractions [Figure 4.15]; therefore are deeply related with

[205] Lambda calculus is developed by mathematician and logician Alonzo Church as a formal system based on mathematical logic for expressing computations based on function abstractions and applications. See: Church, Alonzo. "The Calculi of Lambda-Conversion." Annals of Mathematics Studies. No.6. Princeton University Press, Princeton. 1941.

[206] Peter Landin illustrates a correspondence between Alonzo Church's lambda calculus and expressions of the ALGOL 60 programming language. ALGOL 60 is an old programming language but, lambda calculus is commonly accepted basis to modern functional programming languages. See: Landin, Peter J. "A Correspondence Between ALGOL 60 and Church's Lambda-Notation" Communications on the ACM, Vol.8, No.2. 1965. p89-101.

mathematical/textual expressions such as the code. Basic mechanisms for lambda calculus are involved with abstraction by binding variabilities to function, and reduction by substitution of variables with specific values, thus applying functions. For lambda expressions, each time a variable is bound with a specific value, the solution becomes more reduced, concrete and determinate as abstraction dissolves into specificity, thus each iterative reduction narrows down the result set.

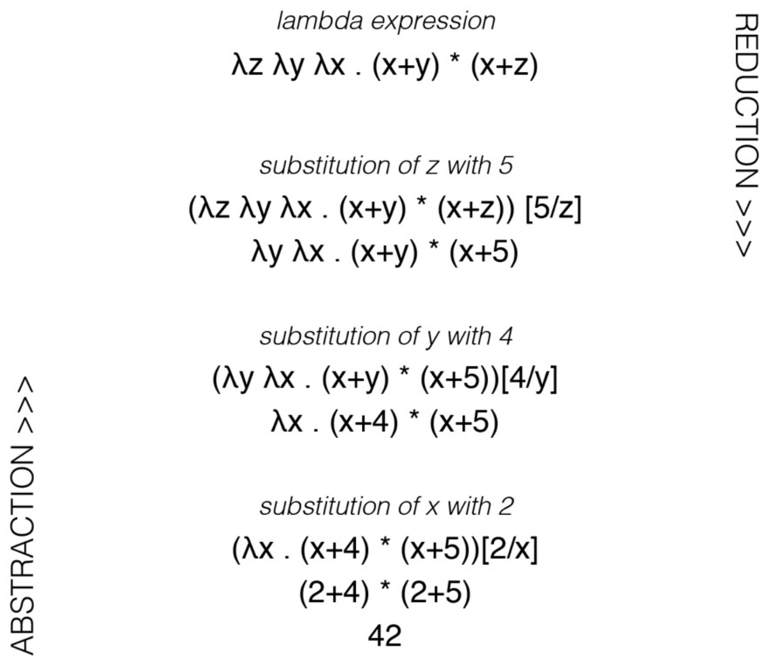


Figure 4.15 Abstraction and Reduction in Lambda Calculus

Produced by the author.

In contrast to characteristics of changeability and generalization in the form of abstraction and the elimination of numerical specificity exhibited in mathematical/textual expressions, changeability in the case of visual expressions is characterized by transformations and transformability in form.[207] Visual descriptions include transformations from basic formal definitions which have the

[207] Goethe, Johann Wolfgang. “Formation and Transformation”, Goethe’s Botanical Writings. Trans. Bertha Mueller. Woodbridge, CT: Ox Bow Press, 1989, p21-24.

capability to provide multiple results, similar to the case of regular expressions and lambda calculus [Figure 4.16].

Greg Lynn mentions Rene Thom's morphogenetic diagrams to discuss "forms of dynamic stability" that are found in these diagrams: "geometric modelling of a multiplicity of possible co-present events at any moment".[208] This co-presence of a multiplicity of conditions introduces indeterminacy and dynamism to the state of geometry. Hence, encapsulations in visual form are defined by abstractions in the form of multiple allowable states of geometries and form. Furthermore, the increased amount of inherent relations in form exhibits complexity and unpredictability in the state of form.

