DISS. ETH NO. 21516

Architectural Design as Combined Modeling

A dissertation submitted to
ETH ZURICH

for the degree of

Doctor of Sciences

presented by

HAO HUA

M.Arch., Southeast University, China

07.02.1983

citizen of P. R. China

accepted on the recommendation of

Prof. Dr. Ludger Hovestadt
Chair of Computer-Aided Architectural Design, ETH Zurich

Prof. Dr. Li Biao
School of Architecture, Southeast University, China

Prof. Dr. Urs Leonhard Hirschberg
Institute of Architecture and Media, TU Graz

2013

Abstract

This work investigates the potentials and the limitations of models in Computer-Aided Architectural design (CAAD). The research focuses on the symbolic models instead of the analogue models that resemble their referents. The symbolic model can represent architecture with abstract signs and produce design solutions through formal operations. Over the past five decades, more and more models have been available for architecture and have been practical with the help of computers. We regard the activity of modeling essential in coupling symbolic models with architectural design. However, modeling architecture is still a challenge mainly because of two reasons. First, design problems are usually not well defined. For instance, Rittel (1973) concluded that design problems have no definite formulations, no stopping rules, and that they are inherently unique. Thus, there is no general model of architectural design that is widely accepted. Second, there are dilemmas in modeling. Previous studies have demonstrated that models are essentially wrong or partial. Besides, the nature of modeling can result in multiple inconsistent models of the same subject matter.

In order to solve the problem of multiple models, two common views are investigated. One holds that a single coherent view on the object is necessary in modeling. The relevant approach is metamodeling, i.e., making the model of models. The other view concedes that multiple inconsistent views are inevitable. Therefore, a single model is unnecessary and is not rewarding. This work finds out that the alternative solution, combining multiple models, could be better than developing a single model or metamodel. We consider combined modeling as a natural response to the dilemmas in modeling. In particular, the fictional combination of models could be fruitful in architectural design. Architects usually modify, add, and transform multiple issues in design. By combining multiple models, the designers can free themselves from the epistemic framework of each model and subsequently make their own position by organizing the interrelationships between the models.

This dissertation discusses on both modeling with computers and without computers, including a brief survey on the historical employment of mathematics and geometry in architecture. The proposed modeling methods are tested in a series of experiments with computers. They mainly focus on the topological and geometrical design of buildings. All experiments are implemented in the Java programming language. The results imply that the combination of multiple models could be productive in architectural design.

Zusammenfassung

Die Arbeit untersucht das Potential und die Grenzen von Modellen im Computerunterstützten Architektonischen Design. Die Forschung konzentriert sich auf symbolische Modelle anstatt auf analoge Modelle, die sich an ihren Referenzen orientieren. Das symbolische Modell kann Architektur mit abstrakten Zeichen repräsentieren, und Entwurfslösungen durch formale Operationen produzieren. Während der letzten 5 Jahrzehnte, sind immer mehr Modelle für Architektur verfügbar und durch die Hilfe des Computers anwendbar geworden. Wir betrachten den Vorgang des Modellierens als essentiell an bei der Kopplung von symbolischen Modellen und architektonischem Entwerfen. Aber hauptsächlich durch zwei Gründe bleibt es eine Herausforderung Architektur zu modellieren. Zum einen sind Entwurfsprobleme normalerweise nicht sauber definiert. Rittel (1973) erklärt, das Entwurfsprobleme sich nicht definitiv formulieren lassen, keine Endbedingung haben, und immer einzigartig sind. Dadurch gibt es kein generelles Model für Architektur, das eine breite Akzeptanz findet. Zum anderen gibt es ein Dilemma im Modellieren. Vorherige Studien haben gezeigt, dass Modelle essential falsch oder unvollständig sind. Des Weiteren kann die Natur des Modellierens zu multiplen, inkonsistenten Modellen zum gleichen Thema führen.

Um das Problem von multiplen Modellen zu lösen werden zwei herkömmliche Standpunkte betrachtet. Der eine setzt voraus, dass eine einzige, kohärente Betrachtung auf das Objekt nötig ist um es zu modellieren. Der relevante Ansatz dazu ist das sogenannte Meta-modelling, also das Entwickeln eines Modells von Modellen. Der andere Ansatz geht davon aus, dass unterschiedliche, inkonsistente Standpunkte unvermeidbar sind. Deswegen ist ein einfaches Modell unnötig und nicht zielführend. Diese Arbeit kommt zu dem Schluss, dass die alternative Lösung, mehrere Modelle miteinander zu kombinieren besser ist, als ein einziges Modell oder ein Metamodell zu entwickeln. Wir betrachten kombiniertes Modellieren als natürliche Antwort auf das Dilemma des Modellierens. Die synthetisierten Kombinationen von Modellen können fruchtbar im architektonischen Entwurf sein. Normalerweise modifizieren, addieren, und transformieren Architekten verschiedene Varianten im Entwurf. Beim kombinieren von unterschiedlichen Modellen können die Entwerfer sich von dem epistemischen Rahmen jedes Modells befreien, und ihre eigene Position beim in Beziehung setzen der Modellen bestimmen.

Die Dissertation betrachtet zugleich Modellieren am Computer und ohne Computer, und beinhaltet einen kurze geschichtliche Übersicht über Anwendung von Mathematik und Geometrie in Architektur. Die vorgeschlagenen Modelliermethoden sind in einer Serie von Experimenten mit dem Computer getestet worden. Diese fokussieren hauptsächlich auf dem toplogischen und geometrischen Entwurf von Gebäuden. Alle Experimente sind in der Programmiersprache Java programmiert. Die Ergebnisse implizieren, das die Kombination von verschiedenen Modellen produktiv im architektonischen Entwurf sein kann.

Contents

Abstract III
Zusammenfassung IV

1 Introduction 1

1.1 Modeling, Computing, and Architectural Design 1
  1.1.1 Modeling 1
  1.1.2 Computational models 3
  1.1.3 Modeling and computing in architecture 6
1.2 Dilemmas in Modeling 10
1.3 Objectives and Scope of the Work 12
1.4 Thesis Structure 14

2 Modeling without Computers 16

2.1 Pythagoras and Numbers 16
2.2 Proportion 18
2.3 Geometry 27
2.4 Durand's Method 32

3 Modeling with Computers 36

3.1 Modeling and Computing 37
  3.1.1 Turing machine 37
  3.1.2 Definition of modeling 38
  3.1.3 Views on computational models 40
  3.1.4 Recent views on computational models 42
  3.1.5 Reflections in science 43
3.2 Modeling in Architecture with Computers 46
  3.2.1 Modeling and computing for architecture 46
  3.2.2 Models of architecture 47
  3.2.3 Models of the design process 49
  3.2.4 Modeling, knowledge and expertise 56
  3.2.5 Parametric/generative models 62
  3.2.6 CAD systems and Building Information Modeling 66
  3.2.7 Discussion 69

4 Combined Modeling 71

4.1 Dilemmas of Modeling 71
  4.1.1 Correctness of models 72
  4.1.2 Effectiveness of models 76
4.2 Metamodeling and Combined Modeling 78
  4.2.1 Metamodeling 78
  4.2.2 Combined modeling 81
  4.2.3 Fictional combination of models and architectural design 83

5 Combining Models in Architecture: Experiments with Computers 87

5.1 Volume + Grid 87
  5.1.1 The two models: volume and grid 87
  5.1.2 A synthesis program 88
5.2 Topology and Form-finding 92
  5.2.1 Topology + Grammar 92
  5.2.2 Topology + Shapes 96
  5.2.3 Topology + Pixels 103
5.3 Floor Plans + 3D Forms 108
  5.3.1 Floor plans and 3D forms of architecture 108
  5.3.2 The models of 3D forms 109
  5.3.3 The models of floor plan layouts 110
  5.3.4 Results and discussion 112
5.4 Case-based Reasoning and Fictional Aggregation 115
  5.4.1 Case-based reasoning and domain knowledge 115
  5.4.2 Recognition model + aggregation model 116
  5.4.3 Results and discussion 120

6 Conclusions 123

6.1 Design with Models: A Point of View 123
6.2 Implications for Computational Design 125
6.3 Fictions Powered with Models and Computers 127

References 129
Figures 146
Acknowledgements 149
Curriculum vitae 150

Chapter 1

Introduction

> "A model is a work of fiction" (Cartwright 1983, 153)

Forty years ago, Horst Rittel (1973) suggested that design problems are "wicked problems." In the contemporary context of Computer-Aided Architectural Design, how can we approach the wicked problems with an increasing number of computational models? Over the past five decades, there have been a great number of works that applied modeling and computing to architectural design. We have observed that people do not only model architectural design, but that they also transform the design itself through modeling and computing. Although computational models have become more and more powerful, some researchers revealed that the models are essentially wrong (Box and Draper 1987) or falsifiable (Stevens 1990). In order to examine the problems of modeling, this work investigates various notions of modeling as well as their applications to architecture. In particular, we find out that the design activities of architects usually involve multiple models instead of one model. Thus managing multiple models is essential for the synergy of modeling, computing and architectural design.

1.1 Modeling, Computing, and Architectural Design

1.1.1 Modeling

Innumerable researches from different disciplines have discussed the notion of modeling. For instance, Rothenberg (1989) alleged that:

> "Modeling in its broadest sense is the cost-effective use of something
> in place of something else."

It suggests that the model always represents something with something else. There are different kinds of models according to their representational functions. Ackoff et al. (1962) made three categories: iconic models, analogue models, and symbolic models. Iconic models look like their subject matter, such as photos and scale models. Analogue models make certain abstraction of the subject matter. For example, maps and bubble diagrams are analogue models. Symbolic models are the most abstract. They are written in mathematics or in other formal languages. This thesis focuses on symbolic models. Nowadays, symbolic models are pervasive. If we shoot a scene with a cell phone, we get a photo in the form of pixels. Each pixel could be represented by a set of RGB color values (the three components are red, green and blue). For instance, (255, 0, 0) denotes red color, (0, 255, 0) green, and (0, 0, 255) blue. The RGB system is a symbolic model of color. The image that we can watch on the screen is an iconic model of the scene, while the digital photo (pixels) is a symbolic model of the scene. However, what is a 3-dimensional model in a CAD (Computer-aided Design) software? This question could be ambiguous, since the 3-d model that we can see on the screen, as a pictorial rendering of the underlying data, is an iconic model of the building[1]. Yet the data structure of the 3-d model pertains to a symbolic model.

Besides the classification of models, some researchers put emphasis on the language of the models. Kleppe et al. (2003, 16) stated that "a model is a description of (part of) a system written in a well-defined language." The well-defined language must have a clear syntax and semantics. Logics and mathematics are often employed as the language of modeling. In software engineering, the Unified Modeling Language (UML) is a common language for modeling (Bézivin 2005). Mitchell (1990) demonstrated the first order logic as a critical language for design. In this language, each sentence is either true or false. For instance, a two-place predicate *parallel (**Line**, **Line**)* examines whether two lines are parallel. The language can enumerate all possible states of the corresponding well-defined design world.

In Allgemeine Modelltheroie, Stachowiak (1973) defined the three features of models: 1) Abbildung: the model represents its original; 2) Verkürzung: the model only represents part of the aspects of the original; and 3) Pragmatismus: the model has its own purpose, relatively independent of the original. His formulation implies that the relationship between the original and the model is essential in modeling; however, this relationship has been controversial. One mindset believes that the model should make faithful representation of its subject matter. For instance, Ashby (1957) held that we should establish the isomorphism between the model and the referent. However, many researchers realized that the model must be a partial representation of the original. Besides, the similarity between the two is subject to a certain epistemic framework. As Broadbent (1973, 89) said "No model can ever be complete, correct and universal in its application. We build a model because we want to focus down on certain aspects of a problem." Moreover, a few studies addressed that models do not necessarily represent something real. For instance, Barberousse and Ludwig (2009) stated that: "models are fictions, that is, precisely,

---

[1] It is either a virtual building under planning or a real building.

representations of fictional situations." Frigg and Hartmann (2012) pointed out that some models do not perform representational functions at all.

It has been widely agreed that the model only catches a part of the aspects of the original. Bender (1978) regarded the model as an abstract, simplified construct of the target. According to Smith (1985), the model is essentially partial. Since the models are always incomplete or partial, it is reasonable that all models are falsifiable (Stevens 1995). However, we sometimes verify a model through trying hard to falsify it. Carson (2004) explained the procedure of model verification: 1) suppose the model is correct; 2) try hard to prove the model is wrong or bad in some situation; and 3) if the model is still valid under these tests, it is verified. Of course, we are not able to test all possible situations; we are only interested in those relevant and important situations instead. Usually, people have little idea about the exceptional situation in which the model fails. For example, the computer program of an American missile warning system mistook the lunar reflection for a Soviet attack on October 5, 1960 (Smith 1985). In this case, the model was not adequate in the real environment, though many experts had carefully verified it.

Yet a number of studies put emphasis on the usefulness of models instead of their representational roles. Hence, the problem switches from "what the models are" to "what are the models good for" Kurpjuweit and Winter (2007) articulated: "A model is created by a modeler and interpreted by one or more users with respect to a certain purpose". Rothenberg (1989) considered cost-effectiveness as the essential attribute of modeling. In accord with them, Pinsky and Karlin (2011, 1) wrote: "In the final analysis, a model is judged using a single, quite pragmatic, factor, the model's usefulness." Besides the usefulness, people care about the costs. For example, if a model is too complex to manipulate or it is too expensive to collect the required input data, then the model is not a wise choice. In practice, people have to negotiate between the cost and the actual effectiveness of the model.

1.1.2 Computational models

Since commercial computers came into use in various fields, the number of computational models has been rapidly increasing. Theoretically, models can be made without computers, however, the burst of "computational" models is obviously due to the synergy between modeling and computing. Many of these models make no sense without computing. The studies of modeling and computing have been highly correlated. The theory of computing has been heavily attributed to Alan Turing. Turing (1950) defined the computer as a "discrete state machine." The machine takes certain signals as inputs, alters its internal states according to the predefined rules and the current inputs, and then maps the internal states to outputs. The good thing about digital computers is that they can implement any specification on inputs, outputs, internal states, and transition rules. As a result, digital computers are termed by Turing with "universal machines." In accord with Turing's formulation, Ashby (1962) defined the computer as follows: a set of internal states (**S**), a set of inputs or surrounding states (**I**), and a mapping $f(I \times S) = S$ that means "its internal state, and the state of its surroundings, defines uniquely the next state it will go to." From a more practical point of view, Newell and Simon (1956) developed the concept of the Information Processing System (IPS). The system consists of memories "that holds information overtime in form of symbols", and information processes "functions from the input memories and their contents to the symbols in the output memories."

One of the most important paradigms in computer science before the 1980s was problem solving. In "Human Problem Solving", Newell and Simon (1972, 809) claimed that:

> "We postulate that problem solving takes place by search in a problem space."

The problem space refers to the enumeration based on the representation of the problem. Thus a model adequately describing the problem is essential for problem solving. Newell and Simon (1972) pointed out that: "the whole difficult of solution resides in finding the right representation. Once that representation has been discovered, solving the problem becomes a trivial matter." Problem-solving consists of two elements: 1) representation of the problem (constructing problem space); and 2) searching for the best solution(s) in the problem space. In this image, the intelligence of computer systems is attributed to the problem space and the search method. As McCarthy and Hayes (1969) stated:

> "Intelligence has two parts, which we shall call the epistemological and the heuristic. The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do. "

Immense efforts have been made to develop smart search methods. The simplest method is exhaustive search (or brute-force search) that searches through all states of the problem space. Another simple solution is the "trial and error" method that repeats the two steps: 1) generate a new state; and 2) adopt the new state if it is better, otherwise restore the old one. However, the elementary search methods are not good enough for most non-trivial problems. Since the 1960s, people have developed a vast number of “clever” search methods under the umbrella of optimization. Various kinds of heuristics are employed to accelerate the search procedures, e.g., Simulated Annealing (Kirkpatrick et al. 1983) and the Particle Swarm Optimization (Kennedy and Eberhart 1995). Holland (1975) described a more sophisticated search algorithm, the Genetic Algorithm (GA), in his “Adaptation in Natural and Artificial System.” Bentley and Corne (2002) made a review on “Evolutionary Computation.”

Besides problem solving, a series of new computational models have emerged, for instance, Cellular Automata, Artificial Neural Networks, the Multi-Agent System, and so forth. Conway’s “Game of Life” (Gardner 1970) made Cellular Automata popular since the 1970s. The “game” is based on a grid in that each cell has two possible states: alive and dead. During each iteration, each cell changes its state according to the states of its eight neighbors. Unexpectedly, the simple discrete system exhibits complex patterns, which had amazed many people. Based on Cellular Automata, Wolfram proposed “a new kind of science” (2002) that studies the equivalent principles underlying computational models, natural phenomena, and our brain. It implies that the idea beneath traditional computational models (e.g. optimization) differs from that in dynamical models (e.g. Cellular Automata). The former plans a path toward the well-defined goal, while the latter defines the components and wonders about the behavior of the whole.

There have been increasing interests in the computational models of decentralized behaviors, such as Cellular Automata, the Self-Organizing System, and Swarm Intelligence. They are regarded not only as computational systems but also as the models of natural phenomena. As Camazine et al. (2001) explained the self-organization model: “Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system”, which is applicable for both computation and natural phenomena. The concept of “emergence” is closely associated with the decentralized behaviors in dynamical systems. De Wolf and Holvoet (2005) addressed that “a system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel w.r.t. the individual parts of the system.” Some researchers put these models under the umbrella of Complex Systems. For instance, Rocha (1999) pointed out that the nonlinear aggregation of the behaviors of the system components leads to the hierarchical self-organization of a complex system.

Recent developments in computer science have led to a renewed interest in data processing. Because of the exponential growth of computer hardware (computers, smart phones, the Internet, wireless networks, and so on) and the rapid expansion of communication platforms (Facebook, blogs, Wikipedia, online shopping platforms, and so forth), effective methods of processing enormous data are in urgent need. The field of Data Mining seeks computational models for discovering the patterns in large data sets. Data Mining is closely related with machine learning algorithms such as Regression, the Artificial Neural Network, and Self-Organizing Maps. A common feature of these approaches is that the parameters or the structures of the model are learned from the data rather than defined manually. As Hand et al. (2001) depicted the learning process as a way "to search over different model structures in a principal manner to find what appears to be the best model for a given task." Google researchers, Halevy et al. (2009), demonstrated how probabilistic models could be "unreasonably effective" in processing natural languages. In their model, the same procedure of learning and translation can be applied to different languages. For instance, translating between English and French is not substantially different from translating between Chinese and Italian (the difference is only the data set). The similar methods of learning can be found in others fields like image processing (Duygulu et al. 2002) and 3-d shape processing (Kalogerakis et al. 2012).

1.1.3 Modeling and computing in architecture

Pioneering works of Computer-Aided Architectural Design (CAAD) started in the 1960s. Following the problem solving paradigm[2] in computer science, the early works focused on the two issues: the building representation and the search process. Mitchell (1977) explicitly formulated architectural design as problem solving:

> "It assumes that we can construct some kind of a representation of the system that interests us, and that problem-solving can be characterized as a process of searching through alternative states of the representation in order to discover a state that meets certain specified criteria."

A few computer programs were developed to carry out design tasks according to predefined criteria, for example, Whitehead and Eldars's (1965) single-storey layouts planning program, and Seehof's (1966) automated facility layout program. Aguilar (1968) proposed a linear model of optimization in architecture. Yet, Brotchie and Linzey (1971) replaced the linear model with the quadratic model. The problem-solving paradigm reduced the architectural design into optimization. The optimization models usually consist of three elements: 1) building representation, 2) objective function or cost function, and 3) a search algorithm.

---

[2] Newell and Simon 1972, Human Problem Solving.

March and Steadman (1971) proposed various descriptions of buildings. They represented the buildings with geometrical objects or lattices that can be written in mathematics and coded by computers. Mitchell (1977) also discussed building descriptions in detail. Regular lattices were widely employed. Another common model was regular shapes, under most circumstances, rectangles. Krejcirik's (1969) Ru-Program and Weinzapfel's (1975) IMAGE system were two early experiments tat use rectangles. Splines, irregular polygons, surfaces and meshes have become popular in approaches that are more recent.

Besides various geometrical models, the graph is essential for representing the topology in architecture. A graph consists of a set of nodes and the links (or edges) between them. For instance, the nodes represent the rooms, and the links represent the connectivity between them. Hillier (1984) employed the graph to address the structure of interior spaces. He held that the interior structure is a function of the social environment. He first modeled the interior spaces of houses with graphs, then calculated several statistics over the graph structures, and finally coupled the statistics with the social concerns. Different from Hillier who used the graph for analysis, many researchers employed graphs as the topological specifications in spatial synthesis. For instance, Whitehead and Eldars (1965) represented the adjacency graph by a table of "association" values for every pair of rooms in the hospital program. In a contemporary computer program of residential building layouts (Merrell et al. 2010), the connectivity between the nodes (rooms) are automatically learned from a case library and then applied to the layout synthesis.

Based on the presentations of buildings, the objective function (or cost function, fitness function) can be defined to specify the design concerns. Koopmans and Beckmann (1957) formulated the assignment problems of economic activities. Planning was reduced to maximizing the profit function:

$$\sum_{k,i} a_{ki} x_{ki}$$

a_{ki} : constants, the profit from the operation of plant k at location i.
x_{ki} : variables, the fraction of plant k at location i.

This is the simplest form of objective function, corresponding to a lattice representation of architecture. In most design environments, the objective function is much more complicated. For instance, Chouchoulas's (2003) apartment planning program has a number of concerns such as the number of apartments, views, footprint area, height, number of balconies and so forth. Thus, the objective function has to formulate all these concerns and allow different balances between them.

Nonetheless, a big challenge in problem solving is to find the best solution(s) efficiently. It leads to the optimization paradigm. The elementary methods such as Generate-and-Test and Gradient Descent are not feasible for complex tasks. Therefore, the evolutionary computation of advanced search algorithms has gained much attentions. The most common algorithm is the Genetic algorithm developed by Holland (1975). Bentley and Corne (2002) gave a comprehensive overview on evolutionary computation and the applications in design. Impressive applications of evolutionary computation include (Rosenman 1996), (Doulgerakis 2007), and (Menges 2012). Yet，such evolutionary designs pertain to the optimization paradigm, since they construct a fixed design space and then search the best solution(s) with the design space.

Besides the problem solving/optimization framework, case-based reasoning/design[3] learns from cases rather than reason with first principles (Schmitt 1993). However, one of the most frequently discussed models is Frazer's (1995) "Evolutionary Architecture". The Evolutionary Architecture is not meant for applying evolutionary algorithms to architecture, but for treating architecture as evolvable organisms interacting with the environment. Frazer (1995, 9) articulated that:

> "It proposes the model of nature as the generating force for architectural form."

He questioned especially the problem solving approach of Mitchell (1977): "it is notoriously difficult to describe architecture in these terms (problem solving)…the other problem is that any serious system will generate an almost unmanageable quantity of permutations."[4] Frazer's work presented the consciousness of architects in using computers since the 1990s. Architects did not blindly follow the paradigm of other fields anymore; rather, they liked to find their own reasons for using computers in design. The most popular approaches in architecture include shape grammar, parametric design, generative design, and self-organization systems.

Parametric design covers an enormous number of computational models. The parametric model is usually regarded as a precise and efficient apparatus for constructing geometries. By fixing the relationships between the elements of the system, a specific geometry (state) can be produced by responding to a specific set of parameters. Recently, parametric models are often employed to create highly differentiated forms. As Schumacher (2008) put it, "components might be constructed from multiple elements constrained/cohered by associative relations so that the overall component might sensibly adapt to various local conditions. As they populate

---

[3] Kolodner 1992; Maher and Gomez de Silva Garza1997.

[4] Frazer 1995, 14-15.

a differentiated surface their adaptation should accentuate and amplify this differentiation.” The parametric design can be performance-oriented, combined with analysis tools. For instance, the Great Court by Foster partners at the British Museum sought the proof form with high structural performance[5]. Oxman (2008) proposed the performance-based design: “Performance-based design is redefined as the ability to directly manipulate the geometric properties of a digital model on the basis of performative analyses in order to optimize performance...Potentially performance-evaluation can inform parametric model and modify the geometrical model, leading to performance-based generative processes.”

The generative processes for architectural design have been intensively studied in the past two decades. The computational models of generative design are very broad. Caldas (2001) considered the generative system as a search algorithm plus a simulation program. However Chase (2005) gave a more general definition: “a generative design paradigm comprises a formal methodology consisting of rules and procedures to apply them in order to generate designs.” Compared to the optimization paradigm, generative designs emphasize the novelty of the generative process. Usually the generative process is task-specific. A very early example is Frazer’s Reptile program[6], which is supposed to cover various spaces with a set of parametric tiles (Fig 1.1). The filling process starts from a seed consisting of a few tiles. New tiles are added to the seed one by one until the specified area is covered. The resulting form exhibits rich and unpredictable patterns. Many contemporary generative systems are hybrid, i.e., employing various models such as optimization, parametric design, and self-organization. Since 2000, the Chair of Computer-Aided Architectural Design at ETH Zürich developed a series of complex generative systems in various design environments. For instance, the Globus-provisorium project (Hovestadt 2010, 42-49) developed a generative system combining the global optimization procedure with self-organization.

---

[5] Burry and Burry 2010, 123.

[6] The program started in 1966, see Frazer 1995.

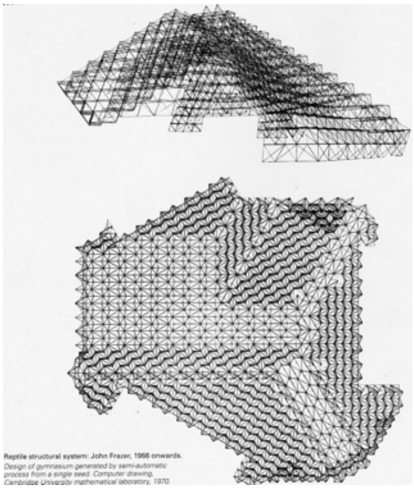


Fig 1.1: The structure produced by the generative process,
Reptile project, John Frazer, 1966 onwards.

1.2 Dilemmas in Modeling

Although models could be very powerful and productive, a number of studies have pointed out that there are dilemmas in modeling that are inherently associated with the nature of models. One controversial issue is the “correctness” of models. There are two distinct attitudes on the issue, the affirmative and the negative. The former supposes that models are genuine representations of the original, thus the modelers should make correct and precise models. The latter holds that the models are essentially incorrect or biased. The affirmative arguments read:

> “A good representation of the actual processes occurring in a real system.” (Oreskes et al. 1994)
>
> “To verify or validate any kind of model means to prove the model to be true.” (Naylor et al. 1967)
>
> “Model verification refers to building the model right; and model validation refers to building the right model.” (Balci 1986)
>
> “The two systems, biological and model, are so related that a homomorphism of the one is isomorphic with a homomorphism of the other.”(Ashby 1957, 104)

However, the negative arguments read:

> “Essentially, all models are wrong” (Box and Draper 1987, 424)
>
> “A cardinal virtue of a theory or model is that it is falsifiable (and of course, not yet falsified).” (Stevens 1990, 285)
>
> “Models are inherently partial.” (Smith 1985)
>
> “A physical theory is just a mathematic model and that it is meaningless to ask whether it corresponds to reality” (Hawking and Penrose 1996, 4)

It is clear that there have been substantial disagreements on the correctness of models. The relationship between the model and its original is still an open problem today. An important problem is that there are multiple models for the same target. These models address the same object but don’t agree with each other. Since we cannot assert which model is correct, we have to live with the coexistence of multiple models. Frigg and Hartmann (2012) mentioned, “Scientists often successfully use several incompatible models of *one and the same* target system for predictive purposes.” Morrison (2011) formulated this as the problem of (multiple) inconsistent models:

> “The problem of inconsistent models often arises because of the limited capacity of its models to account for and explain a system’s behaviour…we usually have no way to determine which of the many contradictory models is the more faithful representation.”