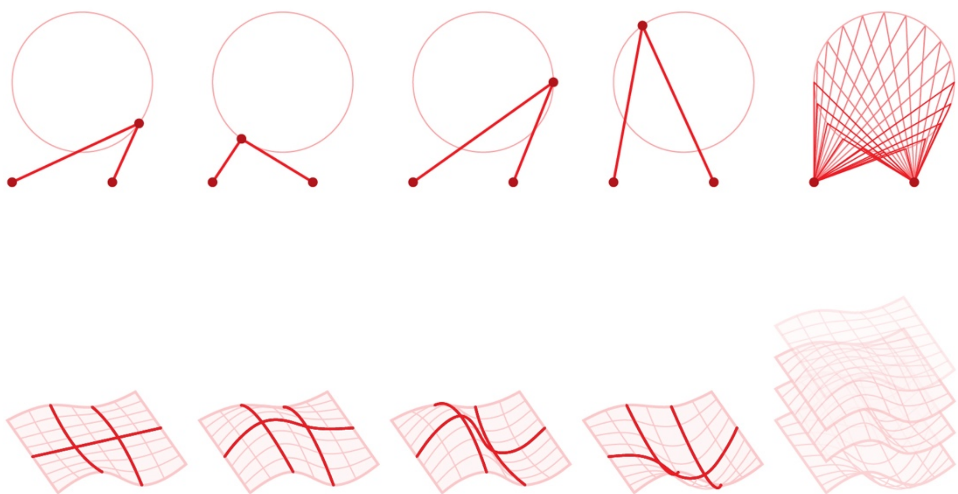


Figure 4.16 Transformations and Transformability in Form in Different Dimensions

Produced by the author.

Both visual and mathematical/textual modes of expressions, in terms of providing descriptions of softforms, are related to form-blindness through abstraction,

[208] Greg Lynn provides a discussion of transformations based on topology and transformability while mentioning Gilles Deleuze's Fold and Rene Thom's Catastrophes in terms of continuous transformations and mathematical probabilities for unpredictable events: A description of transformability for form and spatial conditions that adheres to multiple co-present differentiated events and elements. See: Lynn, op. cit.

generalization and elimination of specificity. Previous examples exhibit methodological and technical implementations of description of possibility spaces. In addition to methodological aspects in terms of descriptions, encapsulations of possibility also refer to conceptual structures of design which regulate phenomenological qualities such as aesthetics, multiplicity or meaning in space. These qualities are based on the relations which define encapsulations. In this sense, encapsulating possibility spaces refers to expressions being observational and descriptive as well as operational; qualities which are exhibited simultaneously by the notions of form-blindness and softform.

This chapter presented preliminaries and essentials in expressions for defining changeabilities in form and implementing complex orders to the design in order to achieve better integrated, flexible results which are required by contemporary design problems. The discussion in this chapter, is involved with the development of two notions, form-blindness and softform, which ensure complexity and changeability in design conceptualization. Both of these notions are mutual and related to each other in the sense that as form-blindness increases, softness in form also increases. Both notions, as approaches taken towards design, imply a lack of intention and specificity in the immaterial aspects which endue form, such as information. Finally, both of these qualities aid design processes together in achieving complexity management, multiplicity and mutability in architectural form conception.

CHAPTER 5

CONCLUSION

> Language [···] is more than just a medium for expressing thought. It is, in fact, a major element in the formation of thought. Furthermore, to use a figure from our own day, man's very perception of the world about him is programmed by the language he speaks, just as a computer is programmed. Like the computer, man's mind will register and structure external reality only in accordance with the program. Since two languages often program the same class of events quite differently, no belief or philosophical system should be considered apart from language.[209]

Expressions have a prominent location in establishing the dialogue between information and conceptualization of design during the process from idea to form. Similar to what is argued by Edward Hall in the context of languages, expressions of design intentions have the power to shape perception and thinking in design.[210] Hence whether in visual or in mathematical/textual form, either referring to perceptual or conceptual structures in design, expressions provide for different definitions, meanings and interpretations regarding their own characteristics and expressiveness. Therefore, different expressions constitute different thinking, different interpretations and readings about situations, as well as producing different outcomes in design. They are results of design languages that display qualities of flexibility, adaptability, variability, complexity, expressiveness and are

---

[209] Edward Hall is an anthropologist whom is a close acquaintance of Buckminster Fuller. He argues that language, as a communication tool shapes thought and investigates this power of language in scrutiny of different cultures and languages. Hall, Edward T. "Culture as Communication." The Hidden Dimension. Garden City, N.Y.: Doubleday, 1966. P1-2.

[210] Ibid.

efficient in providing such qualities to design thinking. In the more general sense, expressions enable the conveying and conception of custom information models that provide for multiplicities through complex orders necessitated by contemporary architectural design.

This thesis presented a discussion based on expression and thinking in the context of the computational design paradigm by comparing different modes of expressions used in architecture during the design process. The study discussed issues that are based on topics such as the relation of information to its representation and the expression of design intention. These topics were illustrated through issues such as; encapsulating and broadening possibility spaces of form; inhabiting multiplicity of approaches; complex orders and multi-layered concerns that provide for better integrated designs; relationships and correspondences between the visual and the relational or the conceptual and the perceptual; contrasts, relations and similarities between visual and relational reasoning; models and expressions; the importance of abstraction and specificity; expressiveness and structure in expression to cite the most prominent sub-topics of the thesis discussion.

These discussions have led to the development of two notions or formal qualities, that of form-blindness and softform, which taken together as design approaches, foster multiplicities and the management of complexity in a new paradigm of architecture in the information age which is defined by custom information models, possibility spaces, and complex orders. Expressions which possess these qualities are necessary for defining the boundaries of solutions and the mechanics necessary for the production of solutions.

Form-blindness is presented as an approach which liberates perceptual structures and architectural form from the conceptual structures during the design process. Specificity impairs complexity management and workability on models.[211]

[211] Dogan and Nersessian mention “generic abstraction” as “a form of reasoning that selectively surpasses information present in a specific representation”. See: Dogan and Nersessian 2010, op. cit., p207-236.

Abstraction enables and encourages complex solutions where the essential prerequisite of abstraction, in the context of complexity management, is the elimination of inessential specificity. In this case, form-blindness is an approach which is involved with maintaining the level of specificity about, or namely an abstraction over form, in order to be able to manage the complexity of the increased amount of information that imbues design.

A possibility space for form has been seen to be wider when design intention is more abstract, formless and less specific. As blindness over form decreases, the expression that is used to describe form becomes less abstract and the image of the final form becomes more specified and rigorous. The importance of maintaining the level of specificity about form is that, the early fixation on the solution narrows down the possibility space, or impairs changeability for alternate designs in providing concretized forms, consequently removing exploration, diversification and multiplicity from the design process. Hence, for wider potential spaces for form, the least possible should be described of form in expressions, as further interest in realization details provides a problematic in terms of managing complexity and providing changeability to form. For manageable complexity and mutability in form, form-blindness needs to be preserved during the design process and reliance on perceptual structures needs to be postponed.

Approaches for broadening possibility spaces for design that are defined by the notion of form-blindness are mutual with approaches defining the boundaries of, or in other words, encapsulating possibility spaces. Both of these factors which necessitate an abstraction over form, highlight a correspondence of perceptual and conceptual structures through controlled means. Through abstraction and form-blindness, it is possible to respond to problems with solutions which arise from complex orders that are constituted by multi-layered concerns, rather than linear determinations; thus it is possible to both increase the correspondence of architectural form with conceptual structures and liberate form from a constraining fixation on a single instance. In this sense, the notion of softform and encapsulations of mutabilities for form, is associated with the notion of softness. Softness in form refers to the degree of changeability by either methodological or conceptual aspects of design intention. In terms of softness, increased specificity in the definition of form reduces this capability of form to change. Therefore, the notion of softness is continous with abstraction and form-blindness in expressions. Together with form-blindness, softness defines how much form can change in terms of its geometry and inherent relations, while hardness would denote its resistance to change.

Softforms are entities which are capable of representing and actualizing possibility spaces, hence entities which are manifestations of possibility spaces. Manuel De Landa provides a discussion of Deleuze and Guattari's notion of topological forms and abstract machines which are "pregnant not only with possibilities which become real, but with virtualities which become actual".[212] A softform is not a real form and it does not define a design solution. Softforms, instead, present variability and flexibility through customization and exploration in a singular mutable instance. Moreover, softforms internalize dynamic responses to complex conditions. They are results of open-ended, probabilistic and indeterminate approaches to form that arise from descriptions of design intentions.[213] Both notions of form-blindness and softform highlight the correlation between possibility spaces and the inherent relations that produce form. Conceiving these relations is a matter regarding both the complexity management of the conditions which make form and the provision of its changeability.

Therefore, as approaches, both form-blindness and softform are presented as frameworks which incarnate broader possibility spaces for form and enable the incorporation of multi-layered concerns. As qualities in expressions, both imply the necessity for a more abstract and generic, therefore less specific descriptions regarding design intention, instead of a definition about a final product, similar to what is exhibited in conceptual diagrams[214]. Namely, whether as representations of perceptual or conceptual structures in design, expressions which obtain a certain

---

[212] De Landa, op. cit.