One important motivation of modeling is to benefit from the intelligibility and the exactness of the formal models; however, we have to pay the price when we encounter multiple inconsistent models. Some instrumentalists who care about the usefulness, instead of the correctness of the models, accept multiple models. As Pinsky and Karlin (2011) put it: “The pragmatic criterion of usefulness often allows the existence of two or more models for the same event, but serving distinct purposes.”

In the field of architecture, the problem of multiple models was widely studied. Rosenman and Gero (1996) illustrated that different views of the same object lead to different models (Fig 1.2). Hence, they criticized the traditional CAD systems that imposed fixed and static representations of architecture. They suggested “a set of models where each model has its own concepts and elements.” However, there is no clear solution to the problem of multiple models in architecture.

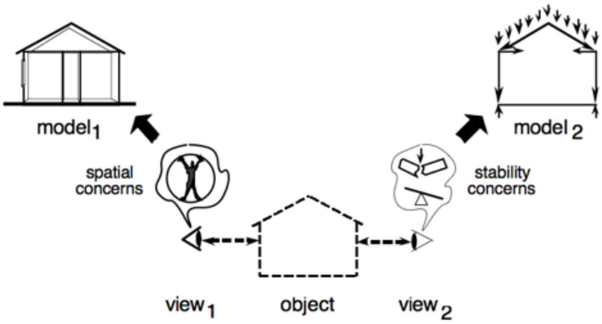


Fig 1.2: Multiple views and models, Rosenman and Gero (1996).

The dominant approach to the problem of multiple models is developing a sing (meta)model—the model of models. The coexistence of multiple views or multiple models seems uncomfortable. As Dave and Woodbury (1990) put it:

> “We feel that it is important to present a single coherent view, and we thus have to embrace a single overall model.”

There have been many attempts to establish a theory of the metamodel and to build practical metamodels. Van Gigch (1991) made a theoretical study and stated, “METAMODELING is MODELING at a higher level of logic and of abstraction…one step further removed from the real world of objects and things.” Usually the notion of the metamodel would lead to an infinite sequence of “world-model-metamodel-metametamodel.” In the field of software engineering, metamodeling often leads to the Unified Modeling Language (UML)[7]. A notable approach in architecture is Building Information Modeling (BIM)[8]. Yet, IFC (Industry Foundation Classes) aims at an open standard for various BIM platforms[9]. Nonetheless, the metamodel approaches are not a final solution to the problem of multiple models. Because the metamodel is again a model and any model is essentially partial. Therefore, metamodeling as modeling cannot be a solution to the dilemmas in modeling.

1.3 Objectives and Scope of the Work

As mentioned in the previous section, there are multiple models for the same target because any model is essentially partial representation of the original. It is not feasible to select the best one of them, for they catch different aspects of the original

---

[7] Bézivin 2005; Kühne 2006.
[8] Eastman and Siabiris 1995; Eastman et al. 2011; Azhar et al. 2008; Penttilä 2007.
[9] Eastman et al. 2011; Pazlar and Turk 2008.

and have distinct purposes (Pinsky and Karlin 2011). One “solution” to the problem is metamodeling, i.e., making a higher-level abstraction of multiple models. However, the metamodel is still a model.

The work asks the question: How to deal with multiple models in architectural design. The research is not going to solve the problem in general; it investigates the problem for architectural design instead. One substantial difference between scientific problems and architectural design problems is that the former is usually well defined while the latter is usually not (Rittel 1973). An architectural project often involves a network of design issues that will constantly evolve. The architect often “invents” new problems and then provides contingent solutions to the problems in design. In Rittel’s (1988) words, “the designer's reasoning appears as a process of argumentation.”

This research is based on two premises: 1) architectural design is usually not wel defined[10]; 2) there are multiple inconsistent models in architectural design. The hypothesis of this thesis is that the coexistence of multiple models is useful for architectural design. Inconsistent models can render design problems vividly. Each model may follow a particular epistemic framework and reflect a particular purpose of the modeler. Thus, multiple models in a design task embody distinct views on the design subject and distinct purposes of design. Suppose the model is the faithful representation of the reality as the scientific realists, multiple models are absurd (at least not preferable) since they present multiple realities. Yet, for the instrumentalists who put emphasis on the cost-effectiveness of the models, multiple models are not promising since accumulating models is not meant to increase efficiency. However, these concerns will not trouble the architects too much, because architectural design usually involves contradictory issues and diverse purposes. It indicates that multiple models instead of a single model are appropriate in design. According to the observations of Rittel (1973), design problems do not have “an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.” We suppose that organizing multiple (probably inconsistent) models could be rewarding in exploring such ill-defined design problems. The objective of this thesis is to investigate how to use multiple models to facilitate and stimulate architectural design.

The research doesn’t contribute to a general model of architectural design; rather, we reveal the paradoxes in modeling and search for better ways of using models.

---

[10] Rittel (1973) enounced the design problems are wicked problems. Coyne (2005) pointed out that design problems are under “redefinition and resolution in different ways over time”. Buchanan (1992) addressed that design has no special subject of matter.

There have been many general models in architecture (either with or without computers). In particular, we study those models that can generate designs through well-defined operations. Early in the 19th century, J. N. L. Durand proposed a successive process of combining architectural elements into architectural designs, taking account of fitness and the economy[11]. When the computer entered the field of architecture, the problem-solving model (constructing a problem space and searching for the best solution within the space) was dominant[12]. Besides, the linguistic approaches to architecture were regarded as the general model of architecture. Mitchell (1990) argued for a critical language referring to the states of the design world. Hence, design is equivalent to selecting/operating the states of the well-defined design world. Among the linguistic approaches, the shape grammar formalism has been widely studied[13]. More recently, many researchers have integrated the generative system and the optimization models such as Performance-based Design[14], Generative Multi-performative Design[15], and Performative Architecture[16]. Nonetheless, none of these general models of architecture is valid for all design circumstances and it is very difficult to tell which one is the best without a given context. So this thesis is not going to make another general model of architecture, it searches for an appropriate way of using models in architecture instead.

The research has three tasks: 1) to examine the notions of models and reveal the dilemmas in modeling, especially the problem of multiple models; 2) to search for a method of managing multiple models for architectural design; and 3) to experiment with the methods by programming and running models in computer.

1.4 Thesis Structure

The thesis contains surveys and discussions on modeling, followed by a series of experiments. In the part of theoretical discussions, we make a historical investigation on the use of numbers, geometry and signs in architecture, because they are the predecessors of modern modeling. Yet, our primary focus is the contemporary notions of modeling and the dilemmas in modeling. Especially, the use of modeling and computing in architectural design is intensively studied. Besides, developing,

---

[11] Durand (1802-5) wrote: “men inevitably sought (1) to derive from their buildings the greatest possible advantage, consequently making them as fit as possible for their purpose; and (2) to build them in the way that would in early times be the least laborious and later – when money had become the price of labor – the least costly.”

[12] Mitchell (1977) summarized the problem-solving approaches.

[13] Stiny 1972; Stiny and Mitchell 1978; Flemming, U. 1990; Knight, T. 1994.

[14] Oxman 2008.

[15] Fasoulaki 2008.

[16] Kolarevic and Malkawi 2005.

running, and testing the models in computers are an important part of the work since we look for the synergy of modeling and computing in architecture.

The first chapter, i.e., this introduction, shows the background of our work, the research question, the objectives, and the scope of the research. The second chapter discusses a few historical topics that are closely related with today's modeling approaches. The topics include the use of numbers, proportion, and geometry in architecture. Chapter 3 introduces the contemporary notions of modeling and various views on modeling. Some important works of employing models and computation in architectural design are discussed. Chapter 4 reveals the dilemmas of modeling and highlights the problem of multiple models. It proposes a new method of managing multiple models, which can be productive for architectural design. Chapter 5 introduces the computer programs developed during the research. They are supposed to test the proposed method of using models. The last chapter gives conclusions of the work and the ideas for future works.

Chapter 2

Modeling without Computer

Nowadays the notion of modeling is associated with computing; however, the notion of modeling is not necessarily bound to computing. Many formal models were already constructed before the birth of computers. However, we can observe that some elements of the contemporary notion of modeling can be traced back to the 19th century or even earlier. Yet, modeling was seldom formally addressed before modern times. Nevertheless, there is a long history of using signs, numbers, or geometry to represent the real world.

Of course, using mathematics and geometry during the ancient times is not equivalent to the modern modeling. On the one hand, the two approaches have something in common because both employ symbolic systems like mathematics or geometry to represent the real world. On the other hand, the epistemic frameworks underlying the two approaches are substantially different. This chapter reviews a few important approaches of mathematics and geometry from ancient times down to early modern times. It is supposed to reflect how the modern approaches of modeling came into being. The historical survey would be helpful to understand the potentials and the limitations of contemporary modeling.

2.1 Pythagoras and Numbers

Pythagoras (c. 570 - c.495 BC) developed the theory of numbers and the musical scales. His most famous contribution might be the Pythagorean theorem. Pythagoras is regarded as a religious teacher, a philosopher and a mathematician. The Greek conception of nature, being, and order were associated with the Pythagoreans. Philolaus (c. 470 - c. 385 BC) is believed to be the oldest Pythagorean whose treatise has survived. Following Pythagoras, Philolaus addressed the great importance of numbers:

> "All things which are known have number; for nothing can be known
> or understood without number" (Kahn 2001, 25)

In his treatise "On the Nature of things," Philolaus interpreted the relationship between the being of things and the thinkable things:

> “The Being (estô) of things, which is eternal, and Nature (physis) itself admit divine but not human knowledge (gnôsis); except that, of the things-that-are (ta eonta) and that are known by us, it was impossible for any of them to have come into being if there was not already the Being (estô) of those things from which the world-order is composed” (Kahn 2001, 26)

Philolaus’s words imply that anything is thinkable, though, only those fitting the mathematical principle can come into being. For example, people are able to think all kinds of right triangles, but only those satisfying $a^2 + b^2 = c^2$ (Pythagorean theorem) can come into being. This is definitely different from today’s interpretation: all right triangles satisfy $a^2 + b^2 = c^2$. For the Pythagoreans, the things must be made by what the world-order (kosmos) is made from. In addition, the world-order is composed of the mathematical principle, or simply numbers. As Aristotle commented on the Pythagorean theory of numbers:

> “Since of these principles (of Pythagoreans) numbers are by nature the first, and in numbers they seemed to see many resemblances to the things that exist and come into being - more than in fire and earth and water... all other things seemed in their whole nature to be modelled on numbers, and numbers seemed to be the first things in the whole of nature, they supposed the elements of numbers to be the elements of all things” (Aristotle 350BC, part 5)

The numbers also play the significant role in the Pythagorean theory of musical scales. The musical scales relate to the numbers, the music, and the harmony together. The theory has two basic points: first, the tune of the sound corresponds to the length of the string; second, harmonious combination of sounds can be produced by a series of ratios 1:2:3:4 (of the length of the string). The ratios are octave (1:2), double octave (1:4), fifth (2:3) and fourth (3:4) (Steven 1990). Pythagoreans believed that the beauty of music came form the harmony of numbers, i.e., the mathematical principle.

It implies that the Pythagoreans modeled everything with numbers. We may call the numbers or the mathematical principle “model,” but we have to distinguish such model with today’s notion of model. For the Pythagoreans, the mathematical principle is of absolute truth; however, today’s scientific models are susceptible to empirical data.

The numbers were often bound to specific meanings in Ancient Greece. Number 1 stands for reason, number 2 as an even is female, number 3 as an odd is male (Steven 1990). Thus, the number 5=2+3 means the marriage of female and male. Number 3, 4 and 5 make $3^2 + 4^2 = 5^2$ with great harmony. The number 10, called Tetractys, was believed to be perfect by the Pythagoreans. It can be constructed by 1+2+3+4=10; the illustration of this construction appears in Raphael's The School of Athens. As mentioned above, the number 1, 2, 3, and 4 are also the elements of harmonies (1:2, 2:3, 3:4). Plato argued the seven divine numbers are 1, 2, 3, 4, 9, 8, and 27. They can be written in a Lambda shape:

```
      1
     2 3
    4   9
   8     27
```

Since the theory of numbers had influences on the Greeks, the Romans and even the Renaissance architects, Wittkower (1949) commented on the theory of divine numbers:

> "(the seven numbers) embrace the secret rhythm in macrocosm and microcosm alike. For the rations between these numbers contain not only all the musical consonances, but also the inaudible music of the heavens and the structure of the human soul."

Before modern times, the numbers often played two kinds of roles[1]. First, some numbers in architecture have static connotations (say, 3 for trinity), hence, they can play a role in architecture independently. Second, the numbers are put into one consistent proportioning system so that each number serves as a part of the system. The use of numbers in the proportioning system is further discussed in the next section.

2.2 Proportion

Architects have employed proportion in architecture since Ancient Greece. However, we have to notice that the "proportion" has distinct meanings through the history. This section compares Vitruvius's notion of proportion, the Renaissance theory of proportion, and the early modern views on proportion.

In Ten Books on Architecture Vitruvius did not explicitly write about modeling. Yet, he explained how the architectural forms should imitate nature. For example,

---

[1] Except for technical use for measuring, constructions and so on.

when addressing why the columns of the upper tier should be smaller than the lower ones in ancient forum (Fig 2.1), he gave two reasons. The first reason is related to load bearing. The second reason refers to the imitation of nature:

> "We ought to imitate nature as seen in the case of things growing; for example, in round smooth-stemmed trees, like the fir, cypress, and pine, every one of which is rather thick just above the roots and then, as it goes on increasing in height, tapers off naturally and symmetrically in growing up to the top. Hence, if nature requires this in things growing, it is the right arrangement that what is above should be less in height and thickness than what is below." (Vitruvius 1960, 132)

Fig 2.1: The columns of the forum in Pompeii.

It seems that Vitruvius applied one aspect of the form of natural plants to the form of architecture. This is not modeling though we can say it involves a verbal model for the characteristics of the referent (the form of plants in this case) are not formulated in a formal system. Besides, Vitruvius mentioned many times the principle of proportion in architecture. For example:

> "Without symmetry and proportion there can be no principle in the design of any temple." (Vitruvius 1960, 72)

It is obvious to him that the symmetry and the proportion is not an issue of visual aesthetics; rather, it is the principle of architecture. This idea can be traced to Ancient Greece and it is further studied by the Renaissance architects. He especially associated the proportion of architecture with the human body. This entry was handed down to the architects and the theorists in the Renaissance and even in modern times. Vitruvius stated that:

> “For the human body is so designed by nature that the face ... is a tenth part of the whole height...Similarly, in the members of a temple there ought to be the greatest harmony in the symmetrical relations of the different parts to the general magnitude of the whole. ” (Vitruvius 1960, 72)

Here Vitruvius regarded the proportion of architecture as the relationship between the magnitudes of the parts and that of the whole. The proportion of the human body was of great importance and it is directly relevant to the proportion of architecture. It is interesting to see how he explained the origins of the three orders, for example the Doric:

> “On finding that, in a man, the foot was one sixth of the height, they applied the same principle to the column... Thus the Doric column, as used in building, began to exhibit the proportions, strength, and beauty of the body of a man.” (Vitruvius 1960, 103)

It suggests that the proportion of the Doric order is the same as that of a man. Because of this, the Doric order exhibits the same character of a man. The proportion as a mathematical system governs the relationship between the parts and the whole in architecture, and subsequently endows the architecture with the essential characters. Hence, we may say the proportion is a “model,” but obviously, such notion of model differs from the modern notion of model.

The Renaissance architects intensively studied the theory of antiquity and they made new processes for employing mathematics and geometry in architecture. To a certain degree, they inherited the theory of the proportion of antiquity, but they established their own systems of proportion. Nevertheless, the way that Renaissance architects used mathematical systems definitely differs from the modern approaches (especial from 19th century).

It is well known that the Renaissance architects emphasized the reasoning of proportion in architecture. As Wittkower (1949 104) put it:

> “The conviction that architecture is a science, and that each part of a building, inside as well as outside, has to be integrated into one and the same system of mathematical ratios, may be called the basic axiom of Renaissance architects.”

According to Alberti, the building should follow a “consistent method and art” (Alberti 1443-52), though the forms of buildings always vary with particular criteria and situations. In his “On the Art of Building”, Alberti argued the form of architecture is subject to the law of “concinnitas”:

> “Everything that Nature produces is regulated by the law of concinnitas… Beauty is a form of sympathy and consonance of the parts within a body, according to definite number, outline, and position, as dictated by concinnitas, the absolute and fundamental rule in Nature. This is the main object of the art of building. ” (Alberti 1443-52)

For him the “concinnitas,” the rule of nature, regulates the form of architecture by means of numbers and geometry. The task of the architect is applying the rule of nature to architecture through experiments and reasoning. It is a question of whether the proportion underlying the building can be effectively conceived by the people inside the building. However, for Alberti, the proportion system is for the absolute value instead of some observable quality. As Wittkower (1949, 18) put it:

> “It is obvious that such mathematical relations between plan and section cannot be correctly perceived when one walks about in a building. Alberti knew that…We must therefore conclude that the harmonic perfection of the geometrical scheme represents an absolute value, independent of our subjective and transitory perception.”

Following the texts of Vitruvius, many painters and architects developed their own proportioning systems based on the human body. For example, Fra Giocondo, Giorgio Vasari, da Vinci and Alberti constructed their “Vitruvian man.” These reconstructions are their own interpretations rather than Vitruvius’s original concept. We can observe a significant difference between Leonardo da Vinci’s illustration and the others’ illustrations (Fig 2.2). In most illustrations, the background seems homogeneous to the figure. In other words, the figure is absolutely coupled with the space. By contrast, da Vinci decouples the figure from the space so that the abstract space can accommodate multiple figures. As in his illustration (Fig 2.2 right), the circle and the rectangle fit two different men respectively, or fit two different gestures of one man respectively. This novel arrangement reflects a new understanding of the relationship between the individuals and the one order.

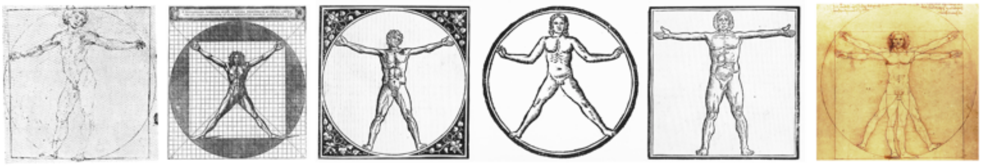


Fig 2.2: Comparison of the illustrations of Vitruvian man (Wittkower 1949).
from left to right:
(1)Francesco di Giorgio, Codex Ashburnham 361
(2)Cesare Cesariano's edition of Vitruvius, 1521
(3)Fra Giocondo's edition of Vitruvius, 1511
(4)Francesco Giorgi, De harmonia Mundi, 1525
(5)Fra Giocondo's edition of Vitruvius, 1511
(6)Leonardo da Vinci, 1487

It seems that the arrangement of multiple overlapping figures (like the Vitruvian man of da Vinci) had influences on Renaissance architects. For instance, in S.Francesco della Vigna, Palladio applied two pediments in two different scales on the facade (Fig 2.3), in order to adapt the high nave in the center and the lower aisles at two sides. The whole facade is thus governed by a novel system consisting of two classical orders, i.e., two proportioning systems. On the one hand, it still follows the order of antiquity; on the other hand, it is subject to the architect's novel composition. Through the sophisticated overlapping scheme, Palladio tried to make a linkage between the particular buildings and the general principle of architecture. Both Palladio's composition and da Vinci's illustration imply that the absolute order can accommodate multiple individuals.

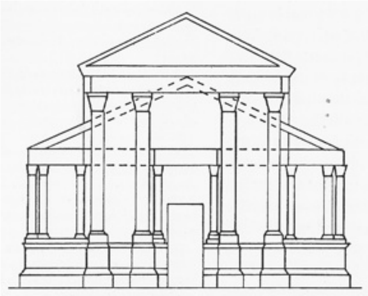


Fig 2.3: The two pediments "overlap" in the facade of S.Francesco della Vigna, Palladio (Wittkower 1949).

The Renaissance people on the one hand sought the proportioning system of absolute value, but on the other hand, they individually created distinct systems. The masters often developed their own proportion system in architecture rather than followed a certain static proportion. Instead of specifying proportion directly, they employed arithmetic processes to yield the proportion. For example, Alberti developed the double progressions of Plato (Scholfield 1958):

1 2 4 8 16 ...
3 6 12 24 ...
9 18 36 ...
27 54 ...

With modern notation, the system can be written as:

$$a_{ij} = \frac{3^i \times 2^j}{6}$$

i: the index of row
j: the index of columns

Different from Alberti's system, Palladio employed the combinations of the progression 1, 2, 3, 4, 5 and the progression 1, 2, 3 as a proportion system. His favorite ratio of 1:1, 1:2, 2:3, and 3:4 (Fig 2.4) can be derived from this system. This system is quite practical as it can be applied to the rooms directly. According to Rowe (1947), Palladio's villa followed a rhythm of 4:2:4:2:4 (called the ABABA system) in one direction and 1:3:3:3:1 in other direction. Rowe's observation is consistent with Palladio's progression system.

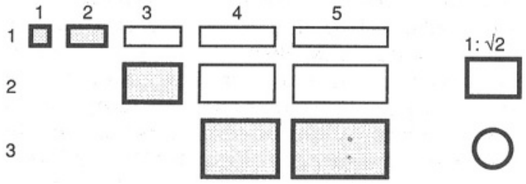


Fig 2.4: Palladio's proportion system (Stevens 1990).

Now we can see how Palladio applied his proportion system to the floor plan. In Villa Emo at Fanzola (Fig 2.5), the dimensions of the central rooms are 12×16, 16×16, 16×27 and 27×27. Besides, the number 3, 9, 12, 24, 48 appear in the two sides of the plan. Thus, there is a series of numbers: 3, 9, 12, 16, 24, 27, 48, which can be interpreted by both musical scales and Plato's progression (Wittkower 1949). First, the series of 3:9:27 and 12: 24: 48 are part of Plato's divine numbers. Second, it fits musical scales: 9:12 and 12:16 are fourth (3:4); 12:24 and 24:48 are octaves (1:2); 16:24 is a fifth(2:3); 24:27 is a Major tone (8:9).

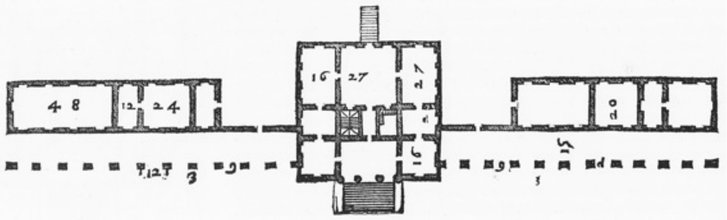


Fig 2.5: The proportion in the floor plan of Villa Emo at Fanzola, Palladio (Wittkower 1949).

To sum up, there are several aspects of Renaissance proportion systems. First, each system is particular. Second, each system is supposed to be universally valid (Pérez-Gómez, A.: 1983) since they are based on the absolute principle. Third, the way that the proportion system is applied to the buildings depends on the particular situations of the design task. If we regard the proportion as a sort of model, then applying such model to architecture is task-specific.

After the Renaissance, the study of architecture had to face the strong impacts from science. Some theorists and architects reformed the traditional methods in order to defend them, yet some established new theories as positive responses to science. Claude Perrault's view on proportion presented a turning point on the theory of proportion in the 17th century. Perrault is famous for his design of Colonnade in the Louvre Palace; however, he is not only an architect but also a scientist and a mathematician. On the relationship between proportion and beauty (sensuous quality), Perrault (1683) addressed:

> "What pleases the eye cannot be due to a proportion of which the eye is unaware."

This argument is in contrast with the traditional view that the human experience and the order of nature are intrinsically unified. Perrault also pointed out that the proportions of ancient architecture are not consistent. He believed that people would choose different proportions according to their cultures and customs. Perrault's (1683) idea on proportion is clear in the following text:

> "The beauty of a building, like that of the human body, lies less in the exactitude of unvarying proportion and the relative size of constituent parts than in the grace of its form, wherein nothing other than a pleasing variation can sometimes give rise to a perfect and matchless beauty without strict adherence to any proportional rule."

This argument reveals the gap between the visual quality and the underlying proportion of architecture. By contrast, Nicolas-Francois Blondel defended the classical principle of architecture, and especially questioned Perrault's point on proportion. In Steven's (1990, 229) words:

> "Blondel fears that Perrault's arbitrary proportioning removes the necessary nature of architectural aesthetics - if architecture does not have stable and unvarying principles, then what is the point of it all?"

Pérez-Gómez (1983, 44-45) gave a direct comparison between the two:

> "(Perrault) using it (number) as an operational device, as a positive instrument for simplifying the process of design or avoiding the irregularities of practice…
> Blondel maintained that geometry and proportion, being transcendental entities, guaranteed the highest architectural meaning."

Perrault's speculation presented a new understanding of using proportion in architecture. For him the sensuous quality of architecture is decoupled from the proportion—the traditional principle of architecture. Thus, the proportion is not the principle of architecture anymore; rather it becomes a mathematical tool subject to the designer's purposes.

In the 18th century, Claude Nicolas Ledoux and Étienne-Louis Boullée abandoned the classical orders and employed regular volumes in architecture instead (Fig 2.6). The proportion lost its transcendental meaning in their designs. In agreement with Perrault, Boullée (c. 1794, Rosenau) denied the proportion as the basic principle of architecture:

> "It is thus evident that although proportion is one of the most important elements constituting beauty in architecture, it is not the primary law from which its basic principles derive."

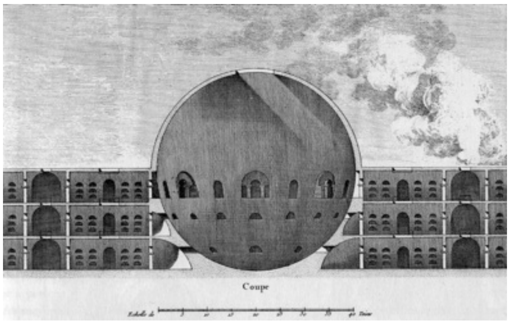


Fig 2.6: Section, Claude Nicolas Ledoux (1804).

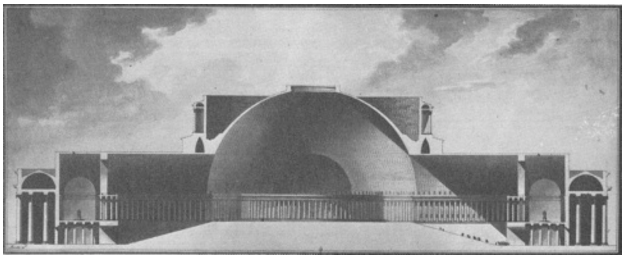


Fig 2.7: Section, Étienne-Louis Boullée (Madec 1989).

The notion of "proportion" for Boullée is not a traditional one; rather, he defined his own notion of proportion:

> "By the proportion of a volume, I mean the effect produced by its regularity, its symmetry and its variety. Regularity gives it a beautiful shape, symmetry gives it order and proportion, variety gives it planes that diversify as we look at them. Thus the combination and the respective concord which are the result of all these properties, give rise to volumetric harmony." (Boullée c. 1794, Rosenau)

It implies that the proportion is the "property" of architectural volumes and that it depends on the disposition of the volumes. As a result, there must be "good" proportions and "bad" proportions. Hence, he explained what a "good" proportion is:

> “As in nature, the art of giving an impression of grandeur in architecture lies in the disposition of the volumes that form the whole in such a way that there is a great deal of play among them... The arrangement should be such that we can absorb at a glance the multiplicity of the separate elements that constitute the whole.” (Boullée c.1794, Rosenau)

Boullée held a very different idea on proportion compared to the Renaissance architects like Alberti. For Alberti, the proportion stands as “the absolute and fundamental rule”[2] in nature and in architecture; thus, a good design must derive itself from this absolute rule. By contrast, Boullée held that the proportion is a property of the disposition of volumes. In general, during modern times the numbers and mathematics gradually lost its transcendental value in architecture. This transformation might be regarded as the modern architects’ responses to the science.