[213] De Landa, op. cit.

[214] Dogan and Nersessian 2010, op. cit.

degree of form-blindness besides describing a softform entity are necessary to be able to work with the requirements of contemporary design problems. Such expressions enable the management of complexity through form-blindness and introduce multiplicity to form through their softness.

Evidently, both softform and form-blindness as concepts/approaches/qualities are inclusive and not quite distinct since both of these notions are deeply related with providing for possibility spaces and require a conditional detachment from specificity, or in different terms, call for dematerialization, generalization and abstraction. Furthermore, the degrees of softness in form and form-blindness in expressions are directly related where both are based on controlled specificity and the lack of intention. These qualities enable exploration on architectural form through liberation of and redefined correspondences between conceptual structures and perceptual structures. A freedom in architectural form is introduced via elimination of inessential specificity, thus allowing for multiplicity and extension in correspondences between conceptual and perceptual structures of design.

In this context of expressions, the duality between the visual and the mathematical/textual modes of expressions presented a pivotal issue in the thesis discussion. In contrast to the visual modes of expression, by virtue of computational techniques, the code makes it possible to maintain higher degrees of form-blindness and softness, but at the same time, to produce an implemented, hardened or realized instance of form. Computation allows form to change however and whenever in the defined boundaries. The code enables realizations of actual forms through softforms in the virtual space, an ability which visual expressions do not possess; it is capable of implementing conceptual and perceptual structures through software, on a digital substrate the results of which are instantiated directly on execution. Consequently, mathematical/textual modes of expressions are more relevant today since they have the power to provide for descriptions of form that are possible to be executed directly.

Nevertheless, this is not to say that the code should be the only mode of expression to be used, as each mode of expression have different qualities in terms of expressiveness hence has the expressive capability to provide descriptions of different objects or concepts that are suitable to their own nature[215]; which is especially the case when concerned with visual and relational reasoning together with conceptual and perceptual structures of design. Either conveyed in contrasting visual or mathematical/textual modes, expressions provide for different descriptions of softform entities, with a different degree of form-blindness and therefore a different purpose regarding the stage of the design process. The problem of contemporary architectural design is, therefore, to create design languages and descriptions of form, such as the ones that are expressed by the code, conceptual diagrams or ideograms, which do not depict and solidify progression or resolution of form, but instead, are descriptions of form and design intentions which are possible to be initiated in multiple ways through the production of form. Both concepts/approaches/qualities of form-blindness and softform, that are developed throughout this thesis are instigated for creating design languages that are efficient in yielding multiplicities that are required by the complexity of contemporary architectural problems.

Further study on this topic would involve a praxis-based implementation of form-blindness in different expressional media. Throughout the thesis, form-blindness has been presented as an inherent and intuitive property of expression and thinking, but not as a deliberate approach taken towards design. Here, especially the code presents a wide discussion on testing the correlations between conceptual and perceptual structures of design in the context of form-blindness, hence it seems more relevant to implement a form-blind architectural form conception process in the code. Namely, as form-blindness offers independent development of conceptual and perceptual structures, it should be researched to what extent form is diversified in phenomenological or perceptual terms according to the same design intention or conceptual structure. Another topic of research would be a further analysis on

[215] Mennan 2014, op. cit.

abstraction in formal systems as this provides for an important point of departure for the discourse of contemporary architectural praxis especially in terms of complexity management. Although this thesis presented a generic discussion in terms of abstraction in both visual and mathematical/textual media, a relevant question would be to inquire into the kind of abstraction that provides the best relational models for multiplicitous designs with manageable complexity. For mutability in form, more discussion on providing, not just multiplicity, but meaningful multiplicity in form seems relevant in implementing meaning and interpretation into architectural form and overcoming the deficiencies exhibited by the formal methodologies of the computational medium. Finally, and perhaps most importantly, a research on building custom design languages which inherit these two notions as basis seems to hold most prominent in this context.

BIBLIOGRAPHY

Any Magazine Issue 23, *Diagram Work: Data Mechanics for a Topological Age* 1998.

Allen, Stan. "From Object to Field", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Braach, Markus. "Solutions You Cannot Draw." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014.

Cache, Bernard. "Philibert De L'Orme Pavilion: Towards an Associative Architecture.", Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Church, Alonzo. "The Calculi of Lambda-Conversion." Annals of Mathematics Studies. No.6. Princeton University Press, Princeton. 1941.

Cook, Peter. Drawing, the Motive Force of Architecture. Hoboken: Wiley, 2008.

De Landa, Manuel. "Deleuze, Diagrams and the Genesis of Form". Any Magazine Issue 23, 1998.

Deleuze, Gilles and Guattari, Félix. "587B.C. - A.D.70: On Several Regimes of Signs." A Thousand Plateaus: Capitalism and Schizophrenia. Trans. Brian Massumi. Minneapolis: University of Minnesota Press, 1987.

Deleuze, Gilles and Guattari, Félix. “1440: The Smooth and the Straited." A Thousand Plateaus: Capitalism and Schizophrenia. Trans. Brian Massumi. Minneapolis: University of Minnesota Press, 1987.

Derix, Christian, ed. Empathic Space: The Computation of Human-centric Architecture. London: Wiley, 2014.

Derix, Christian. "Introduction, The Space of People in Computation." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014.

Derix, Christian and Izaki, Asmund. “New Curricula: Syntonic Systems.” Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014.

Dogan, Fehmi and Nersessian, Nancy. “Conceptual diagrams: representing ideas in design”, Diagrams 2002. Eds. M. Hegarty, B. Meyer & N. H. Narayanan. Berlin: Springer, 2002.

Dogan, Fehmi and Nersessian, Nancy. “Generic Abstraction in Design Creativity: the Case of Staatsgalerie by James Sterling”, Design Studies No.31(3), 2010.

Dogan, Fehmi and Nersessian, Nancy. “Conceptual Diagrams in Creative Architectural Practice: The case of Daniel Libeskind’s Jewish Museum”, arq: architecture research quarterly, 2013.

Eisenman, Peter. *Diagram Diaries*. New York, NY: Universe Pub., 1999.

Evans, Robin. "Architectural Projection." Architecture and Its Image: Four Centuries of Architectural Representation: Works from the Collection of the Canadian Centre for Architecture. Ed. Eve Blau and Edward Kaufman. Montreal:

Centre Canadien D'Architecture/Canadian Centre for Architecture, Cambridge, Mass; Distributed by,the MIT Press. 1989.

Foreign Office Architects. “Yokohama International Port Terminal”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Frazer, John. “The Architectural Relevance of Cyberspace”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Garcia, Mark, ed. The Diagrams of Architecture. Chichester: Wiley, 2010.

Gleiniger, Andrea, and Georg Vrachliotis, eds. Complexity Design Strategy and World View. Basel: Birkhäuser, 2008.

Gleiniger, Andrea. ""The Difficult Whole," or the (Re)discovery of Complexity in Architecture." Complexity, Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008.

Goethe, Johann Wolfgang. “Formation and Transformation”, Goethe’s Botanical Writings. Trans. Bertha Mueller. Woodbridge, CT: Ox Bow Press, 1989.

Hall, Edward T. "Culture as Communication." The Hidden Dimension. Garden City, N.Y.: Doubleday, 1966.

Hensel, Michael, Achim Menges, and Micheal Weinstock, eds. Techniques and Technologies in Morphogenetic Design. London: Wiley-Academy, 2006.

Hight, Christopher, Micheal Hensel, and Achim Menges. "En Route: Towards a Discourse on Heterogeneous Space beyond Modernist Space-Time and Post-Modernist Social Geography." Space Reader: Heterogeneous Space in Architecture. Eds. Michael Hensel, Christopher Hight, and Achim Menges. Chichester, U.K.: Wiley, 2009.

Hight, Christopher and Perry, Chris “Introduction to Collective Intelligence in Design”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Huth, Michael, and Mark Ryan. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge: Cambridge UP, 2000.

Izaki, Asmund and Helme, Lucy. “Encoding User Experiences.” Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014.

Kızılcan, Egemen Berker. "Non-standard Materialities Tailored for Soft and Hard Construction." *What's the Matter? Materiality and Materialism at the Age of Computation*. Ed. Maria Voyatzaki. European Network of Heads of Schools of Architecture (ENHSA), 2014.

Landin, Peter J. “A Correspondence Between ALGOL 60 and Church’s Lambda-Notation” Communications on the ACM, Vol.8, No.2. 1965.

Legendre, George, ed. Mathematics of Space. London: Wiley, 2011.

Lorenzo-Eiroa, Pablo. "Form:in:form, On the Relationship Between Digital Signifiers and Formal Autonomy." Architecture in Formation: On the Nature of Information in Digital Architecture. Ed. Pablo Lorenzo-Eiroa and Aaron Sprecher. London: Taylor & Francis, 2013.