2.3 Geometry

An important contribution from Ancient Greece is the Euclidean geometry. Euclid put various early geometric works into one framework. Euclid’s “Elements” does not only deal with geometry, but also introduces the axiomatic system for proofing. Concerning Euclid’s works, Knorr explained the process from the “intuitive” discovery in geometry to the formal axiomatization:

> “A mathematical theory, like that of incommensurable magnitudes, will pass from the heuristic stage of its first discovery, to the formal axiomatic stage (as in Elements X), via n ‘informal’ or ‘partially formalized’ intermediate stage.” (Knorr 1975, 13)

Euclidean geometry contributed a system of formal reasoning (axiomatic system). It seems that Euclidean geometry has two levels of representation. First, the objects in the real world are represented by geometric objects like points and lines. Second, the geometric objects are further represented with a formal language in that the methods of proofing can work rigidly.

Vitruvius’s texts imply that the Euclidean geometry was used for specifying a certain prescribed order of architecture. The absolute value is embedded in certain geometrical figures. For example, Vitruvius introduced that the plan of the Greek amphitheater fits three rectangles, while the Roman amphitheater matches four triangles (Figure 2.8). It indicates that Roman architects did not reason about or mod-

---

[2] Alberti, 1443-52

ify these prescribed geometrical figures, compared to the Renaissance architects who could make individual investigations on the geometry.

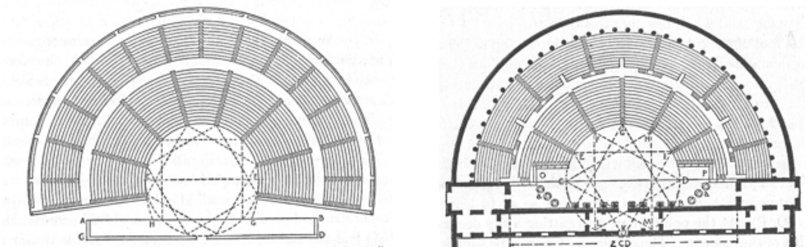


Fig 2.8: Comparison of Greek theatre (left) and Roman theatre (right). (Vitruvius 1960).

During the Renaissance, the geometry played new roles in architecture. The invention of the plane perspective and the use of projective geometry were essential for the Renaissance architects. The perspective can be regarded as an application of projective geometry. From a historical point of view, perspective rationalized our optical view of the world. It is easy to observe that the objects seem smaller when they move away from us, and the parallel lines seem to intersect at a distant point. However, it was impossible to establish a uniform theory or technique to explain all these observations until the fifteenth century. Filippo Brunelleschi was considered as the inventor of perspective. We can consider his perspective as an invention instead of a discovery. Because the plane perspective is "fabricated," and there are other perspective systems besides the dominant plane perspective. White (1949) mentioned three kinds of perspectives reflected by the texts and paintings during 14-16th century. The dominant perspective is called the Artificial Perspective by White (1949):

> "The Artificial Perspective, of which Brunelleschi was the probable inventor, is essentially a mathematical way of constructing space on a flat surface, and at the same time achieving a pictorial unity. "

It indicates the two aspects of Brunelleschi's perspective: the mathematical construction and the comprehensive result. Unfortunately, no writings or drawings of Brunelleschi have survived. Leon Battista Alberti, who was close to Brunelleschi, formulated the perspective in his De pictura (on Painting, 1435). Wright (1983) briefly explained Alberti's method:

> "Alberti regards a picture as a window through which a fixed observer

> sees the outside world… Each ray to the eye, from a point on the object, intersects the picture plane and there determines the position of the point on the picture.” (Fig 2.9)

Alberti illustrated how to make a perspective drawing of a grid (Fig 2.9). There is a vanishing point on the level of the eye. This is based on the observation that the vanishing point is always in the direction of the sight. The vertical lines of the grid are easy to draw for they all go to the vanishing point. The positions of the horizontal lines are determined by the intersections between the rays (from the eye to the horizontal lines of the grid) and the picture plane.

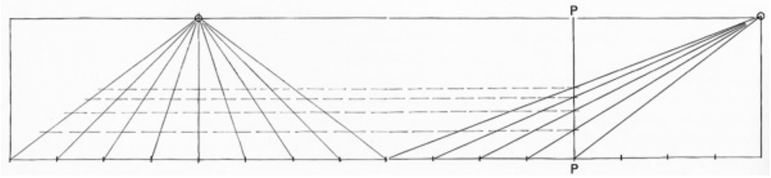


Fig 2.9: Alberti’s method of perspective (Wright 1983.)

Through the method of perspective, any objects observed in the world can be projected onto a plane. We can get different geometry in the image plane with different viewpoints or with different positions of the image plane. In other words, the method of perspective unifies the infinite individual views of the object.

Besides perspective, the method of projective geometry has substantial influences on architectural drawings. Architectural drawings were not only for making pictorial renderings as final results but also for developing the design when the design proceeds. Some Renaissance architects used the projective geometry in a way that some parts of the drawing were “imaginary.” For instance, Sebastiano Serlio (1475 - c.1554) constructed an “imaginary” circular body beneath the main scene (Fig 2.10). This circular body refers to nothing in the real world, and is only for facilitating the drawing of the two arches in the main scene.

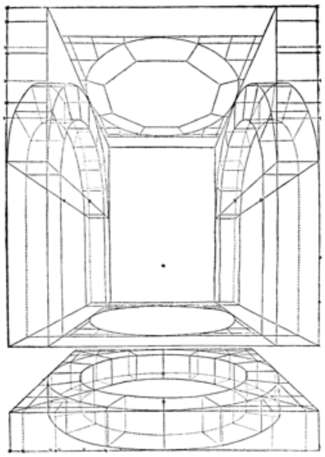


Fig 2.10: Drawing arches, Sebastiano Serlio (Hart and Hicks 1996, 65)

Serlio's drawing implies that the geometrical entities sometimes represent nothing real in the architecture; rather, they can be abstract tools for contrasting other geometrical entities. Compared to Greek and Roman architects who used particular geometry for describing the prescribed order of architecture, Serlio represented desired space/objects with a consistent geometrical system.

Descartes's mathematical speculations changed the landscape of geometry; he built the connection between geometry and algebra. In his time, many people held that geometry was superior to algebra in terms of the exactness of reasoning. In contrast, Descartes (1637, 5) stated: "I shall not hesitate to introduce these arithmetical terms into geometry, for the sake of greater clearness." For this, he developed a new kind of geometry that we called analytic geometry today. The Cartesian coordinate system that is very popular today is also attributed to his algebraic approach to geometry.

In La Géométrie (1637), Descartes introduced the method of representing the magnitude of the geometry entities with algebra. In one of his examples, he represented the position of a point on the curve with x while y. x denotes the length of AB, y denotes the length of BC (Fig 2.11). Then an equation with x and y is constructed to describe all points on the curve. He briefly explained the algebraic approach to geometry:

> "If, then, we wish to solve any problem, we first suppose the solution already effected, and give names to all the lines that seem needful for its construction… making no distinction between known and unknown lines, we must unravel the difficulty in any way that shows most naturally the relations between these lines, until we find it possible to express a single quantity in two ways. This will constitute an equation." (Descartes 1637, 6-9)

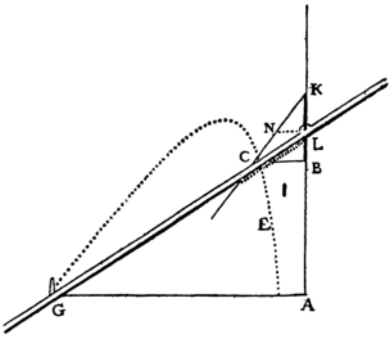


Fig 2.11: Locating the positions of the points on the curve, with x (AB) and y (BC). Descartes (1637).

Descartes emphasized the exact correspondence between geometry and algebraic equations. He prefers the latter:

> "The best way to group together all such curves and then classify them in order, is by recognizing the fact that all points of those curves which we may call "geometric", that is, those which admit of precise and exact measurement, must bear a definite relation to all points of a straight line, and that this relation must be expressed by means of a single equation." (Descartes 1637, 48)

He grounded the exactness of geometry on the algebra (not the other way round). Since Descartes, analytical geometry has played an important role in mathematics and physics. Nowadays the Cartesian coordinate system serves as a standard system for describing geometry. It is a significant paradigm shift after the long tradition of Euclidean geometry.

Many contemporary architects are familiar with D'Arcy Thompson's "On Growth and Form" (1917). Thompson employed coordinate systems to compare different forms of organisms, or to transform one form into another (Fig 2.12). He termed the method "Co-ordinates":

> “This process of comparison, of recognising in one form a definite permutation or deformation of another, apart altogether from a precise and adequate understanding of the original 'type' or standard of comparison, lies within the immediate province of mathematics. This method is the Method of Co-ordinates.” (Thompson 1917/2004, 271)

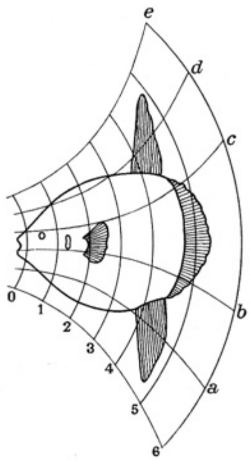


Fig. 155. *Orthagoriscus.*

Fig 2.12: Transformation of shapes through the variable coordinate system. Thompson (1917, 301).

We can observe the connection between Descartes’s coordinate system and Thompson’s application of the system. Descartes’s original motivation is to represent the curves in the space with algebra; however, Thompson represented the space instead of the objects. In other words, Thompson developed a uniform method to describe the differences between the forms.

2.4 Durand’s Method

In the early 19th century, J. N. L. Durand developed a new design method that has had significant influences up to today. He addressed the goal of architecture, the method of design, and criticized some traditional theories of architecture. He asserted that fitness and economy instead of pleasure were the goal of design:

> “Fitness and economy are the means that architecture must naturally employ, and are the sources from which it must derive its principles: the only principles that can guide us in the study and exercise of the art.”(Durand 1802-5, 84)

On fitness and economy, Durand is similar to today's engineers. Before introducing his design method, he clarified the nature of architecture. For him the building is just the assembly of parts:

> "Any building as a whole is not and cannot be other than the result of the assembly and combination of a greater or lesser number of parts." (Durand 1802-5,188)

Durand made a collection of the elements of buildings. In his drawings, the architectural elements (e.g. Fig 2.13) are neither pictorial renderings nor faithful representations of real architectural elements, rather they are signs. Therefore, how an element is drawn on the paper is not essential in his design method.

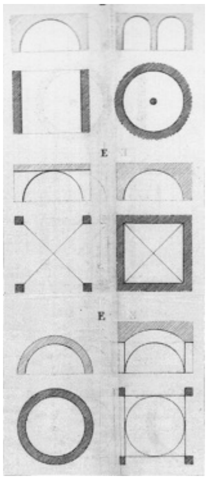


Fig 2.13: Elements of building, J. N. L. Durand (1802-5).

After defining the goal of design and the nature of buildings, Durand introduced a design process for achieving the goal. Since the elements are represented by signs, the main task of design is assembling these signs in croquis. Durand illustrated how to use a series of croquis to develop the design. For example, in the first croquis he makes a rough plan (Fig 2.14, top-left): placing one courtyard in the center, four rectangular rooms on four sides and four squares rooms in four corners. These spaces are represented by simple signs. Then on the second croquis (Fig 2.14, middle-left), a few principal axes are added to the plan for adding more details into each space. In the third croquis (Fig 2.14, bottom-left), more axes are added to define the positions of columns. Finally, detailed plans are created through a series of operations on the croquis. Certain variations in one (or more) of the croquis would lead to distinct results.

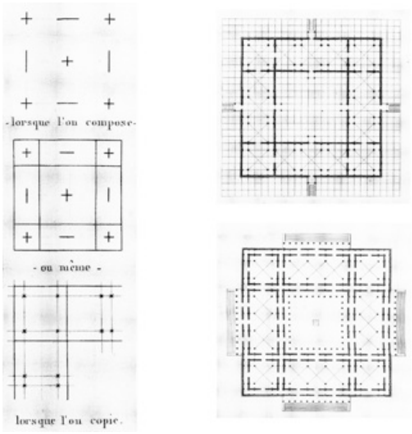


Fig 2.14: Developing plan by a series of croquis, J. N. L. Durand (1802-5).

Durand's method emphasizes the diversity of the operations (adding axes). Each result is obviously specific; the idea is achieving generality through the combinations:

> "Principal axes, might be combined in a thousand entirely different but equally simple ways; that one might apply all the elementary combinations in turn to each of the numerous general dispositions that result from such combinations, and consequently obtain, by a kind of supercombination, a host of different plans." (J. N. L. Durand 1802-5,140)

We can easily find out the combination have freedom in many levels, at least three. The first level is in the initial croquis. If we change signs in this initial croquis, the results would be significantly different (comparing Fig 2.14 and Fig 2.15). The second level is in all the succeeding croquis. Applying different axes leads to different results. The third level is in the final step: applying the detailed elements to the croquis to finish the drawing. Since the signs in the croquis are abstract, the architects still have the freedom to choose the exact architectural element for each sign in the final stage.

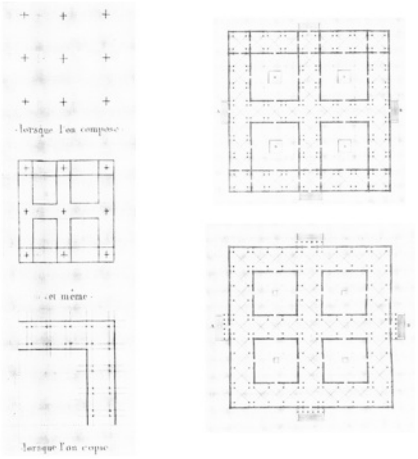


Fig 2.15: Developing plan by a series of croquis, No.2, J. N. L. Durand (1802-5).

To summarize, Durand contributed to several concepts that are connected with some contemporary architectural theories: (1) Building is no more than the assembly of elements; (2) The goal of design is fitness and economy; (3) Design can be organized as a successive process—a design process. (4) Buildings can be represented by signs to facilitate the design process. Durand had developed a model of design. This model does not only represent the buildings but also formalizes the design process. His model is very similar to today's shape grammar formalism (e.g. The Logic of Architecture, Mitchell, 1990). Pérez-Gómez (1983, 304) stated "Architectural design as a whole was reduced in Durand's theory to a formal game of combinations, devoid of transcendental intentions. Meaning was to be derived from within the system." This "formal game" makes sense for Durand. Because his model of combination is a sort of neutral tool that can be used in any ways, so the architect's task is making the right operations with this tool to achieve his/her own goals.

Chapter 3

Modeling with Computers

> “No computation without representation.” (Smith 1985)

In the 20th century, computer science emerged as a new important discipline. Using computers became practical in the second half of the century. Computing has had great influences on all scientific fields. Up to now, computing also changed the way that architects made designs and shifted the understandings of tools in design. Computer science has contributed new ingredients to the concept of modeling, for computer science does not only develop hardware and software of computers but also asks profound questions such as “can machines think?” The contemporary notions of “information”, “data”, “learning”, “reasoning” as well as “modeling” have been re-shaped by computer science. Besides, computer science has coined many terms that are well known today: machine learning, pattern recognition, artificial intelligence, etc. Moreover, a bundle of interdisciplines have arisen, e.g., computational biology, computational chemistry, computational linguistics, as well as computational design. The computing does not only serve as supporting tools for other fields, but also gradually shifts the way that people perceive and solve problems in their own fields.

The forerunners of Computer-Aided Architectural Design employed computers in the 1960s. CAD systems have been become popular in architecture since the 1980s. More and more innovative methods of using computing in design have emerged in the last two decades. There were two lines of using computers in architecture, one was scientific research originated by computer specialists and theorists; the other was developed by architects based on modeling software and scripting / programming tools. However, the two lines have blended with each other especially during the last decade. At the beginning, CAD software provided a powerful drawing tool for architects. Later architects found that there are many other ways of using computers besides drawing. Right now, it is impossible to figure out a general method of using computers in architectural design.

This chapter is arranged as follows: Section 3.1 briefly reviews the ideas of modeling in computer science. Section 3.2 discusses the modeling in scientific fields. Section 3.3 focuses on modeling in architectural design.

3.1 Modeling and Computing

3.1.1 Turing machine

Turing explored some basic concepts about computers before the practical computer came into use. Turing (1950) argued that whether a computer is mechanical, electrical or digital is not critical, since these machines can be categorized as “discrete state machines”. Such machines take certain signals as inputs, alter its internal states according to predefined rules with the current inputs, then gives outputs mapped from the current internal states. Later, Turing’s formulation is summarized as the “Turing machine.” As Savage (1998) put it, “The standard Turing machine consists of a control unit, which is a finite-state machine, and a (single-ended) infinite-capacity tape unit.”[1] The amazing thing about digital computers, like today’s personal computers, is that they can implement any specific specifications on inputs, outputs, internal states, and transition rules. As a result, digital computers are termed by Turing as universal machines.

Based on the definition of a discrete state machine, Turing criticized several common arguments on computer. Based on Gödel's theorem one argument reads: “In any sufficient powerful logical system statements can be formulated which can neither be proved nor disproved within the system, unless possibly the system itself is inconsistent” (Turing 1950). Turing pointed out that both the human mind and computers have limitations. Thus, a particular limitation on one side does not imply one is inferior to the other. Another debate is about whether computers can learn (something new). On Baggage’s general-purpose computer “Analytical Engine”, Lovelace (1842) noted, “the Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform.” In contrast with Lovelace, Turing stated that there is no proof that the machines do not have the property of learning new behaviors. He proposed a sort of learning technique that is called supervised learning nowadays. He even mentioned that machines can learn to play chess.

Furthermore, Turing (1952) originated the concept of morphogenesis based on computing. Turing made two points: First, organic forms can result from a reaction and diffusion between two “chemicals” in a discrete space and time. Second, such reaction-diffusion process can be formulated explicitly and can be implemented by machines. One result of Turing’s system is shown in Fig 3.1. Many years later morphology is intensively discussed in computational architectural design (e.g. Cotes and Makris 1999; Hensel, Menges, and Weinstock 2004; Leach 2009).

---

[1] Savage 1998, 210

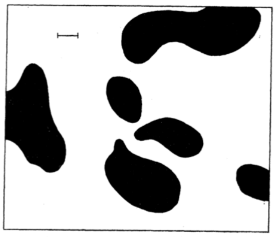


Fig 3.1: Turing's morphogen system (reaction-diffusion system).

It is worth paying attention to Turing's special attitude to computational models. When explaining machine learning, he said, "One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse" (Turing, 1950). It suggests that we have to try first then to know the right way. Turing (1952) also mentioned, "one gets results for particular cases" in his morphogenesis model. Moreover, he suggested that it is not feasible to apply theoretical analysis to predicate the results. The "inexactness" of computation probably made people nervous sixty years ago; however, this aspect of computation has become one of the most inspiring characters of computing nowadays.

3.1.2 Definition of modeling

Smith (1985) addressed the relationship between a computer, a model and the real world: "When you design and build a computer system, you first formulate a model of the problem you want it to solve and then construct the computer program in its terms. (Fig 3.2)"

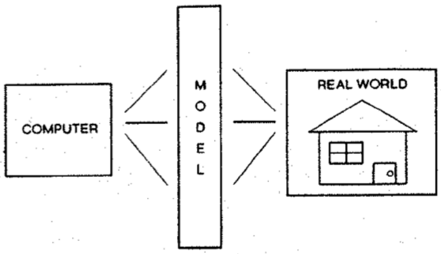


Fig 3.2: Computer, model and the real world, Smith (1985).

Gigch (1991) gave emphasis on the role of "abstraction" of models, with respect to the model's original:

> "Modeling or to model implies that the modeler 'abstracts' properties from things in order to obtain a representation of the physical world...the model stands at one level of abstraction higher than the things from which the properties are obtained."

From the point of view of computer scientists, Kleppe et al. (2003, 16) gave a clear definition of a model:

> "A model is a description of (part of) a system written in a well-defined language. A well-defined language is a language with well-defined form (syntax), and meaning (Semantics), which is suitable for automated interpretation by a computer."

According to Stachowiak (1973, 131-132), models have three features:
1. Abbildung (the model represents its referent)
2. Verkürzung (the model only represents part of the aspects of the referent)
3. Pragmatismus (the model has its own goals, which may be not clearly subject to its referent)

Smith (1985) questioned the relationship between the model and the real world. His conclusion was that we do not have any theories about it because there is no good way to examine it. What we can explicitly examine is the relationship between the model and the desired system. This relationship is illustrated by Kleppe (2003) in Fig 3.3.

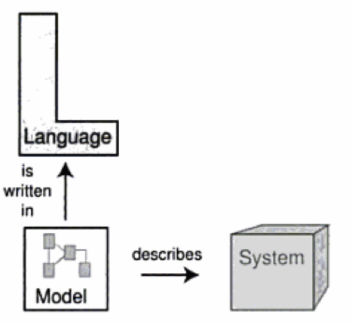


Fig 3.3: Model, system and language, Kleppe (2003)

In accordance with Kleppe, Kühne (2006) on the one hand asserted that the target of a model is a system, and on the other hand distinguishes the realistic system from "imaginary" system. Kühne's dictionary entry reads:

> "Model: a theoretical projection of a possible or imaginary system"

3.1.3 Views on computational models

People make “models” or “representations” of our world in order to solve problems with machines. A list of arguments on models and representation are listed below. Most of them are closely associated with problem solving and machine intelligence.

> “A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality…Any model is characterized by three essential attributes:
> 1. Reference: It is of something (its "referent")
> 2. Purpose: It has an intended cognitive purpose with respect to its referent.
> 3. Cost-effectiveness: It is more cost-effective to use the model for this purpose than to use the referent itself.”
> (Rothenberg 1989)
>
> “Every model deals with its subject matter at some particular level of abstraction, paying attention to certain details, throwing away others, grouping together similar aspects into common categories” (Smith 1985)
>
> “Computational models: These consist of a process-oriented description in terms of a set of data structures and algorithms.” (Steels 1994)
>
> “Every problem-solving effort must begin with creating a representation for the problem - a problem space in which the search for the solution can take place.” (Simon 1996, p.108)
>
> “On this basis we shall say that an entity is intelligent if it has an adequate model of the world (including the intellectual world of mathematics, understanding of its own goals and other mental processes)…”
> (McCarthy and Hayes 1969)
>
> “According to this definition intelligence has two parts, which we shall call the epistemological and the heuristic. The epistemological part is the representation of the world ... The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do. ” (McCarthy and Hayes 1969)

> "We now have models of the symbol level that describe how information processing agents arrive at actions by means of search - search of problem spaces and search of global data." (Newell 1982)
>
> "In common with the mainstream of problem solving and reasoning systems in AI, SOAR has an explicit symbolic representation of its tasks, which it manipulates by symbolic processes." (Laird, Newell and Rosenbloom 1987)
>
> "Perhaps the most straightforward concept of planning is that of using a simplified model of the problem situation." (Minsky 1961)
>
> "The two systems, biological and model, are so related that a homomorphism of the one is isomorphic with a homomorphism of the other. The higher the homomorphisms are on their lattices, the better or more realistic will be the model. "(Ashby 1957, 104)
>
> "Our strategy for choice of formalism and mechanism for representation depends upon our assumption that there is no 'real' or 'true' representation of knowledge or information, but rather many possible representations, each appropriate to particular problems." (Belkin, Oddy and Brooks 1982)
>
> "If the computer can manipulate an explicit model of the goals and potential actions, then it can infer possible action sequences that were not initially programmed but that lead to the desired goals." (Winograd and Flores 1986, 53)

Following the arguments above, we can figure out several key concepts of computational models. First, in order to solve problems of our world with computers, we have to build a model of our model, in order to represent the problems in a domain with certain symbolic system. Some theorists emphasized the "isomorphism" between the model and the world. Some emphasized the model's actual performance instead of the structural similarities between the model and the referent. Once a model is constructed, the model spans a state space of all potential states (solutions). Then problem solving is reduced to searching for the best state(s) within the fixed state space. Very often problem-solving models also include the heuristics for finding good solutions effectively. This setup represents the orthodox of problem solving paradigm. Yet some researches hold a looser view on problem solving. For example, Winograd and Flores (1986) suggested that the computer could achieve goals by certain unprogrammed action sequences based on an "explicit model of the goals and potential actions"[2]. Many computer scientists believed problem solving mimics the way that people deal with problems. The term "Artificial Intelligence" reflects a direct comparison between human intelligence with computer problem solving. Nevertheless, Artificial Intelligence is less often discussed recently. Now fewer people think the computational models should mimic human intelligence directly.

3.1.4 Recent views on computational models

The methodologies and technologies of computing have grown very rapidly in the last two decades. On one hand, the innovations of computer hardware (including networking) and on the social context of using computers (the Internet, Facebook, smart phones and so on) always encourage new computation models and theories. On the other hand, new computation models and theories guided the way computation facilities are developed and the way people use computers and the Internet. The landscape of computation models has shifted. Due to the dramatic growth of internet activities, enormous studies have been made to process "big data"(Howe et al. 2008) or "unstructured data". Pattern recognition, machine learning, and data mining are among the most important topics. Some recent views on computational models are as follows:

> "A model is an abstract representation of a real-world process." (Hand, Mannila and Smyth 2001, 167)
>
> "We can think of the model as an empty table…Model training...The data mining provider uses the algorithm specified during creation of the model to search for patterns in the data. The resulting discovered patterns make up the model content." (Han and Kamber 2006)
>
> "Our aim is to determine, from the data we have available, which model will perform best on data we have not yet seen." (Hand, Mannila and Smyth 2001, 221)
>
> "Our goal is to learn the structure of the model and the parameters…in the model." (Kalogerakis, et al. 2012)
>
> "We require that the probability model must allow… structure induction, where the structure of the model is unknown and must be grown in response to the data." (Zhu, Chen and Yuille 2009)

---

[2] Winograd and Flores 1986, p.53

> "If we already have a model and we want to increase its likelihood, the contrast between where the model puts high probability (represented by samples) and where the training examples are indicates how to change the model." (Bengio 2009)

It is clear that many contemporary researchers do not construct particular models directly; Rather, their models are shaped by real data. They usually commence with certain incomplete model(s) and then train the structure and the parameters of the models from the data set. It is also common that there is more than one model for the same referent. The selection of models does not only depend on how good a model is, but also on pragmatic concerns such as the difficulty of manipulating the model, and the compatibility of the model with other models involved in the task. We can observe an inverse in the notation of modeling, compared to the problem solving paradigm before the 1980s and the contemporary machine learning approaches. The former determines the model first then uses the model to solve the problem in real world; by contrast, the latter collects data from the real world first then uses the data the train the model.

3.1.5 Reflections in science

Modeling in natural science has a much longer history than that in computer science. Due to the continuous impacts of computing during the last century, a bundle of interdisciplines stem from both computer science and traditional scientific fields. Computing also entered social sciences; for example, game theory is closely associated with both economics and computer science. There is no doubt that the methodologies and technologies from computer science have influenced the notations of modeling natural/social science in the last century. First, there are ambiguities in the use of the word "model" in scientific literatures:

> "'Theory' and 'model' are often used interchangeably, or even in combination." (Blaikie 2010, 21)

> "Models as complements of theories… A more extreme case is the use of a model when there are no theories at all available." (Frigg and Hartmann 2012)

> "Scientific theories are models and are frequently mathematical models." (Bender 1978 15)

> “The models of game theory are precise expressions of ideas that can be presented verbally. However, verbal descriptions tend to be long and imprecise; in the interest of conciseness and precision, I frequently use mathematical symbols when describing models.” (Osborne 2009, 2)

> “‘Model’ can refer to a conceptual framework, a hypothesized set of relationships between concepts, a hypothetical explanatory mechanism, or a method for organizing research results.” (Blaikie 2010, 21)

As shown above, “model” can refer to theory, verbal model, or symbolic model like mathematics. Sometimes “model” can also refer to a theory plus certain mathematical model. It implies that the mathematical models in science are often (but not always) associated with theories. Some general notations of modeling in science are shown below:

> “A quantitative description of a natural phenomenon is called a mathematical model of that phenomenon.” (Pinsky and Karlin 2011, 1)

> “A model and its target have to be isomorphic or partially isomorphic to each other.” (Frigg and Hartmann 2012)

> “A mathematical model is an abstract, simplified, mathematical construct related to a part of reality and created for a particular purpose.” (Bender 1978, 2)

> “A model is a representation of a system or process” (Caldas 2001)

> “Scientific modeling has three components: (1) a natural phenomenon under study, (2) a logical system for deducing implications about the phenomenon, and (3) a connection linking the elements of the natural system under study to the logical system used to model it.” (Pinsky and Karlin 2011, 2)

> “No model can ever be complete, correct and universal in its application. We build a model because we want to focus down on certain aspects of a problem.” (Broadbent 1973, 89)

> “A ‘model’ is a conceptual framework, an orderly system of thought, within which one tries to correlate observable data, and even to predict

> data." (Morris 2009)

> "Thales was the first to theorize from the premise that nature can be explained in national terms and that, furthermore, nature can be modeled: It is the achievable task of science to discover a conceptual model which corresponds to nature and accounts for the observed phenomena." (Malin 2001, 37)

The above arguments assert two characters of models: first, the model can effectively reflect the real world; second, the model can only catch a part of the aspects of the referent. However, some researches pointed out that the a model is essentially "fictional," they are just inventions for facilitating our thoughts. Here are some examples:

> "A model, like a novel, may resonate with nature, but it is not a 'real' thing" (Oreskes et al. 1994)

> "These are models which do not perform a representational function and which are not expected to instruct us about anything beyond the model itself." (Frigg and Hartmann 2012)

> "A model is a work of fiction. Some properties ascribed to objects in the model will be genuine properties of the objects modelled, but others will be merely properties of convenience (to bring the objects modelled into the range of the mathematical theory)." (Cartwright 1983, 153)

> "Kauffman's mathematical models of self-organizing processes often do not mention specific biochemical details and there is thus 'the tendency to get further away from real chemistry and to get trapped in the mental world of mathematics'" (Shanks and Joplin 1999)

> "I take the positivist viewpoint that a physical theory is just a mathematic model and that it is meaningless to ask whether it corresponds to reality" (Hawking and Penrose 1996, 4)

> "The simulated climate is significantly different from that which is observed because no model is perfect…Are there more satisfying ways to prove the 'correctness' of a model? Oreskes et al. (1994) argue that a positive answer can be given only if the model describes a closed sub-system of the full system…" (von Storch and Zwiers

> 2004, 129)

> “It is a ‘model’, something that does not really happen in nature, but which helps us to understand things that do happen in nature.” (Dawkins 1976, 74)

According to the various views on modeling, we can summarize several characteristics of models: 1) Some models reflect the real world by their correlations with empirical data. 2) Some models don’t correspond to the real world because they are constructed as abstract tools for thinking and learning. 3) All models are incomplete, for they can only catch a part of the aspects of the problem. 4) There is no way to prove a model is “true”, and there are usually multiple models for the same target.

3.2 Modeling in Architecture with Computers

3.2.1 Modeling and computing for architecture

The term “model” is related with several distinct concepts in architecture. Very often, “model” means scale model made of tangible materials like cardboard, wood, plastics, or digital models in CAD software. As Schlüter (2010) puts it, “In architectural design, the notion of modeling is a different one. Models are used to represent a building design. The range of models spans from abstract geometrical cardboard models to 3D-computer models such as used for photorealistic imagery.”[3] Yet, this section focuses on the abstract models for architectural design. In this context, the definition of a model is very similar to that in computer science and scientific fields:

> “A model is an imitation or approximate representation of a system or of complex functions. It is a simplified or abstract view of the complex reality using a physical, mathematical, or logical representation of the system of entities, phenomena, or processes.” Alfaris (2009, 117)

> “A model is an abstract description of the real world giving an approximate representation of more complex functions of physical systems…A mathematical model is a model that represents a system by mathematical relations.” (Papalambros and Wilde 2000, 1)

Alfaris (Alfaris 2009, 118) clearly distinguished several kinds of models in archi-

---

[3] Schlüter 2010, p.47

tecture: the physical model, the symbolic model, the mathematic model, and the computation model:

> "Models usually fall into one of two categories, physical or symbolic model."
>
> "A mathematical model is a formal model that comprises symbols, assumptions about the symbols, the relations among the symbols, and connections between the actual model and these symbols and relations."
>
> "A mathematical model becomes a computational model as soon as its associated equations are coded into a computer program where it can be studied numerically and graphically."

Over the last five decades, enormous amounts of computational models have been developed for architectural design. The early studies strove to establish one general model for architecture, or at least make one general theory of modeling in architecture. However, it turns out that various models and theories have always been growing. Now, few people would imagine that one single model or theory could fit all kinds of scenarios in architectural design. The following subsections present and discuss the various views on models, in a sequence of "models of architecture", "models of design process", "modeling, knowledge and expertise", "parametric/generative models of architecture" and "CAD systems and Building Information Modeling".

3.2.2 Models of architecture

In order to deal with architecture in computers, the theorists emphasized the representations of the architecture, or the potential solutions, in certain formal systems like mathematics:

> "We can symbolically model a system which is of interest to us by letting specific design variables represent specific properties of the system." (Mitchell 1977, 40)
>
> "The model considers the building as an integrated system composed of various subsystems and the interactions between them, and allows the consequences of changes in the various subsystems to be determined for the building as a whole." (Brotchie and Linzey 1971)

> "An abstract description of the artifact using mathematical expressions of relevant natural laws, experience, and geometry is the mathematical model of the artifact. This mathematical model may contain many alternative designs, and so criteria for comparing these alternatives must be introduced in the model." (Papalambros and Wilde 2000, 1)

The models of architecture can take various forms. March and Steadman introduced a wide range of methods in the book "The Geometry of Environment" (1971). One of the simplest models is the binary string. For example, the ArchiKluge (Miranda 2005) program uses a 64-bit binary string to represent spatial configurations of 4×4×4 grid (Fig 3.4). The value (0 or 1) in a particular position of the grid denotes the on/off state of the corresponding cell in that position.

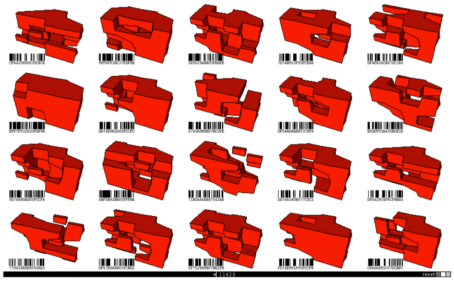


Fig 3.4: Using binary string to represent spatial configurations of 4×4×4 grid, ArchiKluge, Miranda (2005).

Another example is Rosenman's (1996) symbol sequences. Each sequence represents the shape of a room. For instance (W1, N1, E1, S1) makes a square. More examples are shown in Fig 3.12. Thereby the whole house is represented by a group of sequences. The sequences are organized in a tree hierarchy: house-zone-room. Many programs employ regular lattice (2D or 3D) to represent architecture. Doulgerakis's (2007) splitting scheme is one of the few exceptions. The floor plan is produced by a series of the functions of "polygons"(splitting one rectangle into two rectangles), "vertical" (setting the direction of splitting to vertical) and "horizontal" (Fig 3.5). This model can produce non-orthogonal walls (Fig 3.15).

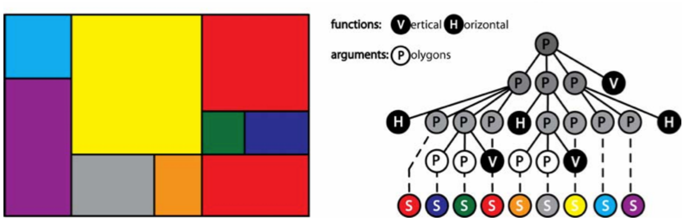


Fig 3.5: A floor plan produced by a series of functions (Doulgerakis 2007).

Sometimes the models of architecture are constructed to facilitate formal reasoning and analysis. One important model of architecture is the graph model of the connectivity of rooms. Hillier (1984) argued that the structure of interior space reflects the social environment. For instance, he showed two graphs (Fig 3.6); the left one has many edges and looks 'shallow' when compared to the 'deep' one in the right. The mean RA (relative asymmetry) value of the one on the left is 0.202 and that of the right is 0.464. The "depth" (measured as RA values) of the graphs was regarded as "a function of the form of social solidarity."[4] Besides the graph model for analyzing spatial connectivity, various models are closed related with the measurements of the "quality" of the buildings, such as gain of sunlight, ventilation, land use, costs of civil engineering.

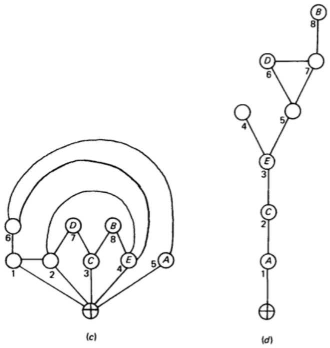


Fig 3.6: Two graphs of interior spaces, Hillier (1984).

3.2.3 Models of the design process

Although the idea of the design process can be traced to the 19th century, the computation models of the design process were originated in the 1960s. In computer science, Simon (1996) described a "design science" in his book "The Sciences of the Artificial". Then architects also wanted to make a general model for solving

---

[4] Hillier 1984, p.163

design problems. One of the most celebrated contributions is Christopher Alexander's "Notes on the synthesis of form" (1964). Alexander developed a constructive diagram that "offers us a way of probing the context, and a way of searching for form…a bridge between requirements and form." [5] He presented a causal graph for eliminating misfit:

> "Its variables are the conditions which must be met by good fit between form and context. Its interactions are the casual linkages which connect the variables to one another. If there is not enough light in a house, for instance, and more windows are added to correct this failure... If we represent this system by drawing a point for each misfit variable, and a link between two points for each casual linkage (Fig 3.7)" (Alexander 1964, 42)

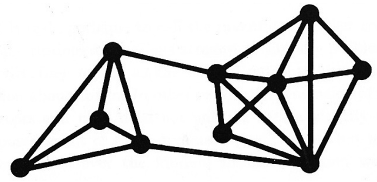


Fig 3.7: A graph of casual linkage (Alexander 1964, 43).

Other researchers of the 1960s clearly defined a problem-solving method for architectural design. Based on the studies of the 1960s and the early 1970s, Mitchell (1977) summarized the problem solving model: first making a representation of the potential solutions so that a state space (or solution space) can be constructed[6], then searching the best one(s) in the state space according to certain defined objectives. Problem solving or optimization became the orthodox of computational models in architectural design, and subsequently stood as the base of Computer-aided Architectural Design in the early years. Many studies regarded architectural design is akin to problem solving or optimization:

> "We have chosen to consider architectural design as a goal-directed search process which relies on prior experiences and knowledge. The purpose of the process is to define an object that achieves some desired behavioral and spatial characteristics…" (Carrara, Kalay and Novembri 1994)
>
> "Architecture is involved with the design of suitable and satisfactory

---

[5] Alexander 1964, p.92

[6] Each point in the state space corresponds to one solution, or one state of the target system.

> environments...The architect is involved in an optimization process" (Gero 1975)
>
> "Design is process of problem solving." (Leeuwen 1999, 46)

Following the pace of Artificial Intelligence in computer science, the forerunners in the CAAD field strove to model the design process in a single framework. One idea is that the model of design should mimic the way that human designers make designs. For example, Gero (1990) suggested some simple models:

$$F \rightarrow D$$
$$S \rightarrow D$$
$$F \rightarrow S$$

→: transformation
F: function
D: design description
S: structure

Finally, Gero proposed a general model of design involving function, structure, behavior, and design description (Fig 3.8).

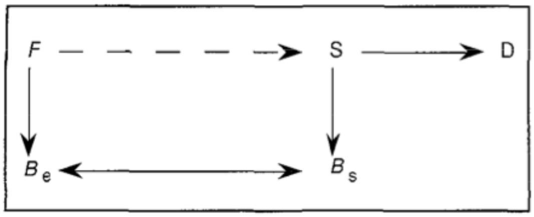
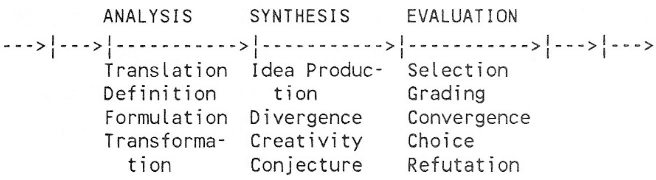


Figure 2. Model of Design as a Process.

B_e = Set of expected behaviors
B_S = Set of actual behaviors D = Design description
F = Set of functions S = Structure
→ = Transformation
---> = Occasional transformation
<–> = Comparison

Fig 3.8: Model of design as a process (Gero 1990).

From a slightly different perspective, Grant (1993) presented a linear model of design that consists of "the three crucial steps being ANALYSIS, SYNTHESIS and EVALUATION". See Fig 3.9.



```
         ANALYSIS     SYNTHESIS     EVALUATION
--->|--->|----------->|----------->|----------->|--->|--->
         Translation  Idea Produc-  Selection
         Definition     tion        Grading
         Formulation  Divergence    Convergence
         Transforma-  Creativity    Choice
           tion       Conjecture    Refutation
```

Fig 3.9: A linear model of designing (Grant 1993).

Problem solving is a typical model of the design process, since it does not only model the potential solutions but also models the search process for good solutions. There were many arguments on the models of design process, many of them lead to problem-solving or optimization:

> "It assumes that we can construct some kind of a representation of the system that interests us, and that problem-solving can be characterized as a process of searching through alternative states of the representation in order to discover a state that meets certain specified criteria." (Mitchell 1977, 27)

> "The model we have developed for the purposes of representation the design process is based on a metaphorical concept which views buildings as living organisms…The objective of any design process is to define a building organism that will achieve certain functional and aesthetic needs." (Carrara, Kalay and Novembri 1994, 153)

> "A model of design must be able to select a schema, discover constraints in terms of that schema, and progressively modify or vary the schema and the constraint network until an acceptable or satisficing design is reached" (Heath 1993)

> "A design model that includes an evaluation criterion is a decision-making model. More often this is called an optimization model, where the 'best' design selected is called the optimal design and the criterion used is called the objective of the model." (Papalambros and Wilde 2000, p.9)

> "The aspects of computation considered here are representation and process." (Knight and Stiny 2001)

These arguments hold that the models of design should support a search process towards "good" solutions. It can be called search, problem solving, decision-making, or optimization. At the beginning, many researchers used linear/nonlinear programming as a problem solving technique, e.g., Aguilar (1968), Brotchie and Linzey (1971). Later, a generalized framework of optimization was summarized: the solution space is created once the representation of building is decided, so the task of optimization is just to find the best instances within the solution space. Since exhaustive search in solution space is normally not feasible, heuristics are essential for accelerating the search procedure.

In terms of architecture, particular heuristics are often promising. Whitehead and Eldars (1965) experimented with a novel method: adding new rooms to the layout one by one. Fig 3.10 illustrates all the possible locations of the new cell (no.13) in the existing layout. The costs corresponding to all possible locations are calculated so that the best location(s) can be selected. Fig 3.11 shows one of the generated floor plan layouts.

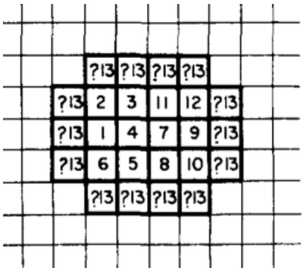


Fig 3.10: All possible locations of the new cell (no.13) (Whitehead and Eldars 1965)

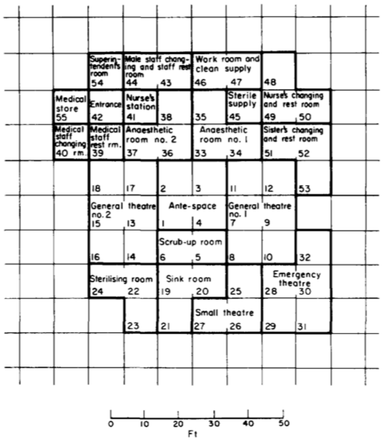


Fig 3.11: One of the generated floor plan layouts (Whitehead and Eldars 1965)

Some evolutionary designs[7] also fall into the category of optimization. Menges (2012) spoke about the evolutionary algorithm:

> "Computationally a problem can be described in terms of a search space, which is the theoretical space of all existing parameters within a defined problem. Potential solutions are generated as populations of individuals over many generations by iterative, stochastic sampling of parameters. The evolutionary algorithm navigates the search space in order to trace the best amongst these many solutions."

The terminology "genotype" in the evolutionary algorithm refers to the symbolic representation of potential solutions. The "phenotype" refers to the structures or the behaviors of the target system that the "genotype" represents. A phenotype can be regarded as a function of a genotype. The fitness function (equivalent to objective/cost function) is defined as a function of a phenotype, thus as a function of genotype. The evolutionary algorithm manipulates a pool of genotypes by reproduction, selection, crossover and mutation operations, in order to discover the best one(s) with the highest fitness value. Impressive applications of evolutionary design are given by (Rosenman and Gero 1996), (Chouchoulas 2003), (Doulgerakis 2007) and others.

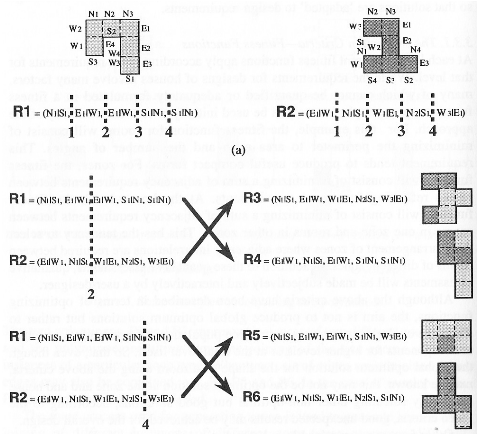


Fig 3.12: Crossover the genotypes (Rosenman and Gero 1996).

---

[7] Here evolutionary design refers to those optimization approaches that employ evolutionary algorithms.

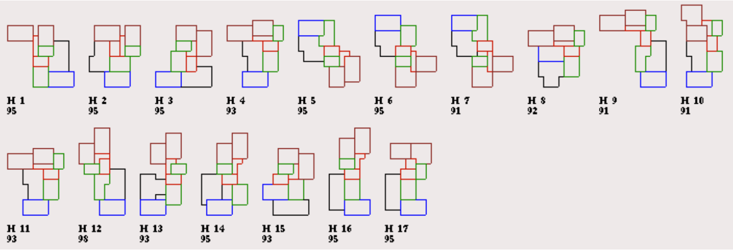


Fig 3.13: Floor plans generated by Rosenman's program (Rosenman and Gero 1996).

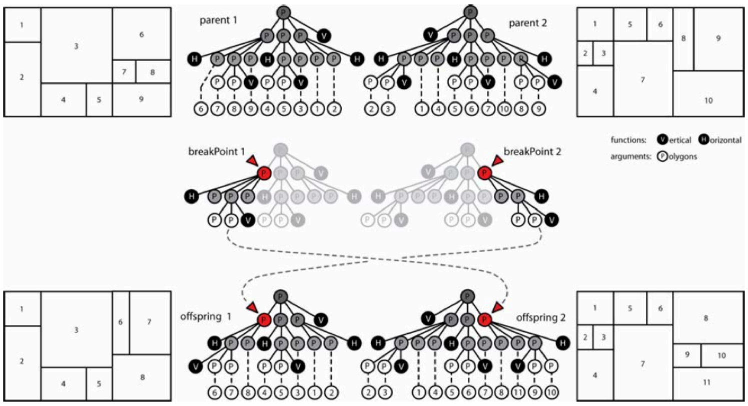


Fig 3.14: Crossover the genotypes in genetic programming (Doulgerakis 2007).

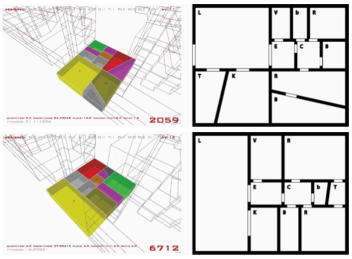


Figure51 Typical layouts [Activities Assignment Embryology]

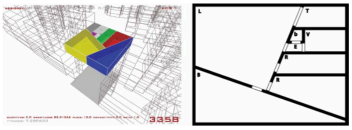


Figure52 Typical layout [Randomly Assigned Activities]

Fig 3.15: Three layouts produced by the Doulgerakis's program (Doulgerakis 2007).

In fact, the optimization/problem solving paradigm has been widely attacked. For example, Chouchoulas (2003, 6) stated: "optimisation does not solve design problems…Optimisation only modifies designs by making small quantitative changes and cannot arrive at a better solution by making a qualitative jump." First, some theorists like Rittel (1973) pointed out that architectural design problems are not well-defined problems at all. Therefore, problem solving is not valid for architectural design. The orthodox problem solving was also criticized inside the CAAD community. For instance Frazer (1995, 14) stated: "It is notoriously difficult to describe architecture in these terms, except in the very limited sense of an architectural brief to which there are endless potential solutions. The other problem is that any serious system will generate an almost unmanageable quantity of permutations." In terms of creative design, Gero (1992) argued that problem solving constructs a fixed solution space (or search space) from the outset. Thus, it would not lead to creative results without exploring beyond the fixed solution space.

The models of the design process are difficult to implement in practice. One problem is that the design problems change a lot during the design process. As Leeuwen, Hendricx, and Fridqvist (2001) addressed, "The conceptual information model, containing the classes of design objects, changes over time as design proceeds." In design practice, one model can only catch a very limited aspect of the design problem, and the scope of the design problem varies a lot with projects and with architects. Even the "multipurpose optimization" approaches can only take account of a few aspects of design. As Caldas (2001, 17) stated:

> "Most early applications of mathematical models and their respective computational implementations to architecture problems were thus confined to limited problem settings, where only a subset of the large complexity of issues interacting in an architecture problem were dealt with. This related to proportion issues, floor plan layout studies, research on urban densities, solar shading studies, etc."

3.2.4 Modeling, knowledge and expertise

Enormous effects have been made in the field of Artificial Intelligence to encode the domain knowledge into formal models. Knowledge representation, ontology, knowledge-based systems, expert systems and case-based reasoning are among the relevant topics. For example, Leeuwen (1999, 21) addressed:

> "Knowledge-based systems, also called expert systems, are decision support systems that utilise formalized definitions of expert knowledge on a specific domain…The knowledge can be represented in the system using various technologies, such as rules, constraints, or case descriptions"

People also suggested modeling the domain knowledge in architecture. For example, Alexander's (1977) Pattern Language is regarded as an approach to reusing domain knowledge and previous experiences. Explicitly, Logan and Smithers (1993) proposed a model of design based on domain knowledge (Fig 3.16). The approaches to knowledge representation had gained great attention in theoretical research, however, they were seldom implemented and had little impact on practice.

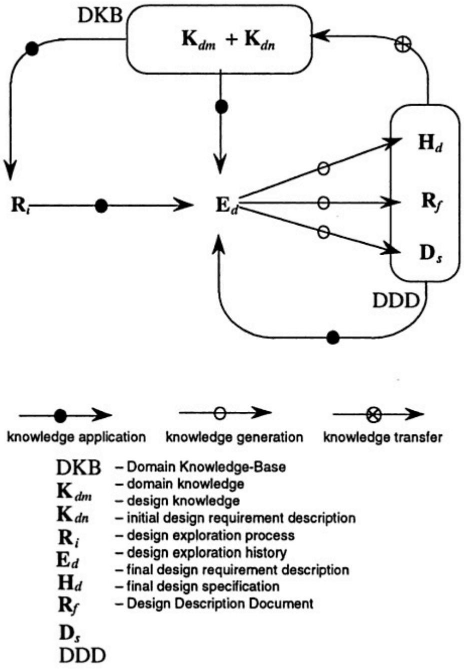


Fig 3.16: An exploration-based model of design (Logan and Smithers 1993).

Some general views on knowledge modeling in architecture and design are as follows:

> “Explicit representation of design knowledge is needed if scientific methods are to be applied in design research, and if computers are to be used in aid of design education and practice.” (Carrara, Kalay and Novembri 1992)
>
> “The modeling of a design knowledge base may thus be consider the formulation and structuring of all knowledge required for dealing with a particular design task and the provision for memory organization and indexing for retrieval in the task” (Oxman and Oxaman 1994)
>
> “Knowledge formulation is basically a model construction process, therefore knowledge exists as long as there is a model of it” (Motta 1999, 12)
>
> “Therefore this knowledge acquisition process is no longer seen as a transfer of knowledge into an appropriate computer representation, but as a model construction process” (Studer, Benjamins and Fensel 1998)
>
> “We need a representation, that is data structures holding the knowledge of our design problem…In our model the knowledge relevant for a situation is represented as a network… It is a network of entities (objects) and relationships, where one can build a coherent model of the world.” (Takala, 1993)
>
> “A model of integrated architectural and design knowledge comprised of four levels…Syntactic and formal elements and operations…Syntactic structures and compositional operations…Generic knowledge structures…Design paradigms and schemata” (Oxman and Oxman 1990)

Cased-based reasoning (CBR) and case-based design (CBD) are closely associated with knowledge modeling. The cased-based design is supposed to benefit from previous cases without an explicit model of the domain knowledge. If we store a great number of cases in a proper way, it can help us to study the successful cases and to reuse them when encountering similar situations. A typical case-based model includes case indexing, case retrieval, case reuse/revise and case adaptation. One example is de Silva Garza and Maher’s (2001) model of cased-based design (Fig 3.17).

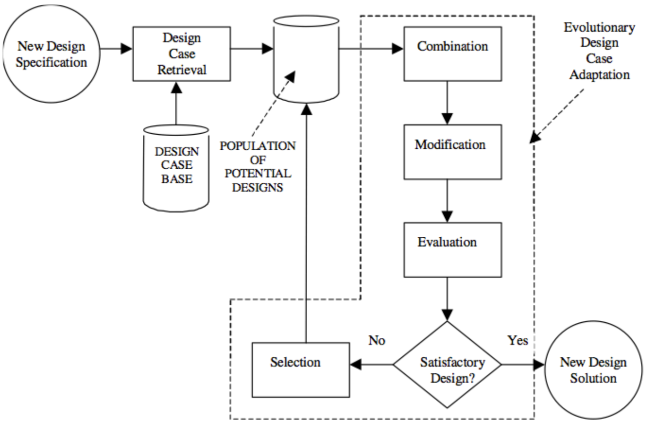


Fig 3.17: A process model for case-based design with evolutionary case adaptation (de Silva Garza and Maher 2001).

Case-based design became an important topic of the CAAD field in the 1990s. Here are some general ideas on case-based reasoning/design:

> "CBR is useful tool for intelligent system development in a domain where either an explicit model does not exist or one is not yet adequately understood (Kolodner 1993). Design is such a domain." (Watson and Perera, 1997)

> "Case-based reasoning is a general paradigm for reasoning from experience. It assumes a memory model for representing, indexing, and organizing past cases and a process model for retrieving and modifying old cases and assimilating new ones. Case-based reasoning provides a scientific cognitive model." (Slade 1991)

> "(CBR) as a problem-solving approach of a reasoned which inferences from previous solutions which are adapted to current situations." (Oxman and Oxman 1994)

Kolodner (1992) formulated CBR as a cycle system of retrieve, propose, adapt, justify, criticize/evaluate, and store (Fig 3.18). Adapting cases to new situations is a big challenge to case-based design. There have been many applications of case-based design in architecture. Some recent overviews on CBD can be found in (Heylighen and Neuckermans 2001) and (Richter, Heylighen and Donath 2007). Weber et al. (2010) made a comparison of CBD applications in architectural design (Fig 4.19). Generally speaking, these applications fall into two categories: retrieval-oriented and adaptation-oriented. Archie-II system (Domeshek and Kolodner 1993) is a typical retrieval-based system (Fig 3.20).