Lynn, Greg. “Architectural Curvilinearity: The Folded, the Pliant and the Supple”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Mainzer, Klaus. “Strategies For Shaping Complexity in Nature, Society, and Architecture”, Complexity Design Strategy and World View. Gleiniger, Andrea, and Georg Vrachliotis, eds. Basel: Birkhäuser, 2008.

McCullough, Malcolm “20 Years of Scripted Space”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Mennan, Zeynep. “Des Formes Non Standard: Un ‘Gestalt Switch’.” (“Of non standard forms: A ‘Gestalt Switch’”) Architectures Non Standard. Eds. Migayrou, Fréderic and Zeynep Mennan. Paris: Editions du Centre Pompidou. 2003.

Mennan, Zeynep. “Non-standardization Through Non-visualization: Scripting the Dom-ino House” Game Set and Match II; On Computer Games, Advanced Geometries, and Digital Technologies. Eds. Kas Oosterhuis and Lukas Feireiss. Episode Publishers. 2006.

Mennan, Zeynep. “Mind the Gap: Reconciling Formalism and Intuitionism in Computational Design Research." Eds. Henriette Bier, and Terry Knight. Footprint, Delft Architecture Theory Journal, Vol.15. 2014.

Novak, Marcos J. "Computational Composition in Architecture." Computing in Design Education, ACADIA '88 Workshop Proceedings. Ed. P. J. Bancroft. Ann Arbor: University of Michigan, 1988.

Pallasmaa, Juhani. “Empathic Imagination, Formal and Experiential Projection." Empathic Space: The Computation of Human-centric Architecture. Ed. Christian Derix. London: Wiley, 2014.

Perella, Stephen. “Bernard Cache/Objectile: Topological Architecture and the Ambiguous Sign.”, Ed. Carpo, Mario. The Digital Turn in Architecture 1992-2012. Chichester: Wiley, 2013.

Peters, Brady, and Xavier De Kestelier, eds. Computation Works: The Building of Algorithmic Thought. London: Wiley, 2013.

Scheurer, Fabian, and Hanno Stehling. "Lost in Parameter Space?" Mathematics of Space. Ed. George Legendre. London: Wiley, 2011.

Sprecher, Aaron. “Architecture in Formation: On the Affluence, Influence, and Confluence of Information.” Architecture in Formation: On the Nature of Information in Digital Architecture. Ed. Pablo Lorenzo-Eiroa and Aaron Sprecher. London: Taylor & Francis, 2013.

Sprecher, Aaron and Lorenzo-Eiroa, Pablo. eds. Architecture in Formation: On the Nature of Information in Digital Architecture. London: Taylor & Francis, 2013.

Sudkamp, Thomas A. An Introduction to the Theory of Computer Science. Reading, Massachusetts: Addison-Wesley, 1988.

Terzidis, Kostas. Expressive Form: A Conceptual Approach to Computational Design. London: Spon, 2003.

Terzidis, Kostas. “Algorithmic Complexity: Out of Nowhere.” Complexity, Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008.

Tschumi, Bernard. "Responding to the Question of Complexity." Complexity: Architecture, Art, Philosophy. Journal of Philosophy and the Visual Arts. Ed. Andrew E. Benjamin. No. 6. London: Academy Group, 1995.

Türkay, Seray. The Orthographic Set: Making Architecture Visible. M.Arch. Thesis METU, Ankara, 2011. Retrieved from: http://etd.lib.metu.edu.tr/upload/12613652/index.pdf

Van Berkel, Ben and Bos, Caroline. “Diagrams - Interactive Instruments in Operation”. Any Magazine Issue 23, 1998.

Van Schaik, Leon. “*Spatial intelligence: new futures for architecture*”. Chichester, England: Wiley, 2008.

Van Schaik, Leon. Ideograms. Melbourne: Lyon Housemuseum, 2013.

Van Schaik, Leon. "How Can Code Be Used To Address Spatiality in Architecture." Ed. Christian Derix and Asmund Izaki. Empathic Space: Computation of Human-Centric Architecture. 2014.

Vrachliotis, Georg. "Popper's Mosquito Swarm: Architecture, Cybernetics, and the Operationalization of Complexity." Complexity Design Strategy and World View. Ed. Andrea Gleiniger and Georg Vrachliotis. Basel: Birkhäuser, 2008.