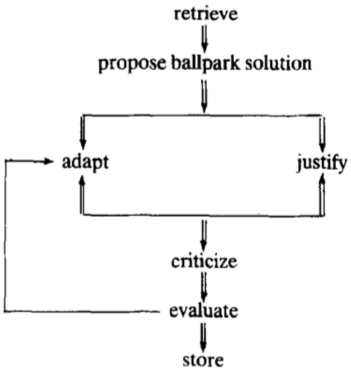
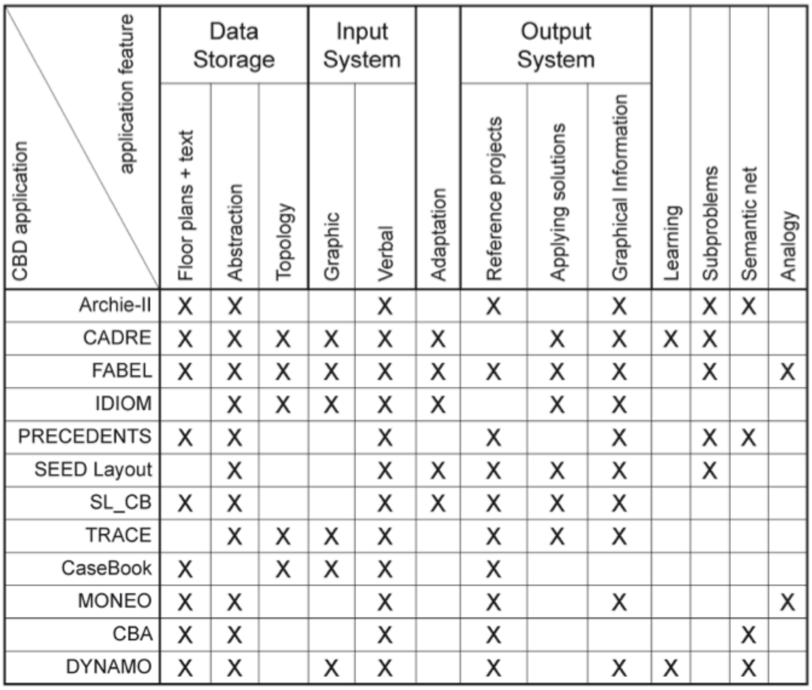
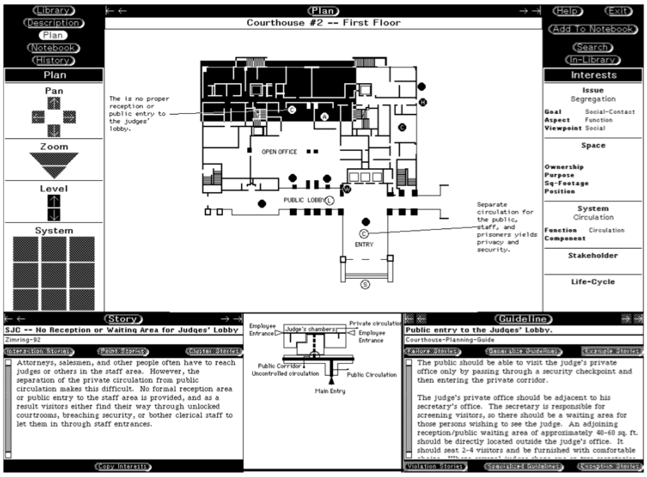


Fig 3.18: A cycle of CBR (Kolodner 1992).



application feature / CBD application	Data Storage			Input System			Output System						
	Floor plans + text	Abstraction	Topology	Graphic	Verbal	Adaptation	Reference projects	Applying solutions	Graphical Information	Learning	Subproblems	Semantic net	Analogy
Archie-II	X	X			X		X		X		X	X	
CADRE	X	X	X	X	X	X		X	X	X	X		
FABEL	X	X	X	X	X	X	X	X	X		X		X
IDIOM		X	X	X	X	X		X	X				
PRECEDENTS	X	X			X		X		X		X	X	
SEED Layout		X			X	X	X	X	X		X		
SL_CB	X	X			X	X	X	X	X				
TRACE		X	X	X	X		X	X	X				
CaseBook	X		X	X	X		X						
MONEO	X	X			X		X		X				X
CBA	X	X			X		X					X	
DYNAMO	X	X		X	X		X		X	X		X	

Fig 3.19: Comparison of CBD applications in architectural design (Weber et al. 2010).



The is no proper reception or public entry to the judges' lobby.

Separate circulation for the public, staff, and prisoners yields privacy and security.

SJC -- No Reception or Waiting Area for Judges' Lobby

Zimring-92

Attorneys, salesmen, and other people often have to reach judges or others in the staff area. However, the separation of the private circulation from public circulation makes this difficult. No formal reception area or public entry to the staff area is provided, and as a result visitors either find their way through unlocked courtrooms, breaching security, or bother clerical staff to let them in through staff entrances.

Public entry to the Judges' Lobby.

Courthouse-Planning-Guide

The public should be able to visit the judge's private office only by passing through a security checkpoint and then entering the private corridor.

The judge's private office should be adjacent to his secretary's office. The secretary is responsible for screening visitors, so there should be a waiting area for those persons wishing to see the judge. An adjoining reception/public waiting area of approximately 40-60 sq. ft. should be directly located outside the judge's office. It should seat 2-4 visitors and be furnished with comfortable

Fig 3.20: The user interface of the Archie-II system (Domeshek and Kolodner 1993).

De Silva Garza and Maher (2000) developed an adaption-oriented CBD. The system focuses on the sketch-level layouts with a three-level hierarchy: landscape level, house level and room level. A 3×3 grid is applied to each level (Fig 3.22). Based on this uniform coding scheme, case retrieval and case adaptation can proceed effectively. A great diversity of cases will be selected from the case library if they meet one of the design objectives. Such an initial population is later used by an evolutionary algorithm to reproduce new solutions. Various goals can be achieved by combining "good" parts from different cases. The experiment employs the prairie houses of Frank Lloyd Wright as the case library. Fig 3.22 shows one of the results.

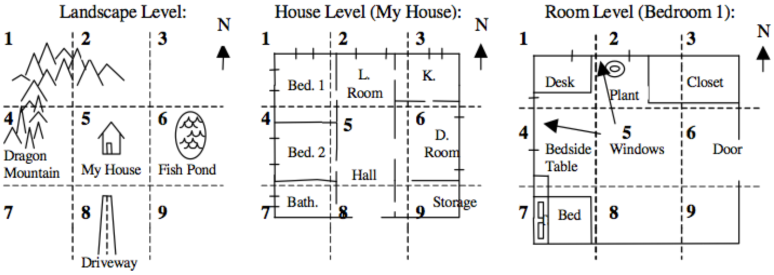


Fig 3.21: The method of encoding cases: the 3×3 Grid applied to the landscape level, house level and room level. (de Silva Garza and Maher, 2000).

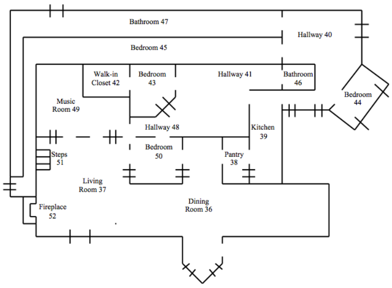


Fig 3.22: One solution produced by case adaptation/combination.
(de Silva Garza and Maher, 2000).

Usually domain knowledge is required for making sense of original cases[8] and for avoiding invalid operations during revision/adaption. Ironically, the requirement of rich domain knowledge is against the original motivation of case-based design: benefiting from the implicit information in cases. As Hua and Faltings (1993) put it:

> "Case-based reasoning has been credited for its advantages in solving design problems. However, we have shown that case-based design poses fundamental problems, some of which may make it impossible to benefit from the promises of the paradigm. For example, there does not seem to be a way to guarantee the correctness of adaptations without using a complete domain model"

In general, there is a gap between the limited capability of a knowledge-based system and the great diversity of design situations. As Oxman and Oxman (1994) pointed out:

> "With few exceptions, models of expert knowledge appeared to have limited utility for the range and complexity of design tasks…New approaches to Knowledge-Based Systems were required which could accommodate the cognitive complexity of design thinking."

3.2.5 Parametric/Generative models

One of the most successful computational approaches to architecture is parametric modeling. By fixing the topological relationship between the elements, the parametric model can make various articulations by responding to different sets of parameters. Woodbruy (2010) gave a comprehensive example of a parametric model (Fig 3.23). A data flow (Figure 3.21 left) illustrates the dependences between the

---

[8] The cases are often in the form of unstructured data.

parameters. For instance, w_t (sum of widths) and w_0 (width of room$_0$) are independent; together they determine the value of w_1 (width of room$_1$) and w_2 (width of room$_2$).

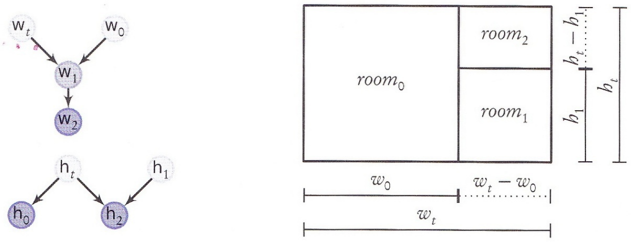


Fig 3.23: A parametric model of three rooms (Woodbury 2010, 25).

Nowadays there is software supporting parametric modeling. For example, the Grasshopper works as a plug-in for Rhinoceros. Fig 3.24 shows a simple example: a triangle (left) is constructed by connecting the "Ln"(line) components to the point components (right). The software facilitates the modeling process with graphical representations; however, it does not change the nature of the parametric modeling.

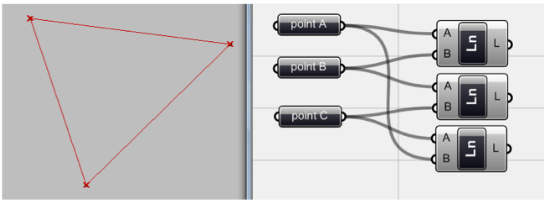


Fig 3.24: Parametric modeling with Grasshopper (Khabazi 2010, 12).

Today many architects are familiar with the concept of parametric modeling:

> "Conceiving, arranging and editing dependencies is the key parametric task…Designers use dependencies in combination to exhibit some desired aggregate form or behaviour." (Woodbury 2010, 25-26)

> "Parametric modelling has been understood as instrumental for its ability in improving workflow, its rapid adaptability to changing input and its delivery of precise geometric data for digital fabrication and performance analysis." (Menges 2006)

> "Digital models, which aim at describing and simulating aspects of real objects, need to be set up carefully, with the right focus and the appropriate level of abstraction to deliver meaningful results. Especially when using parametric models, the hierarchic dependencies within complex structures have to be thoroughly untangled and precisely described in formal algorithmic and mathematic notations." (Scheurer and Stehling 2011)

> "Potentially performance-evaluation can inform parametric model and modify the geometrical model, leading to performance-based generative processes." (Oxman 2008)

> "The goal in formulating a grammar is to fully define all designs of interest by modeling knowledge about how designs are generated from combinations of topology and geometry transformations." (Shea and Cagan 1999)

Shaper grammar, or more generally the linguistic approach to design, is closely related to parameter modeling. Stiny (1972) introduced the shape grammar to generate painting and sculpture forms. Later, the model was employed to formulate certain building corpuses (Stiny and Mitchell 1978). Flemming (1990) interpreted shape grammar formulism as "a collection of rules that embody the compositional principles or conventions that underlie a certain piece of architecture." Mitchell's "Logic of Architecture" (1990) established an architectural theory based on the shape grammar formulism. Recently, shape grammar has also adapted for procedural modeling (Parish and Müller 2001).

Under many circumstances, parametric models in architecture are deterministic models that maps a set of parameters to geometries while generative models prefer probabilistic models such as cellular automata, agent-based systems, self-organization, or complex adaptive systems. Generative models often take account of the design process besides design representations. Recently, Menges (2012) summarized the concept "computation design" compared to traditional CAD models.

> "The transition from computer-aided to truly computational design entails a shift from (i) modelling objects to modelling processes, (ii) from designing shape to designing behaviour, (iii) from defining static digital constructs to defining computing systems capable of reciprocal data exchange and feedback information."

Based on computational models, Menges (2012) also explained morphogenesis or form-generation in architectural design:

> "The proposed morphogenetic form generation can derive multiple valid equilibrium states between system-intrinsic constraints and system-external influences ... Similarly to form finding, it also requires an understanding of material systems not as derivatives of standardized building systems and elements facilitating the construction of pre-established design schemes but rather as generative drivers in the design process."

Leach (2009) also addresses the concept of digital morphogenesis based on generative models:

> "It (morphogenesis) has been appropriated within architectural circles to designate an approach to design that seeks to challenge the hegemony of top-down processes of form-making, and replace it with a bottom-up logic of form-finding. The emphasis is therefore on material performance over appearance and on processes over representation."

Menges (2012) presented a case of "urban block morphologies." The units in the block are constructed by a series of subdivisions (of the overall volume), which are encoded by abstract "gene sequences." One gene corresponds to one spatial organization of the block. To evaluate any genes, a fitness function with five climatic criteria and two spatial criteria is defined. Then the evolutionary algorithm can manipulate a pool of genes to search for genes with high performance. Fig 3.25 shows three iterations of the evolutionary process.

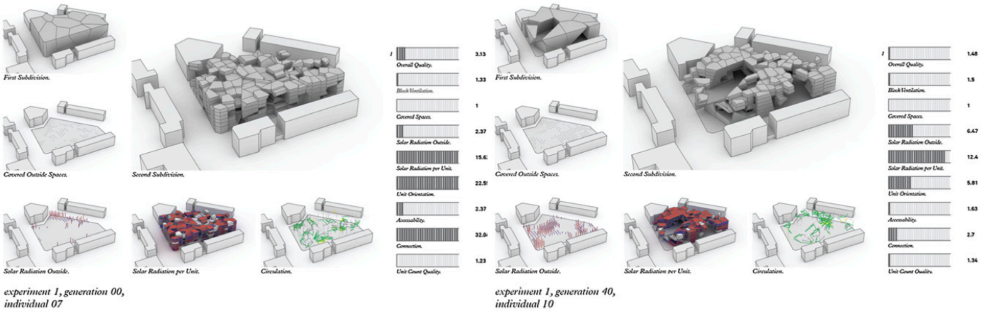


Fig 3.25: Morphologies in the evolutionary process (Menges 2012).

Architects use parametric/generative models in different ways from scientists and engineers. Strongly reflecting architect's self-consciousness in using computers, Frazer (1995) proposed "an Evolutionary Architecture":

> "An Evolutionary Architecture investigates fundamental form-generating processes in architecture…It proposes the model of nature as the generating force for architectural form." (Frazer 1995, 9)

A number of topics are closely associated with generative models; for example, Morphogenesis (Kolarevic 2003a; Leach 2009, Menges 2012), emergent design (Hensel Menges and Weinstock 2004), form-finding (Leach 2009), and Performance-based design (Oxman 2008; Kolarevic and Malkawi 2005). Among these approaches, there is a great diversity in the nature of models and in the methodology of using models. Neither one model nor one methodology is adequate to address all kinds of design situations.

3.2.6 CAD systems and Building Information Modeling

Building computer-aided drawing system for architects was already successful in the 1980s. Nowadays most architects are familiar with some CAD software. Some software only take account of the geometric aspects of buildings. Thus, there have been numerous attempts to create building models with rich semantic information. One contemporary development in this direction is Building Information Modeling (BIM). Some speculations on general CAD systems are as follows:

> "The Generic Building Model is presented in two formats: graphic and textual." (Eastman and Siabiris 1995)

> "CAD employs the computer as a helpful extension of established design processes based on geometric information that represents the designed object or architecture as a metric construct of points, lines, surfaces and solids if represented in three dimensions." (Menges 2012)

> "Data structures in data models are entities, relationships, attributes etc…An object (or entity) is a set of closely interrelated data about something in the modelling domain. 'Something' can be a physical object but it could also be an equation system, or any kind of abstract object."(Björk 1992)

> "A model or abstraction of an object is a representation of that object resulting from a particular view taken. Since there are many different views of a building there will be many corresponding models. (Fig 3.26)" (Rosenman and Gero 1996)

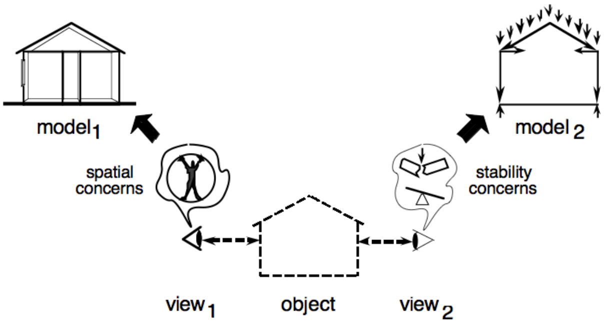


Fig 3.26: Multiple Views and Models (Rosenman and Gero 1996).

"The result is a set of models where each model has its own concepts and elements. Some elements may be common to more than one model although some of their properties may differ and some elements in one model are related to elements in other models. (Fig 3.27)" (Rosenman and Gero 1996)

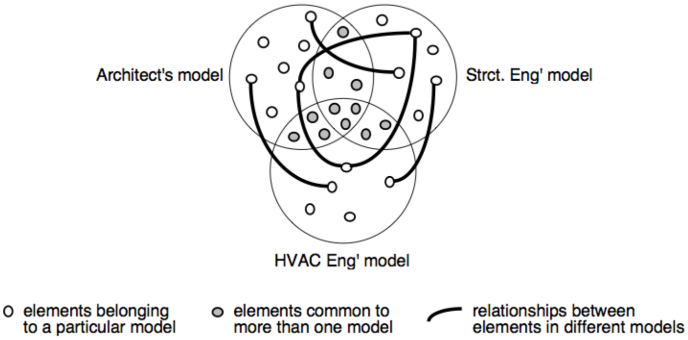


Fig 3.27: HVAC Eng' model (Rosenman and Gero 1996).

"Therefore the underlying representation of the design elements in the shared workspace must be shared…While the shared visual representation provides the basis for visualising design elements, the shared underlying representation provides a persistent memory of design information, ideas, and intents." (Saad and Maher 1993)

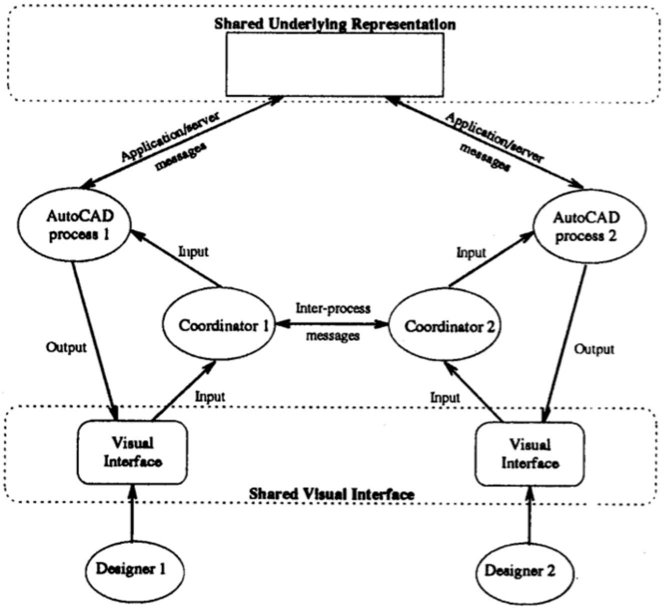


Figure 4: A synchronous multi-user CAD system

Fig 3.28: Shared Underlying Representation in CAD system (Saad and Maher 1993).

> “Some attempt the explicit representation of all domain’s concerns in one integrated model, while others propose the used of multiple, domain specific models” (Haymaker, Fischer and Kunz 2002)

As shown above, a critical problem in CAD systems is that one object can be interpreted in many different ways. In other words, there are potentially multiple models for the same target. This is a natural consequence of the very nature of the model: models only represent a part of the aspects of the referent.

Compared to traditional CAD systems, Building Information Modeling (BIM) catches richer information of architecture, taking account of architecture, engineering, and construction. It is supposed to allow the designers from various disciplines to work on the same model. Moreover, BIM also tries to build a continuous dataflow through all stages: early state planning, design, civil engineering, construction, and maintenance. Since there are different BIM products like Autodesk’s REVIT, Bentley’s MicroStation, and Graphisoft’s ArchiCAD, the Industry Foundation Classes (IFC) aimed at an open standard for various BIM platforms. IFC defines a hierarchical structure that is compatible with various BIM platforms. The concept of IFC is similar to the frame based design (Mark 2008):

> "Design proceeds by retrieving and specifying design "frames" (Minsky 1986), each of which consists of a collection of "slots" that represent attributes. The frame (like a prototype, object instance, or even a case) represents knowledge about a typical design".

In order to fit as many situations in architecture as possible, the IFC standard of BIM modeling is supposed to be very general and flexible. However, Cerovsek (2011) stated: "'Industry Foundation Classes' (IFC), ISO/PAS 16739, do not yet exhibit the three roles of any standard enabling innovation: (1) inter-operability, (2) trust, and (3) comparability."

3.2.7 Discussion

A great number of computation models have been developed for architecture since the 1960s. According to subsections 3.3.2 - 3.3.5, the models can be categorized in different ways. First, some models are akin to theories or verbal models; others are defined with certain formal systems like mathematics. Second, there are models of architecture and models of the design process, though sometime the line between the two is vague. The former are formal specifications of buildings, including geometric, topological, and sematic information. The latter usually include the former as a part, but emphasize the formulation of the process of producing desired solutions. Third, some models are bound to certain theories, while others are only used to solve particular problems. Fourth, some models are abstract without any direct reference to the real world; however, other models are subject to particular design problems or particular architecture systems.

Usually theories or mythologies are closely related with the relevant models. For example, linear/quadratic programming models (Aguilar 1968; Brotchie 1971) are under the umbrella of the theory of problem solving; a fluid simulation model (Verebes 2009) for form finding is under the theory of parametric/generative design. However, there are exceptions. In the last decade, computational models gradually entered into every corner of design, parallel with the widespread of modeling software and scripting tools. As a result, many architects have become "conscious " when (re)using models. Their purpose of using a model is often independent on the original context of the model and independent on the motivation of the original modeler.

Sometimes a model is constructed without any theory of it. Sometimes a model is constructed without any obvious referent. Besides, the referent of the model might be contingent in specific design tasks. For example, if a model of Voronoi tessellation is employed to construct a set of rooms, then the referent of the model is the rooms; yet, if the Voronoi model is used to create a pattern on the facade, the referent of the model becomes the facade elements. In short, the model is not to necessarily be bound to a fixed referent, nor necessarily to follow its original theory (if there is any). Rather the model could follow the specific purpose of the design task. Thus, the interface and the inner structure of the model might need to be modified, and subsequent the model will get a new role in the new context. It is consistent with (Mark, Gross, and Goldschmidt2008) if we consider computational models as tools for design: "the designer is the tool builder. In this respect the tool is not selected, but created contingent on the type of design task, the stage it is in, and adapted to circumstances."

Chapter 4

Combined Modeling

Although people have benefited a lot from models, there are dilemmas in modeling that are inherently associated with the nature of modeling. It is annoying that: there are multiple models for the same original because the models are essentially partial[1]. Moreover, the multiple models are usually not consistent with each other. The approaches to metamodeling strove to solve this problem by making abstractions of multiple models, i.e., making the model of models. However, there must be many different metamodels, because the metamodels themselves are models that are essentially partial. As a result, the problem of multiple models cannot be solved by this notion of metamodeling. Yet this chapter argues for an alternative approach—combining models. It creates connections between the models in order to make them work together. Combined modeling is not a final solution to the problems of modeling; however, it is not worse than modeling and making metamodel of models. In particular, combining models can be fruitful in architecture, for it is good for arranging the complex network of design issues. The architect can convey the design concept by constructing novel relationships between the models.

4.1 Dilemmas of Modeling

Modeling quests for correctness and effectiveness, however, both correctness and effectiveness lead to dilemmas. The dilemmas come from the very nature of models: partial representation of the original. Smith (1985) fabricated an example when discussing the "limits of correctness" of models: "Suppose the people want peace, and the President thinks that means having a strong defense, and the Defense department thinks that means having nuclear weapons systems, and the weapons designers request control systems to monitor radar signals, and the computer companies are asked to respond to six particular kinds of radar pattern, and the engineers are told to build signal amplifiers with certain circuit characteristics, and the technician is told to write a program to respond to the difference between a two-volt and a four-volt signal on a particular incoming wire." The story implies that the resulting model is far from the original intention after a series of modeling activities. If the modeler is not familiar with the intention of modeling and the task envi-

---

[1] See Smith 1985

ronment, it probably leads to a wrong model. If the user of the model does not understand the modeler's logic, it is likely that the model will be misused. Even if the modeler and the model user is the same person, modeling is still critical. Modeling fails if it cannot fulfill what it is supposed to do. The common roles of models include:

1. Faithful representation of the original.
2. Knowledge representation in the domain.
3. Supporting formal reasoning and manipulation.
4. Usefulness and effectiveness in particular tasks.

Following McCarthy and Hayes's (1969) categorization, the four notions would fall into two categories: the first two notions are more closely associated with "epistemological adequacy"; the notion 3 and 4 are more closely related with "heuristic adequacy." However, there is no definite boundary between the two categories. There have been many controversies about these notions so far. The first one and the last one are most essential. The following texts are organized as follows: subsection 4.1.1 exploits the problem of faithful representation (correctness). Subsection 4.1.2 discusses the problem of usefulness and effectiveness. Section 4.2 discusses the solutions to these problems of modeling.

4.1.1 Correctness of models

There have been enormous attempts to achieve the correctness of models. Faithful representation has been regarded by many studies as the fundamental goal of modeling. One interpretation of faithfulness refers to the isomorphism between the model and the referent (Ashby 1957). Yet many researchers conceded that there are no "correct" models, but there should be some similarities between the model and the referent in certain aspects. For example, van Fraassen (1980) suggested the models should be empirically adequate. It is widely agreed that models are approximations of the referents, i.e., the models contradict with the referents in certain aspects. As smith (1995) put it: "Every act of conceptualization, analysis, categorization, does a certain amount of violence to its subject matter, in order to get at the underlying regularities that group things together."

The issue of correctness is closely associated with the model verification. Carson (2004) explained how to verify a model:

> "First, make an hypothesis: the model is "correct". Second, try as hard as you can to prove the hypothesis is false; that is, try to prove that the model is "bad" in some way. If only after great effort, you have only confirmations and no evidence of a faulty model, then conclude (tentatively) that the model is verified."

However, such verification can never be complete, for the potential tests of the models are unlimited. It means that the faithfulness of the model is always temporary. In spite of various attempts to make faithful models, some scholars addressed that models cannot be faithful: "Essentially, all models are wrong, but some are useful" (Box and Draper 1987, 424); "A model is a work of fiction" (Cartwright 1983, 153); "It is meaningless to ask whether it (a mathematic model) corresponds to reality" (Hawking and Penrose 1996, 4). Furthermore, Smith (1985) pointed out the key problem:

> "We in general have no guarantee that the models are right - indeed we have no guarantee about much of anything about the relationship between model and world."

However, how can a model make sense if it is not faithful? Alternatively, what makes a model valuable? A few theories investigated this issue. For instance, Wiger (1959) proposed the "unreasonable effectiveness" of mathematical models; Peschard (2011) laid the value of models in their support of exploring the epistemic space of a domain. Wiger observed that a mathematical model for one phenomenon is often competent to address other phenomena. In his words, "mathematical con-concepts turn up in entirely unexpected connections. Moreover, they often permit an unexpectedly close and accurate description of the phenomena in these connections." In scientific fields, some mathematical models are employed for many unrelated problems, to name a few: graph, normal distribution, and the Markov chain. The mathematical models seem 1) faithful to the original; and 2) unexpectedly feasible for other systems. Wiger called it unreasonable effectiveness because "the enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and that there is no rational explanation for it." Yet, the problem is that people do not understand the correspondence between the mathematical models and natural phenomena. In response to Wiger, Hamming (1980) intended to explain the myth: first, we have always been looking for those most effective mathematics; second, we can always choose what kind of mathematics we are going to use. The resulting "illusion" is that the mathematical models always work and sometimes work unexpectedly.

As mentioned above, the relationship between the model and the reality is a critical issue. When we talk about the faithful representation of models, we implicitly separate the model world from the real world. We can compare the models with the "reality" in form of empirical data; however, the concept of reality is far from clear. A few theoretical studies implied that it is impossible to make a true observation of the world without bias, since people can only formulate the world from a specific point of view. Experiments play a tricky role in scientific modeling. In the leaning tower of Pisa, Galileo deliberately selected dense falling bodies to verify the model of "falling velocity is independent on the weight of the body." However he did not explain the legitimacy of ruling out the air resistant and other environmental influences. In contemporary science, the experiments are very sophisticated; otherwise, it is very difficult to find the correlations between the proposed model and the empirical data. Frigg and Hartmann (2012) pointed out that the experiments serve for the "distorted models": "Physicists build models consisting of point masses moving on frictionless planes, economists assume that agents are omniscient, biologists study isolated populations, and so on. It was characteristic of Galileo's approach to science to use simplifications of this sort whenever a situation was too complicated to tackle." The "distorted models" are carefully built to eliminate the unexpected aspects of the target system. In Rouse's (2002, 22) words, "Experimental science does not merely aim to represent the world; it intervenes in the world to enhance its intelligibility." In a similar manner, when a person perceives the world, he comprehends a "distorted" representation of the reality instead of the reality itself. It is often futile to speak about the objectivity of the reality. Hacking (1983) stated that the reality is an "anthropomorphic creation"—a second-order concept: "The first peculiarly human invention is representation. Once there is a practice of representing, a second-order concept follows in train. This is the concept of reality, a concept which has content only when there are first-order representations." It suggests that we cannot address "reality" without modeling; the reality and the model are a pair of interdependent notions.

Besides the model-world issue, the relationship between different models is also critical. Since models are incomplete representations of the original, there must be multiple models for the same original. Moreover, it is hard to find an objective metric system to evaluate the multiple models. This is the paradox of modeling. The paradox has raised both theatrical studies and pragmatic solutions.

Peschard (2011) investigated the way people choose a model among many candidates. She argued that it is not adequate to evaluate the model based on its original; rather, the model's contribution to the whole domain is essential. Peschard thought models help people explore the "epistemic space" of a domain. The epistemic space is about how people look at the problems in the domain, and how the knowledge of the domain is structured. In her own words, "An epistemic space is a set of fundamental questions or problems and what makes the construction of the model scientifically valuable is the new light it casts on such problems and the new terms it offers to deal with them." For example, suppose we are going to model two situations: the wake behind one cylinder (Fig 4.1 left) and the wake behind two cylinders (Fig 4.1 right). For the first situation, the scientists do not only care about how precise the "single cylinder model" is, but also care about how useful the model is for the second situation. Here the policy of choosing models is not only a scientific issue but also a political one. It suggests that scientists prefer versatile models to "ad hoc" models, even the former are less precise than the latter. Scientists pursue a set of interconnected models for all phenomena in a field. Peschard revealed an important relationship between models: one model is usually supposed to be a module of other models. As she put it:

> "The value of a model depends on how they facilitate people to explain all the related phenomenon besides the target phenomena. On that view, whether a model-of-X is epistemically valuable or scientifically worthwhile depends on the difference it makes in this epistemic space with respect to the investigation of scientifically significant problems."

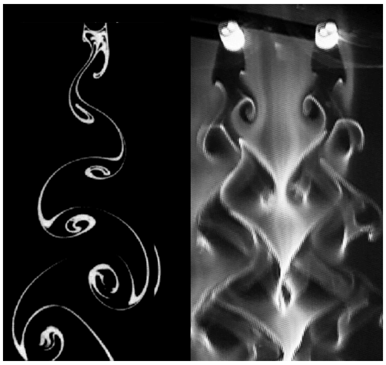


Fig 4.1: A single wake behind a cylinder (left). Double wakes behind a pair of cylinders (right). Peschard (2011).

In Peschard's picture, the phenomenon of multiple models would not be a problem since multiple models are good for exploring the epistemic space of the domain. Plainly, multiple (perhaps inconsistent) views instead of a single view on a subject are the momentum towards new knowledge. Hence, modeling is a successive learning process towards better models, but we cannot hope to attain a final correct model. This is in accord with Stevens (1990, 285): "A cardinal virtue of a theory or model is that it is falsifiable (and of course, not yet falsified)." We have to live with multiple models since a single correct model is not available.

Since the notion of correctness is problematic, the notion of encoding domain knowledge is also in question. In the early years of Artificial Intelligence, the domain knowledge was often formulated with certain ontology. Mylopoulos and Levesque (1983) interpreted the world as "a collection of individuals and relationships between them". Therefore, it is reasonable that the models can represent all possible "states" of the world. However, Newell (1981) had doubts on the static knowledge-model relationship: "Until a clear conception of the former (knowledge) exists, the latter (representation/model) will remain confused. In fact, this is not so." Rather than defining knowledge directly, Newell proposed the notion of knowledge level and then interpreted the knowledge as the medium between the knowledge levels. As Newell (1981) put it: "(the role of knowledge) is filled only indirectly and approximately by symbol systems at the next lower level…The knowledge cannot so easily be seen, only imagined as the result of interpretive processes operating on symbolic expressions." In his image, the knowledge and the models are a pair of interdependent concepts; one cannot be clearly defined without the other.

4.1.2 Effectiveness of models

The correctness of models and domain knowledge representation are concerned with epistemology. Nonetheless, there is another line of modeling that concentrates on the usefulness of models. Hence the problem switches from "what the models are?" to "how can we benefit from the models?" The pragmatic approaches to modeling were associated with the early problem-solving paradigm. However, recent studies (e.g. data mining and machine learning) in computer science also put emphasis on the usefulness of models, more precisely, the effectiveness of models (the balance between usefulness and costs). For instance, Pinsky and Karlin (2011, 1) stated that: "In the final analysis, a model is judged using a single, quite pragmatic, factor, the model's usefulness." From a different point of view, Rothenberg (1989) stated, "Modeling in its broadest sense is the cost-effective use of something in place of something else". In Rothenberg's image, the first question for the modeler is Is modeling better than nothing in the situation? Bender (1978, 1) answered this question: "The model is often modified, frequently discarded, and sometimes used anyway because it is better than nothing." The effectiveness of models depends on particular situations. The concerns usually include: 1) The effectiveness of the model in the task, and 2) the model's relationship with other models involved in the task. The second concern becomes very important in the design task that involves multiple, contingent models.

One advantage of formal models is that they support formal manipulations. For example, Boolean algebra represents the truths in the real world with truth-values (true and false), and people can manage "truths" with algebraic operations. For instance, the formula $\neg(x \vee y) = \neg x \wedge \neg y$ means "the result of checking whether x or y is true is opposite to the result of checking whether x and y are both false". In this case, the symbolic model is clearer and more effective than a verbal description. However, the model is inherently partial. Thus, the model may contradict with the original in certain aspects, especially when the task environment has changed or the modeler's understanding of the task has changed. Sometimes the model is very efficient in formal operations but it fails to address the real problem. For instance, consider the design of a roof, the variables are the shape (flat, dome, double-pitched, and hip roof) and the material (tiles, metal, and concrete). There will be 3×4 combinations as the candidate solutions. If a new variable color (green, grey, and crimson) is introduced, there is a three-dimensional design space. Zwicky's (1969) Morphological Box is ideal for modeling such combinatorial problems (Fig 4.2). The layer, the row and the column of the box correspond to the shape, the color and the material respectively. Each cell of the box represents a roof design. However, the model may not be very helpful for an architect because the essential interrelationship between these variables is hidden. To draw a short conclusion, some models are systematic and efficient in formal operations; however, it does not mean that the model effectively entails the key aspects of the subject matter.

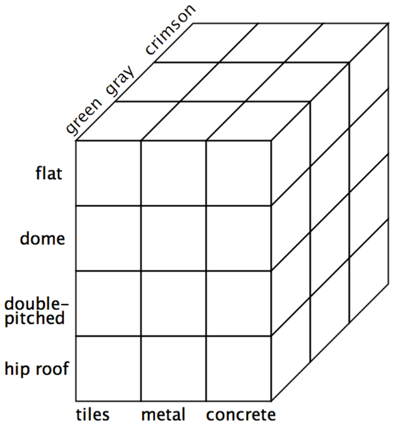


Fig 4.2: A morphological box for three variables.

The balance between the efficiency of formal manipulation and the representation capability of models is also an important issue in architecture. Many CAD systems only represent geometric entities, such as lines, curves, polygons and meshes. The symbolic manipulations for processing geometric entities are clear and efficient. However, a vast number of studies pointed out that geometrical models cannot support high-level manipulation in architectural design. As Eastman and Siabiris (1995) put it: "A full representation of a design consists of representing a range of intermediate concepts lying between the human intention and the actual physical structure of a product." They proposed a generic building product model that contains richer information of buildings than traditional CAD systems. However, the drawback is that the formal manipulations become much more complicated than that of geometrical models. The complexity may confuse the users and increase the working time. Nowadays BIM (Building Information Modeling) models are unprecedentedly sufficient in catching rich information of architecture and subsequently capable of supporting some high-level manipulations in design. Yet, the manipulations of the BIM model could be too complex for architects, and it is painful if the defined manipulations do not fit the architects' intentions. A good model could be both highly versatile and detailed, but it might be quite difficult to fit the model into a particular task environment. It indicates that effectiveness does not necessarily lead to a single general model.

Since the 2000s, the method of machine learning based on enormous data has presented a contemporary concept of effectiveness. Google researchers Halevy et al. (2009) proposed a learning model for processing natural language (e.g. translation). The actual structure of the model is learned from the data set, in their own words, "relying on overt statistics of words and word co-occurrences." The effectiveness of the model is addressed in several aspects. First, the structure of the model is simpler compared with traditional models. Second, it avoids manual work such as labeling and segmentation. Third, the same model can be applied to different languages; as a result, translating between English and French is not so much different from translating between Chinese and Italian (the difference is only the data set).

To summarize, there have been distinct approaches to the effectiveness of models. Some focused on the efficiency of formal manipulation, some emphasized the generality and versatility of the models, and others tried to learn models from data set. Here we can find some paradoxes. If the effectiveness of formal manipulation is the primary concern, the actual effectiveness of applying the model to particular situations cannot be guaranteed. However, if the model is calibrated to the particular task in minute detail, the formal manipulations in the model are probably cumbersome. Another paradox is raised by multiple models. Suppose model A is most effective for task a, and model B most effective for task b. If a new task involves both tasks, can we simply use both models? It is problematic if the two tasks are overlapping and the two models are not coherent.

4.2 Metamodeling and Combined Modeling

4.2.1 Metamodeling

Metamodeling is going to solve the dilemmas in modeling. Yet, there are quite distinct approaches. The common notion of "metamodeling" in computer science[2] seems narrow. The prefix "meta-" comes from the Greek word "μετά". It often re-

---

[2] van Gigch 1991, Kühne 2006, Bézivin 2005.

fers to a higher level of abstraction. For example, metaphysic is interpreted as "the science of physics." Hence, a metamodel is usually regarded as "the model of models." However, this work regards metamodeling as the mastership of multiple models. Here the prefix "meta-" means "between[3]", hereby metamodeling makes the interplay between multiple models possible. This notion of metamodeling is orthogonal to (if not opposite to) that of the contemporary studies.

The common notion of metamodeling usually leads to the hierarchy of "world-model-metamodel". For instance, van Gigch (1991) said, "METAMODELING is MODELING at a higher level of logic and of abstraction…one step further removed from the real world of objects and things." Such formulation needs a certain definition of the relationship between the model and the original (target system). According to Bézivin (2005), the system is "representedBy" the model, and the model "conformsTo" the metamodel (Fig 4.3). More precisely, he explained that "A metamodel is a formal specification of an abstraction…from a given system we can extract a particular model with the help of a specific metamodel. A metamodel acts as a precisely defined filter expressed in a given formalism." Slightly different from Bézivin's definiton, Kühne (2006) defined that model is "instanceOf" metamodel, and metamodel is "instanceOf" meta-metamodel (Fig 4.4).

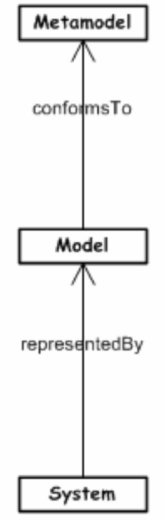


Fig 4.3: System, model and metamodel, Bézivin (2005).

---

[3] "Beween" is one of the original meanings of "μετά", Liddell and Scott 1940.

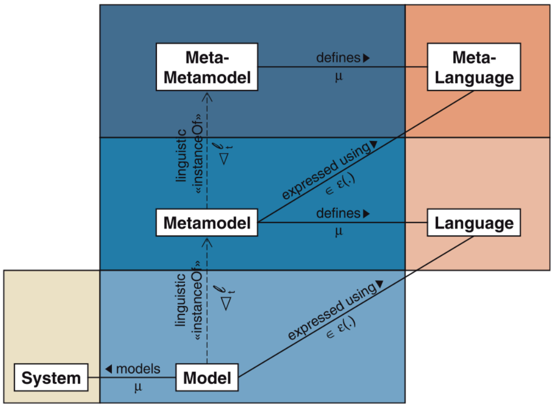


Fig 4.4: System, model, metamodel, and meta-metamodel, Kühne (2006).

In the field of software engineering, metamodeling often leads to the Unified Modeling Language (UML). In accord with Kühne (2006), OMG UML document (2007) stated, "A model that is instantiated from a metamodel can in turn be used as a metamodel of another model in a recursive manner." The document even defined a four-layout hierarchy: M0: run-time instances of model elements; M1: Model; M2: metamodel (UML language); and M3: Meta-Object Facility (MOF) — the standards for defining UML. Besides, the metamodel represents both the elements and the relationships between elements. For example in the case of Fig 4.5, the model "Person" and the model "Car" are instances of the metamodel "Class". The association between the two models is an instance of the metamodel "Association." Moreover, both metamodels are instances of the "meta-metamodel."

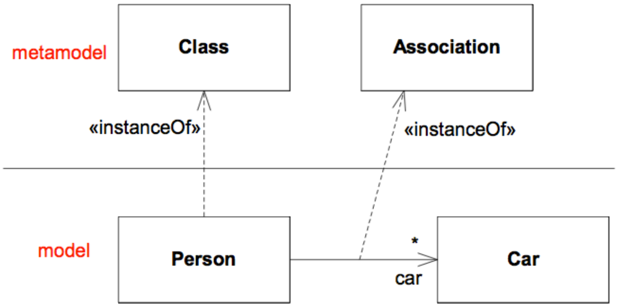


Fig 4.5: Model and Metamodel in UML, OMG UML (2007).

It is not difficult to find out whether the common notion of metamodeling leads to the "meta-meta-meta-…model" hierarchy. In contrast with building a hierarchy of abstractions, this thesis suggests combining multiple models without abstraction.

Suppose multiple models are involved in a task, metamodeling should enable them to work together. Hence, the metamodel is a contingent aggregation of models. If we compare the two notions of metamodeling, one is to conquer the multiplicity of models by making abstractions repeatedly; the other is to accommodate multiple models by connecting the models.

4.2.2 Combined modeling

The previous subsection describes two notions of metamodeling. One suggests making a model of models. The other suggests combining models. Both approaches intend to solve the problems in modeling. The problems come from the nature of the model: a partial representation of the original. Different views of the target would result in different models. Modeling represents the complex, infinitely variegated [4] reality with definite models, but a new problem arises: the models essentially don't agree with each other. To solve this problem, one can 1) Construct a new (better) model from scratch; 2) Make the model of models; or 3) Combine the multiple models. The first is modeling. The second pertains to the common notion of metamodeling. The subsection discusses the third. It is going to show that combining models will not be worse than the other two.

One model presents one particular view of the original and the particular purpose of the modeler. Thus choosing an appropriate model is critical. People may consider the correctness of the model (subsection 4.1.1), or take account of the effectiveness of using the model (subsection 4.1.2). Nonetheless, it is common that multiple models are involved in one task, with ambiguities and conflicts between them. First, it is not promising to presume that there is one best model for a given problem. Pinsky and Karlin (2011) asserted that: "There is no such thing as the best model for a given phenomenon. The pragmatic criterion of usefulness often allows the existence of two or more models for the same event, but serving distinct purposes." To resolve the problem, it is worth examining the model-world relationship again. The common understanding is that the signs in the model "refer" to the objects in the real world, and subsequently the objects are "represented" by the model. However, this model-world relationship does not always make sense. For instance, the imaginary number is originally derived from mathematical operations and it refers to nothing "real" in the world. In scientific fields, there are a number of "abstract" models that are only good for supporting the constructions of other models. Thus the reference to the real world is not essential for the models. In Allgemeine Modelltheroie (Stachowiak 1973), the model refers to its "original" that can be either real or fictional. We can either describe something that already exists, or invent a new subject matter with models. Then it is reasonable that there are multiple

---

[4] Smith 1985

models for in one design task, since people may model the same problem differently, or model different problems.

The problem of multiple models frequently occurs in architectural design. For instance, if the client specifies a room by width and depth (assuming the room has a rectangular shape), but the designer models the shape of the room with an irregular polygon. There is no obvious solution for such problem. Rosenman and Gero (1996) formulated such problems as “multiple views of objects.” They criticized that CAD systems usually provide a fixed and static model that cannot accommodate multiple views of the participants in the design environment. For instance, architects use floors and walls to define spaces, while structural engineers view them as structural elements. Even one architect may have different views on the same problem when the design proceeds. Rosenman and Gero illustrated the “overlapping” between multiple models in Fig 4.6. In this image, some elements belong to one particular model; others are shared by more than one model. Here the concept of “shared elements” is a little bit misleading because it doesn’t distinguish the signs (signifier) from the originals (signified). One element (sign) of model A is not the same as one element (sign) of model B, even the two signs refer to the same original. Nonetheless, it is important that they explicitly revealed the problem of multiple models in architecture.

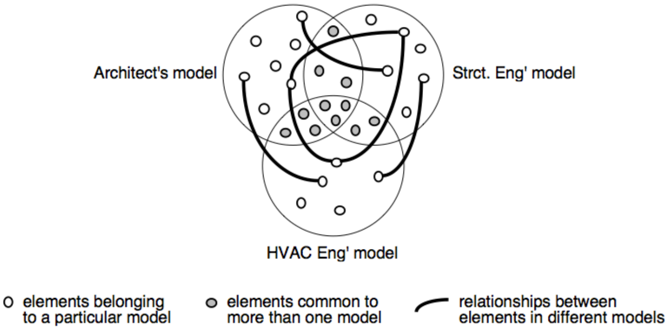


Fig 4.6: Multiple models, Rosenman and Gero (1996).

The critical question is how to deal with multiple models. Rosenman and Gero’s (1996) result is “a set of models where each model has its own concepts and elements,” which is very close to this thesis. However, they finally fell into the common metamodeling paradigm, as they sought a general model from that different participants can derive their own models. In accord with them, Watanabe (1994) suggested to “make an integrated architectural meta-model which can represent and generate various specific types of knowledge and information of every subfield in architecture.” This is a typical metamodeling approach to architecture, which is going to be examined carefully here. Why do people attempt to make abstractions of multiple models? We find a clue in Dave and Woodbury's (1990) words: "We feel that it is important to present a single coherent view, and we thus have to embrace a single overall model."

Many people find that it is painful to comprehend the coexistence of multiple models, for the epistemic frameworks underlying the models are not consistent. Broadbent (1973) gave the reason why using multiple modes is nonsense: "Nor should we necessarily seek to combine them (different models). This would lead to a lowest common denominator, thus blurring our focusing on different aspects which was the strength of our original models and which lead to the value of comparing them." However, is the distraction of focuses harmful? We have to notice that multiple models result from the nature of modeling. Thus being aware of the different characters of multiple models is more reasonable than focusing on one model. In fact, combining multiple models does not necessarily destroy the strength of the models; rather, it could lead to the synergy of models. If each model can address its own concerns without unacceptable conflicts with other models, the combination of models is promising.

Combined modeling is an alterative to metamodeling as it handles the intrinsic problems of modeling. The contemporary approaches to metamodeling argue for metamodel, for multiple models can be derived from different configurations of one model (the metamodel). By contrast, this thesis suggests combining multiple models to let them work together. The former results in one model; the latter leads to a set of associated models. The former seeks the consistency in general, making high-level abstraction over various task environments; The latter allows inconsistency in general, but let the models reach agreements in specific tasks.

4.2.3 Fictional combination of models and architectural design

Combined modeling is not only error elimination but also an organization activity requiring imagination and creativity. Cartwright (1983) stated that models are fictional. Combined modeling is fictional use of models, because it does not fit the epistemological framework of each model nor does it follow the purpose of each model. The combination of models is a fiction based on genuine models of the real world. There have been studies on the fictions in science. Fine (2009) discussed the "semi-fiction" that are in contradiction with reality but not self-contradictory. Moreover, the role of scientific fictions is indispensable since they convey important information with its obvious fictional character. In accordance with Fine, Barberousse (2009) stated, "Representing fictional situations gives them (models as fictions) the power to convey new usable scientific knowledge."

It seems that fictions made of models are suitable for architectural design. Architects usually 1) allow the coexistence of several models; and 2) re-contextualize a model to convey new meanings. In a sense, architectural design is about fictional organization of models, since it often deals with multiple models of architecture and transforms their meanings by combining them in a novel way. Such combination of models has the same characters of fictions: 1) referring to the real world; 2) contradicting the common understanding (model) of the subject matter[5]; and 3) conveying meaningful information. Combined modeling is a formal game on a higher-level of the original models; however, it appreciates the specificities of the models rather than generalizes the models.

A model is re-contextualized when it is put into a new environment. Usually a model is bound to its target and subject to the modeler's purpose. Neverthless, we separate the model from its original target and purpose. It is a sort of de-modeling since it undoes what the modeling process does. The resulting "naked model" loses its original role and subsequently is capable of taking on new roles. By assigning new references to the model or by connecting it to other models, the model gets a new role with new functions and meanings. This process is a sort of re-modeling. Hereby we re-contextualize the model, i.e., we transform the model for our own purposes without making substantial changes to the model. Combining models usually involves successive re-contextualization of multiple models, until there are no unacceptable conflicts between them and the whole aggregation of models fit the new purpose. Thus, combining models is not modeling—it is a fiction making instead. Modeling usually takes account of three aspects: 1) the nature of the original, 2) the inner structure of the model, and 3) the relationship between the model and the original. Yet, fictional combination of models is characterized by the transformation of the models through re-organizing the relationships between the models. It is a sort of metamodeling for it manages models rather than makes models. The outcome of such metamodeling is a fictional aggregation of models instead of a metamodel of models.

We can break models into parts and assemble them in new ways. However, the simplest way of combination is just constructing the connections between models, i.e., specifying how one model exchanges data/information with other models. Here the main concern lies in the overall behavior of the connected models. Connecting models is obviously valid in terms of formal operations; however, it is not easy to capture the meaning of the approach. For instance, it is convenient to apply the "square root" operation to a negative number, but it took centuries for mathematicians to make sense of it. Using the CFD (computational fluid dynamics) mod-

---

[5] : Barberousse and Ludwig (2009) enounced: "fictions rather convey contents that contradict our best theories about physical, biological, psychological, and also metaphysical matters."

el for form finding in architecture (Verebes 2009) is operationally valid in computers, but it is not easy to comprehend its meaning. The easy doing (applying formal operations) and difficult understanding (making sense of the operations) is a common phenomenon in today's modeling. Sometimes we have to accept the situation that the models are working but we do not fully understand how they work. It is not reasonable to abandon a tool just because we do not understand it. For instance, artificial neural networks had good performances in learning and predicating, but some people rejected them because they are difficult to understand. In Francis's (2001) words: "Despite their advantages, many statisticians and actuaries are reluctant to embrace neural networks. One reason is that they are a 'black box'." The emergence of new computational models suggests that it is better to study the models by running/using them than by analytic study.

Thanks to computers, it is convenient to run the models and to experience their actual behaviors. Sometimes the important behaviors of the model can only be observed by running the model. In practice, the mastership of modeling tools (say a programming language or software) is essential for creating, testing and modifying the models. As Schumacher (2009) stated, "Computationally advanced design techniques such as scripting and parametric modelling are becoming a pervasive reality such that it is no longer possible to compete within the contemporary avant-garde architecture scene without mastering and refining them." The importance of running the models is not only due to instrumental concerns but also to methodological concerns. It is not a rare circumstance that the designer cannot adequately understand how the model works, but the actual behaviors of the model are essential for the designer's decision-making. Running the model is "to act," so that the modeler can get the feedback of how good the model is. Smith (1985) stated, "You often learn, when you do act, just how good or bad your conceptual model was." Running the models is especially important when the models are connected in a novel matter, because the actual behavior of each model may differ from normal conditions. The actual behavior of the connected multiple models reflects the complexity of the real design environment.

Essentially, models can be combined in many ways, which indicates that design issues can be associated with each other in many ways. We can make a fictional combination of models for our own purposes. The novel combination would expose the contrasts and the conflicts between the models. The diversity of models is resulted from the complex design problems, and the fictional combination presents the designer's new position in dealing with the complexity.

Design is a complex activity because it does not only solve defined problems but it also searches for new problems. Thus, it is impossible to make a fixed and static model for design. Rittel (1973) termed design problems with wicked problems, because they do not have definite formulations, have no stopping rules, and they are essentially unique. Rittel (1988) described design as a process of argumentation:

> “The various issues are interconnected in intricate ways, usually several of them are 'open' simultaneously, others are postponed or reopened. He finds himself in a field of positions with competing arguments which he must assess in order to assume his own position.”

It is feasible to support such argumentation with a combination of multiple models. Each model takes account of a set of issues and presents a view on the issues. When two or more models are combined, the relationships between all the issues become complex; the views on the issues become contradictory. The architect’s task is addressing all the problems properly by combining the models in a particular manner. In other words, the design concept is embodied in the interrelationships between the models. A valid aggregation of models contains more rich and meaningful information of the design than a single model.

Chapter 5

Combining Models in Architecture

Experiments with Computers

Following the method of combining models that is proposed in Chapter 4, this chapter introduces a series of experiments with computers. The experiments are designed and programmed by the author. The computer programs are all written in Java. The programs employ a wide range of architectural models: volume, grid, shapes, topology, grammar and so forth. The motivation of the experiments is to explore the possibility of combining two or more models rather than to investigate each model. Most models are common in the field of computational design; however, few studies have tried to combine these models. The experiments are going to demonstrate whether the fictional combination of multiple models could be productive in architectural design.

5.1 Volume + Grid[1]

5.1.1 The two models: volume and grid

The experiment combines the two basic models of architecture: the grid and the volume. The grid is set of parallel/orthogonal axes. It is very helpful to organize the positions and the orientations of various architectural elements. The volume model defines the extents of the spaces, based on the insight that the spaces are the protagonist of the architecture. Rather than studying the two models respectively, the interplay between the two models is the main focus of the research. It is convenient to use just one model, if the gird is coupled with the volume, i.e., the axes of grid are aligned with the boundaries of the volumes. Nonetheless, the assumption of alignment is not necessary. Once we decouple them, we have to manage the differences between the two models.

The research is partially inspired by Colin Rowe's (1947) "The mathematics of the Ideal Villa". The paper revealed the rhythm of the grid underlying both Le Corbusier's and Palladio's villa design. The proportion of the grid is 2:1:2:1:2 in one direction, the other direction takes the interval of 1.5 and 2 (1.5+0.5). It seems that

---

[1] The content of this section uses the material from Hua (2012).

Rowe assumed that the volume and the grid of the architecture are well coupled. By contrast, some deconstructivism architects strove to evoke the conflicts between the two models. Re-organizing the conflicts leads to meaningful compositions. Eisenman's diagram (Fig 5.1) explicitly illustrates the interplay between grid and volume.

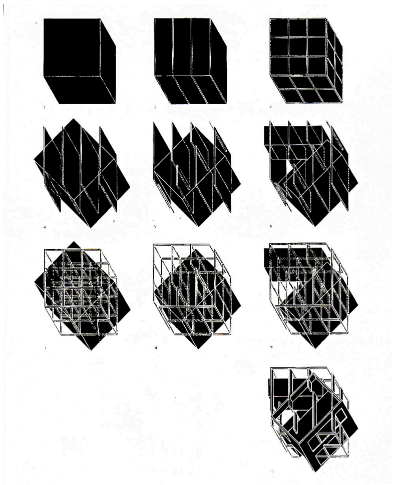


Fig 5.1: The interplay of volume and grid, Eisenman.

The model of grid or the model of volume could be employed as the generative system for architecture. Many approaches to spatial synthesis employed either of them as the main model, but few approaches tried to combine both models. Many researchers believed that dealing with one model is more feasible than using multiple models. Despite that, this experiment investigates how the two models could work together. The strategy for managing the conflicts between the two models is the focus of the program.

5.1.2 A synthesis program

The program arranges the rooms and the functional units (e.g. entrance hall, terrace, and staircase) on a grid and within a single cuboid volume. The grid adopts the rhythm of the grid in Villa Stein (Rowe 1947). The intervals of the grid repeat the rhythm of 4:2 (in meter) in one direction and 1:3:3:3:1 in another direction. The cuboid volume is divided into several layers (each layer corresponds to one floor). These layers of volumes are further subdivided into smaller volumes by the grid. Since the gird is not aligned with the cuboid volume, each subdivided volume could be either a cuboid or a more complicated volume resulted from the intersection between the cuboid volume and the grid (Fig 5.2).

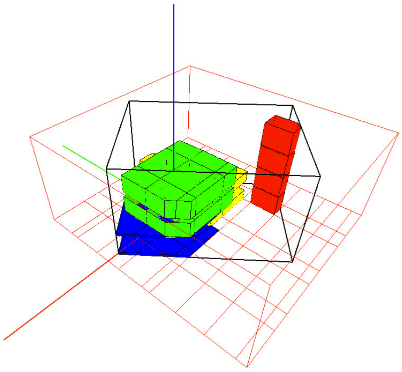


Fig 5.2: The grid is in red, the cuboid volume in black.
Several functional units are placed on the grid and within the volume.

The whole volume is occupied by two kinds of spaces: the unspecified rooms and the specified functional units. The program only arranges the functional units. The functional units occupy the subdivided volumes. One unit could occupy more than one unit. Four functional units are defined: a two-layer high entrance hall, a staircase, a two-layer high conference room and a terrace. The units are randomly initialized under certain constraints. For example, the entrance hall is located on the first layer and must be accessible from outside. The area of each unit is also constrained.

The algorithm searches the proper composition of functional units. First, the position of the staircase should facilitate the circulation. The circulation criteria include:

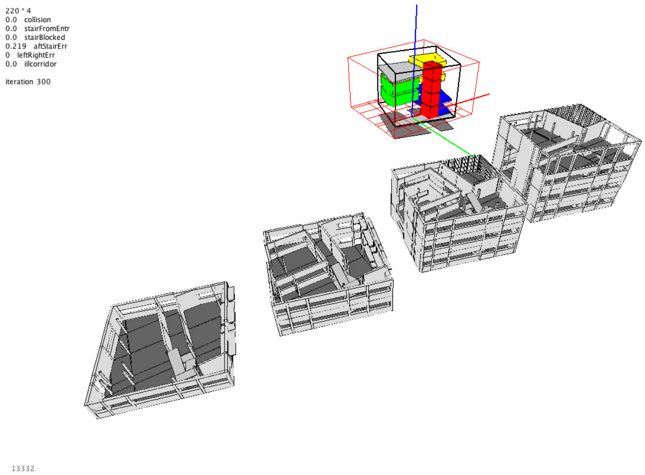
1. The staircase should not be blocked by other functional units
2. The staircase should connect the entrance hall directly.
3. Suppose there is a straight corridor starting from the staircase, this corridor should not be interrupted by other functional units.
4. The position of the staircase should be proper in the plan (corresponding to $E_3(x)$ and $E_4(x)$ in the error function)

Second, the composition should avoid the collisions between the functional units. According to these considerations, the error function for one composition is defined as:

$$E(x) = \sum w_i \, E_i(x)$$

$E_1(x)$: 0 (if the staircase directly connects the entrance hall) or 1 (otherwise).
$E_2(x)$: the number of units that block the staircase.
$E_3(x)$: the area of the regions behind the staircase[2] on the ground floor.
$E_4(x)$: the difference between the area of the region on left side of the staircase and that on the right side[3] on the ground floor.
$E_5(x)$: the number of units that block the straight corridor starting form the staircase.
$E_6(x)$: the number of collisions.

A "generate and test" algorithm is sufficient for minimizing the error function. During each iteration of the search algorithm, a new composition is generated based on the current one (by slightly changing the current composition). If the error of the new solution is smaller, the algorithm adopts the new one. Otherwise, the new one is abandoned. Under most circumstances, a satisfying solution can be found after hundreds of iterations.



```
220 * 4
0.0   collision
0.0   stairFromEntr
0.0   stairBlocked
0.219   aftStairErr
0   leftRightErr
0.0   illcorridor

iteration 300
```

Fig 5.3: Search for the proper composition based on the volume and the grid.

The program is implemented in Java. Usually it takes less than five seconds for the computer to generate a solution. The compositions in the solutions exhibit a wider range of patterns. A set of additional rules are programed for generating the facade according to the underlying units (e.g., the entrance hall has big openings on the facade). As a result, the patterns on the facades reflect both the rhythm of the hidden grid and the volumetric compositions of the function units. Three cases (out of many others) are shown in Fig 5.4-5.7. It seems that the interplay of the volume and the grid could result in novel compositions. The program implies that multiple

---

[2] Suppose the region behind the staircase is difficult to reach (making circulation difficult).
[3] A big difference between the two areas indicates that the position of the staircase is not efficient.

models could be feasible in spatial synthesis. The model of grid and the model of volume could play distinct roles and simultaneously interplay in one process.

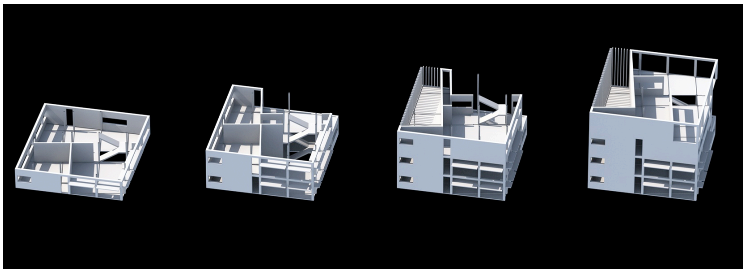


Fig 5.4: Solution A, perspective.

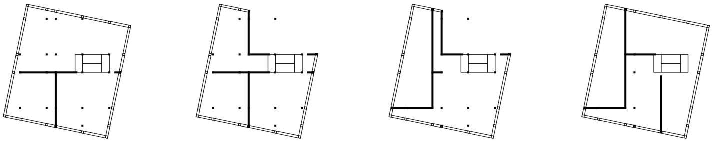


Fig 5.5: Solution A, floor plans.

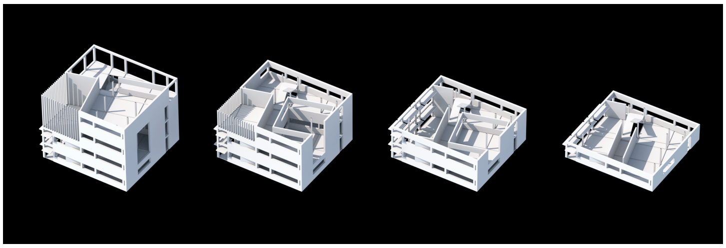


Fig 5.6: Solution B, perspective.

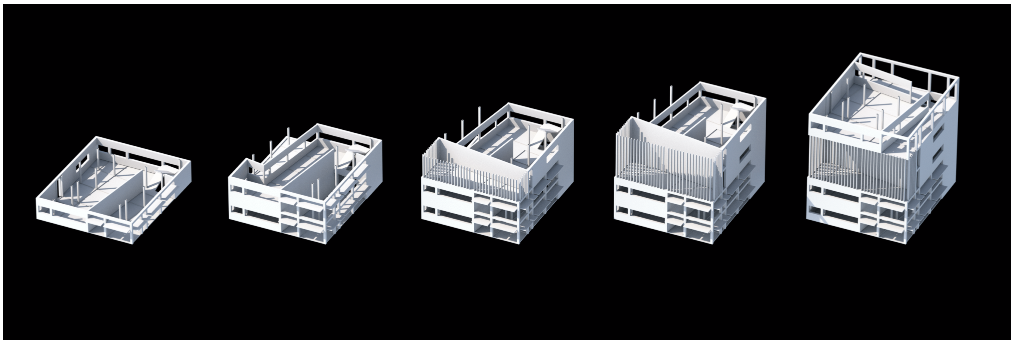


Fig 5.7: Solution C, perspective.

5.2 Topology and Form-finding

5.2.1 Topology + Grammar

This program combines a shape grammar with a topological model of buildings. According to Stiny and Mitchell (1978), Shape grammar can formulate a corpus of buildings that share certain likenesses. For instance, they gave a shape grammar of the Palladian villa. From a linguistic point of view, the shape grammar can characterize the vocabulary and the syntax of architectural design. The vocabulary refers to the elements and the operations. The syntax specifies how to put elements and the operations together.

The experiment focuses on the interaction between the shape grammar and the topology model. In contrast with the standard shape grammar, the grammar in this experiment employs an incomplete syntax, so that the articulations of the grammar heavily depend on the topology model. In this case, neither the grammar nor the topology model can work independently; rather, they can only work together. The grammar has three “nouns”: c (circle), q (quad), and t (triangle), see Fig 5.8. They are the leaf nodes of the tree structure of any sentences. The four “verbs” are D (divide), M (mirror), S (subtract), and R (replace). The verbs could be applied to the nouns or to the result of the operations (made by the verbs). Beside this elementary rule, there are no other syntactic rules.

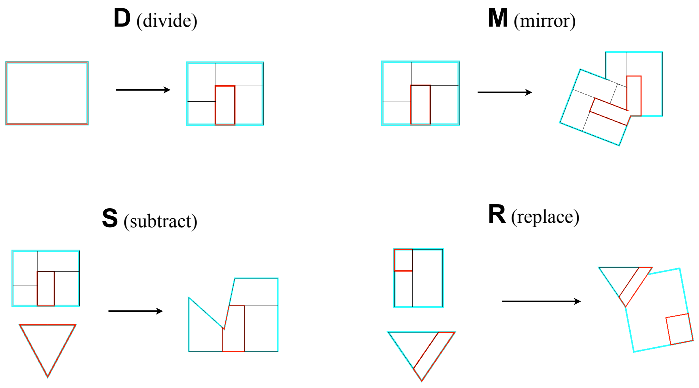


Fig 5.8: The four “verbs” of the grammar. The public space is in red.

The program randomly generates a sentence that can be represented by the tree structure. For instance, the four operations (sentences) in Fig 5.8 can be written as:
D
|
q
(upper-left, dividing a quad)

```
M
|
D
|
q
```

(upper-right, mirroring)

```
   S
  / \
D    t
|
q
```

(bottom-left, subtracting one element with a triangle)

```
   R
  / \
D    D
|    |
q    t
```

(bottom-right, replacing/cutting one element with the other)

The generated sentences (geometry) are parametric, because all the operations have a set of parameters. For instance, the D (divide) operation has several patterns of division and the M (mirror) operation has a parametric axis for mirroring. Therefore, one sentence consists of many geometrical variations that will be selected by the topology model. In order to interact with the topology model, both the nouns and the results of the operations of the grammar produce two kinds of spatial enclosures: the public spaces (in red, Fig 5.8) and the private rooms. If two public spaces are overlapped during the operation, the two spaces will be merged into one space (upper-right, Fig 5.8). The topology model checks the topological characters of the products of the grammar model. If the product (geometrical composition) has more than one public space, it will be abandoned. Otherwise, the product is valid. As a result, the final product always has only one public space with all rooms attached to it.

Both the grammar model and the topology model are implemented in the computer with Java. Therefore, the computer can automatically derive the valid sentences (geometrical compositions) from the interaction of the two models. The depth of the tree structure varies within a certain range, so that the complexity of the compositions can be roughly controlled. Parts of generated results are shown in Fig 5.9-5.10. The results imply that the interaction between the incomplete grammar and the simple topology model could produce diverse compositions.

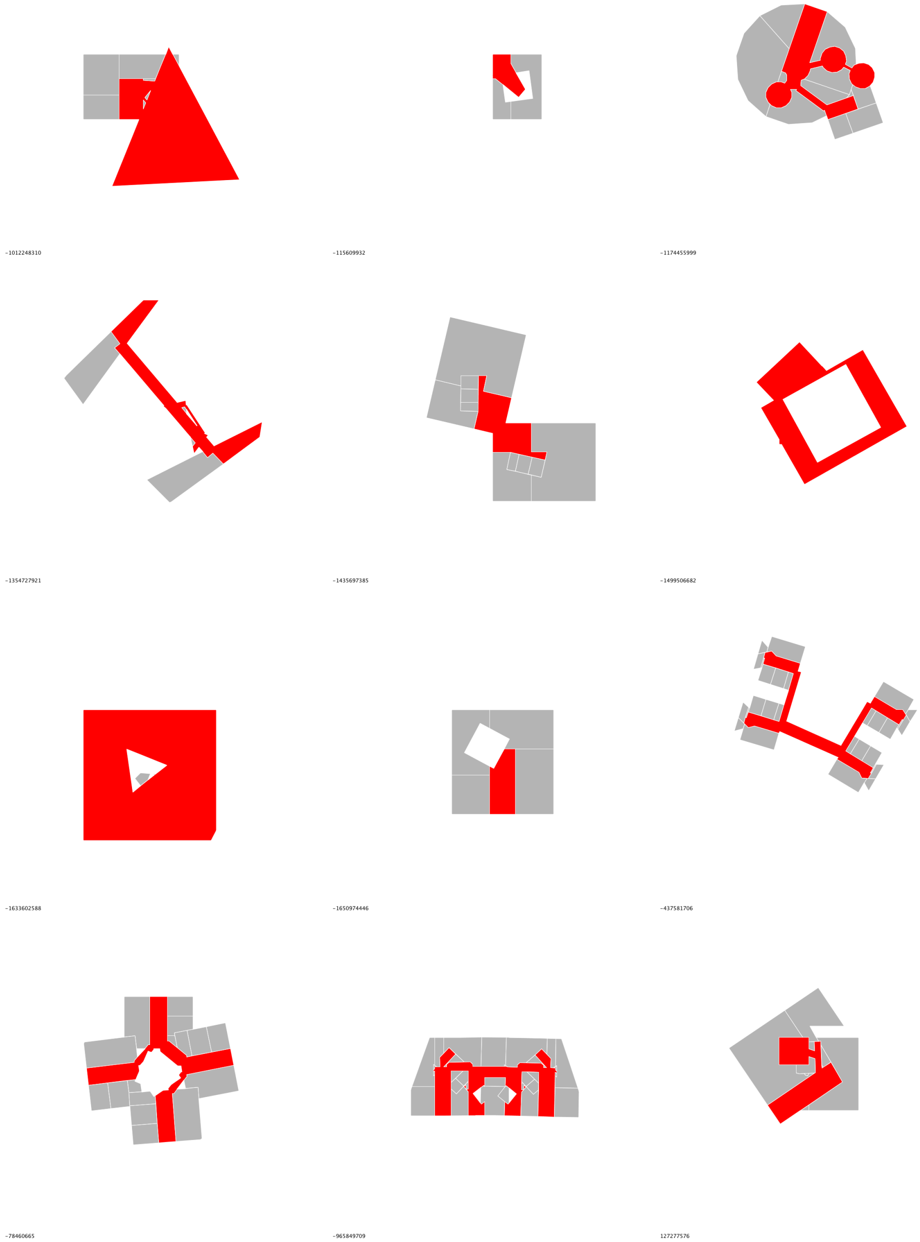


Fig 5.9: Collections of generated compositions, set A.

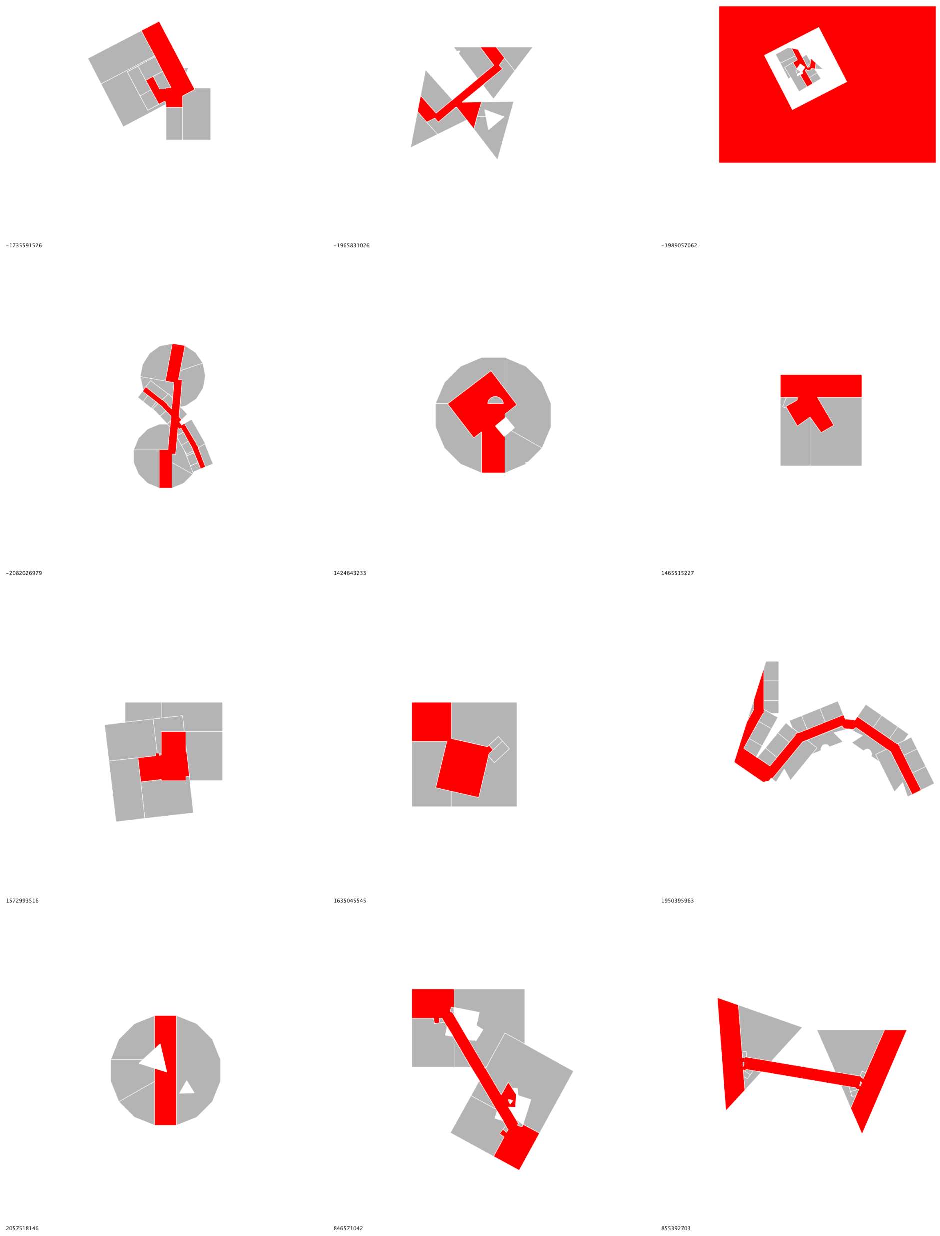


Fig 5.10: Collections of generated compositions, set B.

5.2.2 Topology + Shapes

Correlating the two models

Floor plan layout synthesis is closely associated with two kinds of models: the spatial model (representing the rooms) and the solid model (representing walls, columns and other architectural elements as solids). The former is essential for addressing the topology and the dimensions of rooms, while the latter is natural for representing the physical body of the buildings. There is a mutual relationship between the two: the spaces are enclosed by the solid elements and the solid elements serve as the boundaries of the spaces. Most approaches to floor plan synthesis employed the spatial model (often based on rectangular rooms). Yet, this experiment employs both models: a topology model and a shape model of solid walls.

The topology model describes the topology of rooms and the dimension of each room. By representing each room as a node, the desired topology can be represented by a graph $\boldsymbol{a}$. Thus we can define a cost function $T(M)$ measuring the differences between the desired graph $\boldsymbol{a}$ and a generated graph M. For the dimensional property of each room, we can measure the error (or fitness) of any generated room shape with a cost function: $G(\{r_i\})$, r_i: the ith room.

To communicate the shape model of solid walls with the topology model, we need the two functions:

$$M = m(\boldsymbol{v}) \qquad (1)$$

$$\{r_i\} = r(\boldsymbol{v}) \qquad (2)$$

$\boldsymbol{v}$: the parameters of the shape model.
$r(\boldsymbol{v})$: calculates (the shape of) the rooms out of the shape model.
$m(\boldsymbol{v})$: calculates the adjacency graph M of the rooms.

Then the cost function of the generated floor plan is:

$$C(\boldsymbol{v}) = w_1 T(M) + w_2 G(\{r_i\}) = w_1 T(m(\boldsymbol{v})) + w_2 G(r(\boldsymbol{v})) \qquad (3)$$

This function builds a bridge between the shape model and the topology model. Hence, correlating the two models is to minimize the function. However, it is not proper to understand this approach as optimization; rather, it just enables the two models to communicate.

The program takes a desired specification of rooms (Fig 5.11) and a shape model (Fig 5.12) as inputs, thereby producing valid floor plans as outputs. The specification of rooms includes the desired topology $\boldsymbol{a}$. $\boldsymbol{a}_{i,j} = true$ means room i and room j should be adjacent to each other; when $\boldsymbol{a}_{i,j} = false$, the two rooms could be either adjacent or not adjacent to each other. The dimension of each room is specified by a rectangle with the “ideal” area and ratio. The generated floor plan should meet the topological and dimensional requirements by adjusting the parameters of the shape model. The process actually minimizes the cost function (3).

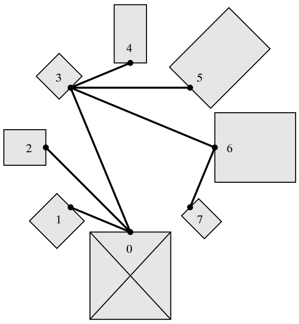


Fig 5.11: The specification of rooms: the dimension of each room and the topology between the rooms. The room marked with a cross must be accessible from outside.

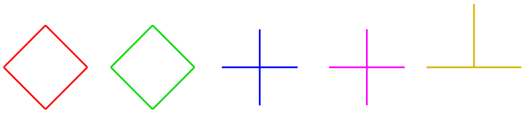


Fig 5.12: One shape model specifying a set of parametric shapes.

Shape model

The shape model defines a set of parametric 2-d shapes (Fig. 5.12) like “T” shape, quadrilateral, cross and so on. For instance, the “T” shape has seven parameters:
x: the x coordinate of the shape.
y: the y coordinate of the shape.
θ: the rotation of the shape, in radian.
m, l, n, a: the parameters controlling the shape, denoted in Fig 5.13.

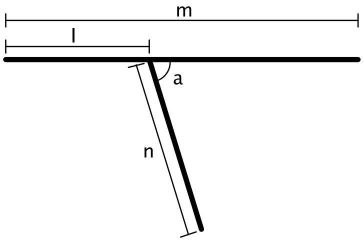


Fig 5.13: The parametric “T” shape.

The parameters of a shape can be put into one vector s. Hence, all the shapes in one model can be represented by a vector of vectors:

$$\boldsymbol{v} = (s_i)$$

$\boldsymbol{v}$: the vector specifying the state of the shape model.
s_i: the vector specifying the state of the ith shape.

Detecting enclosed rooms from shape model

Function (1) and function (2) quest for enclosed rooms resulted from the shape model. The room detection builds the bridge between the shape model and the topology model. First, the polygon detection algorithm calculates the intersections between the line segments from the shapes. Then the search procedure runs until all segments have been processed twice. The routine for finding the polygon r_i by processing segment $b_p b_q$ (the segment is directed, so $b_p b_q \neq b_q b_p$) is as follows:

```
if $p(b_p b_q)$ = true, return
$b_0 \leftarrow b_i \leftarrow b_p; b_j \leftarrow b_q; add(b_i); add(b_j)$
while true
    $b_c \leftarrow next(b_i, N_j)$
    $p(b_j b_c) \leftarrow true$
    if $b_c = b_0$ break
    $add(b_c)$
    $b_i \leftarrow b_j, b_j \leftarrow b_c$
```

$p(b_p b_q)$: true if the segment $b_p b_q$ is already processed, otherwise false.
$add(b)$: adds vertex b to the polygon r_i as the last vertex.
N_j: the neighbor points of point b_j, the points are ordered clockwise.
$next(b_i, N_j)$: returns the next point of b_i in set N_j (clockwise ordered).

The above routine produces polygon r_i. So all detected polygons make $\{r_i\}$ for function (2)(3). If two polygons share one segment, they are connected. Therefore, an adjacency graph M can be constructed (implementing function (1)).

Meeting topological requirements

There are usually more detected polygons than the number of desired rooms. Thus, only a subset of detected polygons $\{r_i\}$ is selected as the rooms. When a subset of polygons meets the topological requirements, the first term $w_1 T(m(\boldsymbol{v}))$ of cost function (3) equals to zero. A "trial and error" algorithm is employed for meeting the topological requirements. Each attempt is as follows:

1. Changing $\boldsymbol{v} = (s_i)$, the parameters of the shape model.
2. Calculating $M = m(\boldsymbol{v})$, i.e., detecting polygons (rooms) and their topology.
3. Searching the sub-graph $\boldsymbol{m}$ of *M* that is most similar to the desired graph $\boldsymbol{a}$. (Gold and Rangarajan 1996). The result includes a one-to-one map $f(i)$.
   i: the index of the *i*th node of $\boldsymbol{a}$ (desired topology)
   $f(i)$: the index of the corresponding node in $\boldsymbol{m}$ (subset of detected topology)
4. Examining whether the difference between $\boldsymbol{m}$ and $\boldsymbol{a}$ equals to 0 (function (4)). If yes, terminate the whole process; otherwise go back to step1 (restart).

$$T(M) = T(m(\boldsymbol{v})) = \sum_j^N \sum_i^N dif(i,j) \qquad (4)$$

$$dif(i,j) = \begin{cases} 0, if\ (m_{f(i),f(j)} \vee \neg a_{i,j}) = true \\ 1, \qquad\qquad\qquad\qquad otherwise \end{cases}$$

$a_{i,j}$: true or false, specifying the adjacency between two rooms in graph $\boldsymbol{a}$.
$m_{i,j}$: true or false, specifying the adjacency between two rooms in graph $\boldsymbol{m}$.

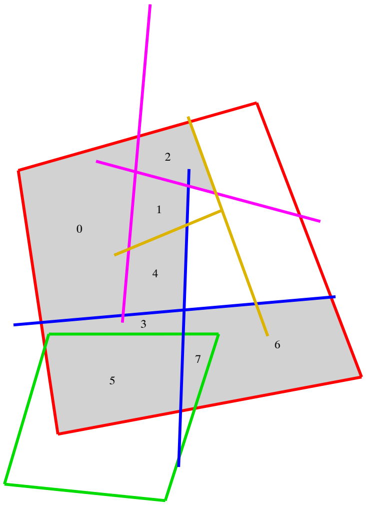


Fig 5.14: Selecting detected polygons as the rooms in the floor plan.

Measuring dimensional differences

After finding the set of polygons that satisfies the desired topology, the remaining task is making every polygon meet the dimensional requirements. In this program, the ideal shape of the room is a rectangle; thus, the cost function measures the difference between the polygon and the target rectangle:

$$G(\{r_i\}) = w_1 \sum_i^N tri(r_i) + w_2 \sum_i^N dar(r_i) + w_3 \sum_i^N ovl(r_i) \qquad (5)$$

$$tri(r_i) = \begin{cases} 0, if\ polygon\ r_i\ \ is\ a\ triangle \\ 1, \qquad\qquad\qquad\quad otherwise \end{cases}$$

$$dar(r_i) = \begin{cases} d, if\ d = |area(r_i) - area(R_i)| > A \\ 0, \qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

$$ovl(r_i) = \begin{cases} d, if\ d = H - maxOvl(r_i, R_i) > 0 \\ 0, \qquad\qquad\qquad\qquad\quad otherwise \end{cases}$$

A, H: thresholds.
R_i: the target rectangle
$area(r_i)$: the area of the polygon
$maxOvl(r_i, R_i)$: the maximum overlapping area between polygon r_i and the target rectangle R_i.

Calculating the maximum overlapping area is to find the position/orientation of the target rectangle that maximizes the overlapping area. The process is implemented by simulating annealing (2013):

```
$s \leftarrow s_0; e \leftarrow overlap(s); e_{max} \leftarrow e;\ k \leftarrow 0$
**while** k<kmax
   $slast \leftarrow s$
   $\beta \leftarrow \beta_0 T$
   $s \leftarrow neigh(\beta, s)$
   $e \leftarrow overlap(s)$
   **if** $e < e_{max}$       //new state is worse
       **if** $(T \geq T_0)$  $\alpha \leftarrow \exp\,((e - e_{max})/T)$
       **else** $\alpha \leftarrow 0$
       **if** $(random() < \alpha)$    $e_{max} \leftarrow e$
       **else**  $s \leftarrow slast$
   **else**                //new state is better
         $e_{max} \leftarrow e$
    **if** $(T \geq T_0)$
       $T \leftarrow T - \gamma$
    $k \leftarrow k + 1$
$maxOvl(r_i, R_i) \leftarrow e_{max}$
```

s: the position/orientation of the target rectangle R_i, $s = \{x, y, \theta\}$
$overlap(s)$: calculates the overlapping area between the polygon r_i and target rectangle R_i, using Sutherland Hodgeman algorithm (Toxiclibs 2013).
$neigh(\beta, s)$: finds a neighbor state of state s, according to the maximum distance β
β: maximum distance between two states.
$random()$: gets a random value between 0 and 1.

T: temperature, see (Simulated Annealing 2013)
T_0, β_0, γ: constants.

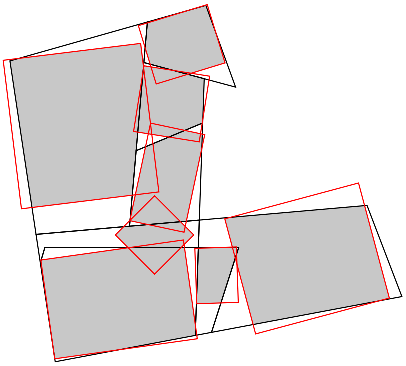


Fig. 5.15: The overlapping areas (in gray) between the polygons and their target rectangles (in red).

Minimizing dimensional differences

Simulating annealing is employed to minimize the cost function (5). The process is divided into two stages using the same framework:

```
$v \leftarrow v_0; e \leftarrow err(v); e_{min} \leftarrow e;\ k \leftarrow 0$
while $e_{min} < \delta$ or k<kmax
    $vlast \leftarrow v$
    $v \leftarrow neigh(sv)$
    $e \leftarrow err(v)$
    if $e > e_{min}$        //new state is worse
        if $(T \geq T_0)$  $\alpha \leftarrow \exp((e_{max} - e)/T)$
        else $\alpha \leftarrow 0$
        if $(random() < \alpha)$    $e_{min} \leftarrow e$
        else  $v \leftarrow vlast$
    else                    //new state is better
        $e_{min} \leftarrow e$
    if $(T \geq T_0)$
        $T \leftarrow T - \gamma$
    $k \leftarrow k + 1$
```

$\boldsymbol{v}$: the vector specifying the shape model, as the argument of cost function (5).
$err(v)$: equals to the first two terms of function (5) in the first stage; equals to function (5) in the second stage.
$neigh(s)$: finds a neighbor state of the current state *v*.
T: temperature.

T_0, γ, δ: constants

Results and discussion

All the methods are implemented and tested in Java. After satisfying the topological and dimensional requirements, there is a post-process. The first part of the process is, illustrating the edges of the rooms as solid walls (Fig 5.16); The second part is, making an opening (door) between two rooms if they need to be connected in graph ***a*** (Fig 5.11); The final part is, making an additional opening as the main entrance for the room that need to be accessible from outside (the room with a cross in Fig 5.11). Different specifications have been tested. Due to the space of the thesis, only four results for each specification are shown below.

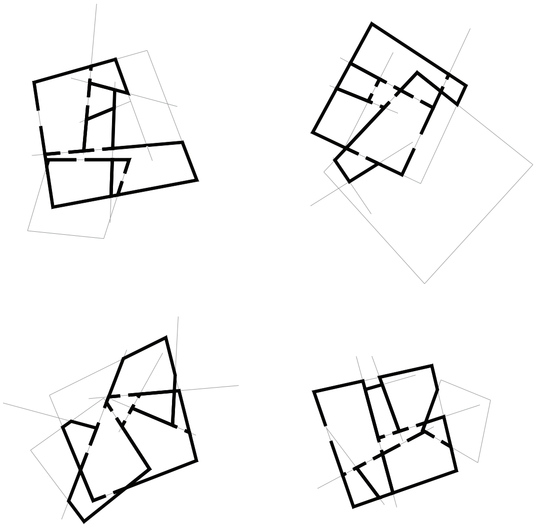


Fig.5.16: Four results of specification I (Fig. 5.11-5.12).

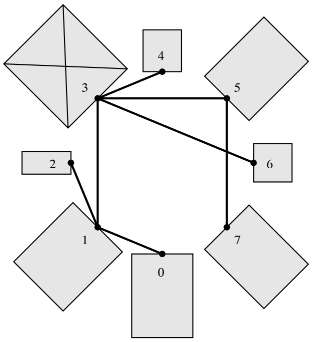


Fig 5.17: Room specification II

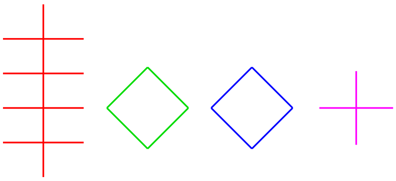


Fig 5.18: Shape specification II

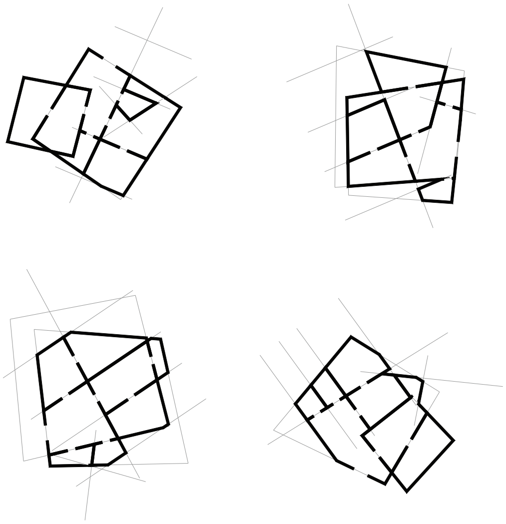


Fig 5.19: Four results of specification II (Fig 5.17-5.18).

The results imply that the topology model and the shape model can be correlated through computing. More precisely, if we have 1) a parametric model of the solid parts (usually as the walls) in the floor plan and 2) a detector that extracts rooms from the shape model, then the topology model and the shape model could be potentially correlated. In this program the correlation is implemented by optimization; however, it is important to realize that the correlation does not necessarily lead to optimization. Any procedure that can enable the two kinds of models to work together is a valid method of correlation.

5.2.3 Topology + Pixels

Graph, Pixels and blobs

This experiment combines the topology model and the tessellation model based on pixels. The pixels serve as the data structure of raster images. An area of similar color in the image constitutes a blob—a set of pixels of similar color. This experiment represents the rooms of floor plans with blobs. The topology between the rooms (blobs) is represented by graph. The program takes the image and the topological specification of rooms as inputs and then produces valid floor plans. The focus of the experiment is the interplay of the topology model (graph) and the model of pixels (image).

The image is an extremely flexible media that can record all kinds of spatial patterns. Thus, various patterns in the image could be used to shape the rooms of the floor plan. This program adopts a region detection algorithm (Nock and Nielsen 2004) that detects the regions of similar color in the image. In other words, the algorithm partitions the image into disjoint regions (blobs). Two tools are used to evaluate the detected blob (examining whether the shape of the blob is suitable for a room). The first tool is the depth analyzer [4] calculating the depth (shortest distance to the boundary of the blob) of pixels in a given blob (Fig 5.20). The depth values are used to calculate the center of the blob:

$$x = {\sum d_i x_i}/{\sum d_i} \qquad y = {\sum d_i y_i}/{\sum d_i}$$

x, y: the coordinates the center of the blob.
d_i: the depth of the ith pixel in the blob.
x_i, y_i: the coordinates of the ith pixel in the blob.

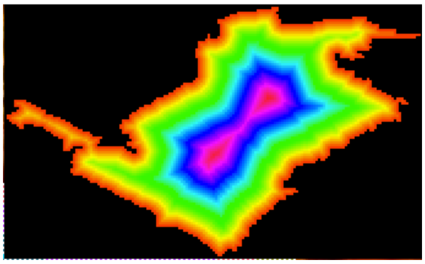


Fig 5.20: The depth analysis: the hue denotes the depth, namely the shortest distance to the boundary.

A circle with the same area as the corresponding room is placed at the center of the blob. The similarity between the shape of the blob and the circle approximates the likeness between the blob and the corresponding room. More precisely, the error of the shape of the blob (as a room) is defined as follows:

$$S_i = {a + b}/{t + s} \qquad (1)$$

S_i: the error of shape of the ith blob

---

[4] Developed by Benjamin Dillenburger at CAAD/ITA/D-ARCH, ETHZ.

a: the number of pixels that are inside the circle but outside the blob.
b: the number of pixels that are outside the circle but inside the blob.
t: the number of pixels inside the circle
s: the number of pixels inside the blob

To make the blob more flexible for accommodating rooms, each blob is divided into two parts. The two parts could server as either one room together or two rooms respectively. Fig 5.21 shows some possibilities of dividing a blob. The “best” one (the lower-middle) will be saved for the blob. The error of division is defined by:

$$p_1^2/a_1 + p_2^2/a_2$$

p_1: the perimeter of the first part of blob
a_1: the area of the first part of blob
p_2: the perimeter of the second part of blob
a_2: the area of the second part of blob

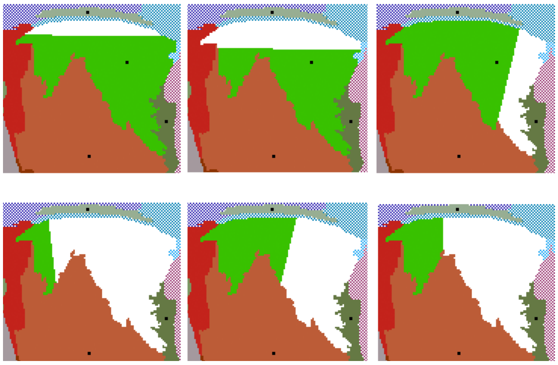
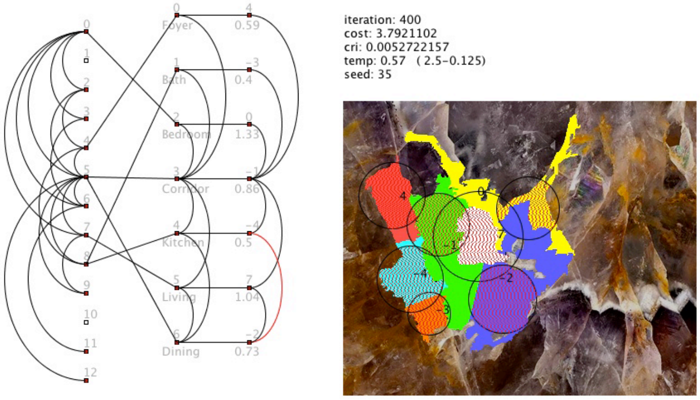


Fig 5.21: Variations of dividing the blob.

Synthesis based on the two models

The topological and dimensional requirements of the rooms are predefined. The task is to assign the rooms to the blobs detected from the image. There are two goals: first, the selected blobs should satisfy the topological requirements of the rooms. Two blobs are regarded as connected if they share edges or they are close enough to each other. Second, each selected blob should have a proper shape with respect to the corresponding room (the shape evaluation is discussed above). The diagram in Fig 5.22 shows the assignment of rooms. The first column of nodes denote all the blobs, the topological relationships between them are depicted by arcs. The second column of nodes represent the predefined rooms, the topology between them is also illustrated by arcs. The connections between the first two columns denote the assignment of rooms. One blob could connect two rooms, since one blob could be divided into two parts as discussed above.



```
iteration: 400
cost: 3.7921102
cri: 0.0052722157
temp: 0.57   ( 2.5-0.125)
seed: 35
```

Fig 5.22 Assigning rooms to blobs, iteration 400 (840 iterations in total).

A search algorithm is used to minimize the error of assignment. The error is defined as:

$$ {}^{w_1}\!/_{\sum a_i} + w_2 T(x) + w_3 \sum S_i $$

a_i: the area of blob i
$T(x)$: the sum of topological errors, with assignment x
S_i: the error of the shape of blob i
w_1, w_2, w_3: constant weights

Simulated annealing (Kirkpatrick et al. 1983; Aarts and Laarhoven 1989) is employed for searching the best assignment. The annealing scheme is defined as:

$$ p = \begin{cases} 1 & \mathit{if}\ e' < e \\ \exp{(e - e')}/_{t} & \mathit{else} \end{cases} $$

$$ t = ab^k $$

e: the error of current assignment
e': the error of new assignment
p: the probability of adopting the new assignment
t: temperature of simulated annealing
k: the number of iteration (the discrete time)

a, b: constant

The results

The algorithms are implemented and tested in Java. The program predefines the rooms as follows [5]:

0: Foyer (F) 10 m^2
1: Bathroom (W) 4.4 m^2
2: Bedroom (B) 9.3 m^2
3: Corridor (C) 9 m^2
4: Kitchen (K) 10 m^2
5: Living room (L) 19 m^2
6: Dining (D) 11 m^2

The topology of the rooms is defined as:
0-3 (Foyer connects Corridor)
1-3
2-3
3-4-5 (Corridor connects kitchen and living room)
4-6
5-6

The results imply that the two models can work together (the model of pixels specifies the shape/form of rooms; the graph model governs the topology of the rooms). Moreover, the interplay between the two models could produce novel compositions. The program is tested over the image of nature objects (Fig 5.23), of paintings (Fig 5.24-5.25) and of collage (Fig 5.26).

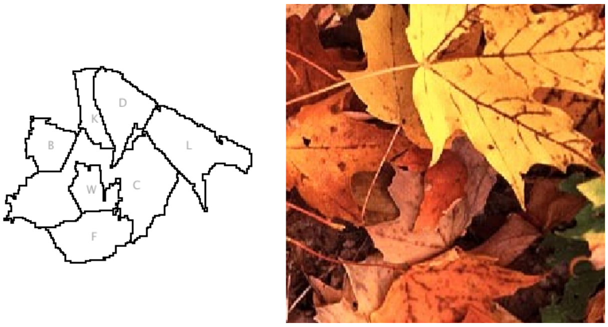


Fig 5.23: One solution generated from the image of natural objects.

---

[5] The set of data is just an example. It is not critical to the method of the program, since it serves as the inputs of the program.

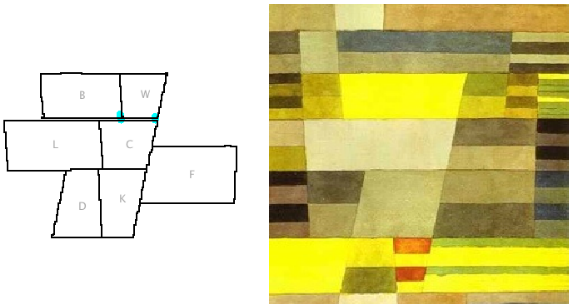


Fig 5.24: One solution generated from image of painting (Paul Klee).

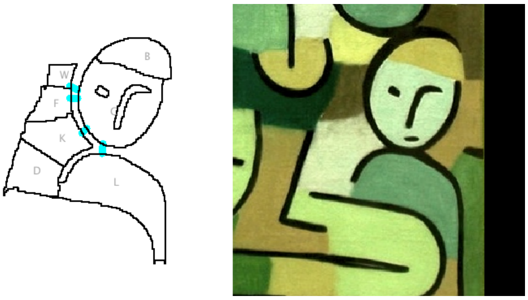


Fig 5.25: Another solution generated from image of painting (Paul Klee).

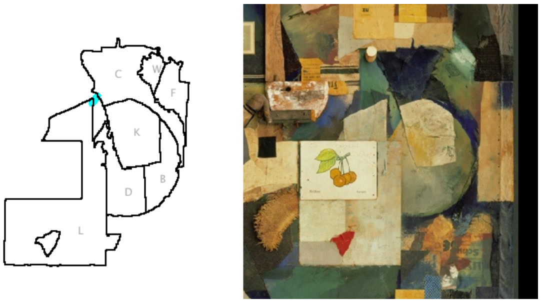


Fig 5.26: One solution generated from the image of collage (Kurt Schwitters).

5.3 Floor Plans + 3D Forms[6]

5.3.1 Floor plans and 3D forms of architecture

This project works with families of models instead of individual models. One family is made of the models of layout planning, the other for 3D forms. No shared rep-

---

[6] The content of this section uses the material from Hua (2013).

resentation between the two families is required. The key is the valid communication between the two families of models. To some degree, the nature of communications rather than the models defines the family of models. Each member from one family should work with any member from the other. In this project, any model of floor plans can work with any model of 3D forms, which leads to a rich variety of structures and forms. If there are *n* models in one family and *m* models in the other, there would be $n \times m$ combinations.

In terms of the Java programming language, the models of floor plan layouts extend an abstract class. The other family of models extend another abstract class. The Java program takes one particular model of layout planning and one particular model of 3D forms as two "parameters," then produces solutions of both floor plans and 3D forms. The algorithm consists of three stages:

1. Generating a particular 3D form via a particular model (one model can produce infinite number of 3D forms).
2. Constructing floor plan layouts via a particular model, under the constraints of the generated form in step1. This procedure could fail if the conflicts between the two models cannot be solved.
3. If step 2 succeeds, a valid solution will be produced. Otherwise, there is a return to step 1 (restart).

The two kinds of models are not strictly coupled with each other. The composition of the floor plan is still open when the 3D form is fixed. The other way round, there are many variations of 3D forms after the floor plan is made. In this project, a procedure of communication is defined as follows: first, the model of 3D form defines two types of volumes: the positive (to be occupied by buildings if possible) and the negative (not to be occupied if possible). Second, the whole volume produces a set of 2D planes by making horizontal slices of the volume. Every plane saves the information of the positive regions and the negative regions. The former are supposed to be occupied by rooms, the latter to be occupied by circulations (corridor, staircase and elevator) if necessary. Then the model of floor plan layouts can operate on these 2D planes.

5.3.2 The models of 3D forms

Any model that generates 3D geometry can be adopted by the family of 3D forms. This project builds two particular models: the "Cubes" and the "Perlin." The "Perlin" model creates two "iso-surfaces" that divide the whole volume into three parts. At first, the volume is voxelized and then Perlin's (2002) noise function gives every voxel a "density" value. The iso-surfaces (Fig 5.27 left) are constructed where the density value is equal to the predefined iso values. Thus, two values lead to three disjunct volumes. One of the three volumes is marked “negative”, the other two are “positive.”

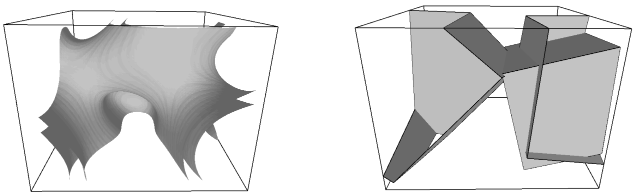


Fig 5.27: The 3d geometry generated by the “Perlin” model (left); the geometry generated by the “Cubes” model (right).

The 2D planes are created by cutting the whole volume at the level of floors (with equal intervals). The planes inherited the information of positive/negative regions in the volume. The light gray areas in Fig 5.28 denote the negative parts. Besides, the virtual floors at the level of 1m and 2m higher than each floor are also calculated. The dark gray areas denote the negative areas from the virtual floors.

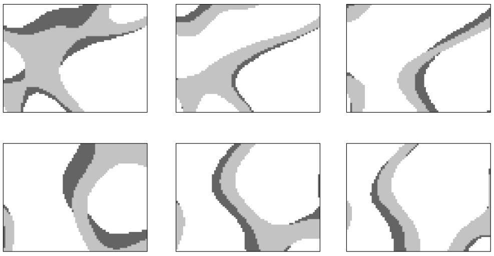


Fig 5.28 The 2D planes resulted from the “Perlin” model

The model of “Cubes” first generates three cubes and then the cubes are cut by the cuboid—the boundary of the whole volume (Fig 5.28 right). The whole volume of the cuboid is divided into voxels, the sign of each voxel is calculated by:

$$s = (-1)^n$$

Sign n denotes the number of cubes that contain the target voxel. According to the formula, the volume outside all cubes is positive. Actually, the volume is negative if it is inside an odd number of cubes. Otherwise, it is positive.

5.3.3 The models of floor plan layouts

The models of floor plan layouts manipulate the 2D planes resulted from the model of 3D forms. The positive areas on the plane are for rooms and circulation (corridor, staircase and elevator), while the negative areas are only for circulation if necessary. The “Central Corridor” model constructs a corridor along the long side of the floor. The staircase and the elevator must be connected to the corridor. Other spaces are further divided into small rectangular rooms that are all connected to the central corridor. Fig 5.29 shows a result under the condition that all the areas are positive.

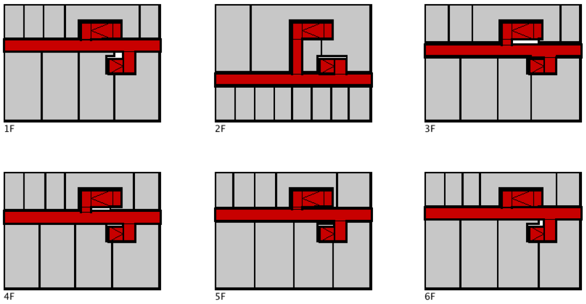


Fig 5.29: One layout generated by the “Central Corridor” model, when all the areas on the plane are positive.

However, the situations become more complicated when there are negative areas on the planes. To be precise, there are four “invalid” situations (Fig 5.30): a room is too small; a room is too narrow; a room is not connected to the central corridor; or the shape of a room is invalid for opening a door to the corridor. In order to avoid these situations, the algorithm merges the invalid room with its neighbors. If the room is still not valid after that, it will be eliminated or be merged as part of the corridor.

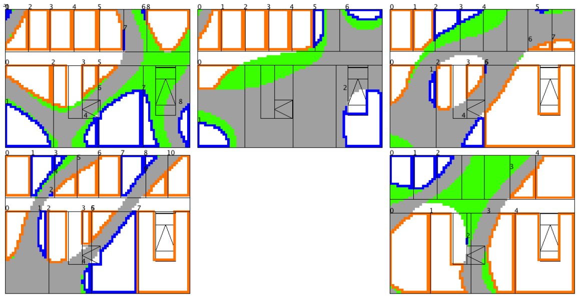
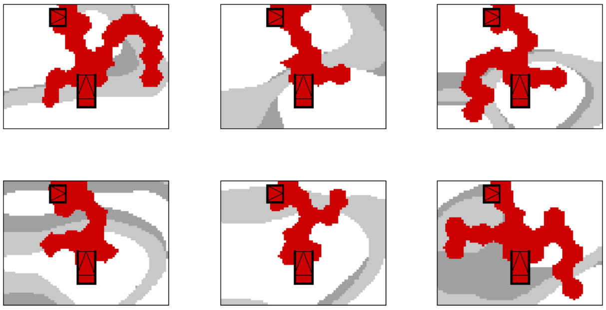


Fig 5.30: Analysis of the central corridor layouts.

The “Voronoi” model arranges the floor plan based on the Voronoi tessellation of the plane. It first generates the corridor with three goals: first, the corridor should connect the staircase and the elevator; second, the corridor should reach all the regions isolated by negative areas (Fig 5.31 (a)); third, the area of the corridor should not be very large. Then the algorithm subdivides the positive areas of the floor by grouping the cells in the Voronoi tessellation (Fig 5.31 (b)). Each room should have a relatively compact shape.


(a)

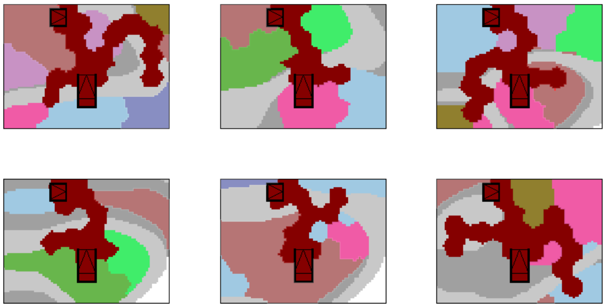

(b)

Fig 5.31: A corridor generated by the "Voronoi" model (a); Grouping cells into rooms (b).

5.3.4 Results and discussion

The models are constructed and tested in Java. The results (the 3D forms with floor plans) below are directly produced by the Java program[7] without tuning by hand. The results imply that each model from the family of 3D forms works fine with each model from the family of architectural layouts. Different pairs of the models make different articulations. Though there are only two members in each family, it is possible to add more members into each family according to the communication channel between the two families.

The first results came from the model of floor plan layouts working with none of the 3D forms. It confirms that a single model from one family can function alone. However, a combination between two families constitutes a novel "model" that is more productive. There are four different situations: Central Corridor - Cubes (Fig

---

[7] The 3D forms are exported from Java and then rendered in Maxwell

5.32), Voronoi - Cubes (Fig 5.33), Central Corridor - Perlin (Fig 5.34), and Voronoi - Perlin (Fig 5.35). Only one result from each situation is shown here, though there are infinite numbers of results in each situation.

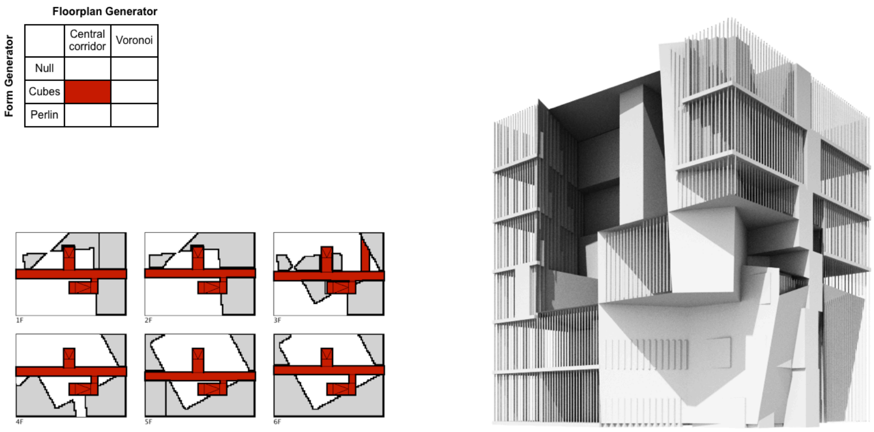


Fig 5.31: The layouts and the form generated by Central Corridor – Cubes

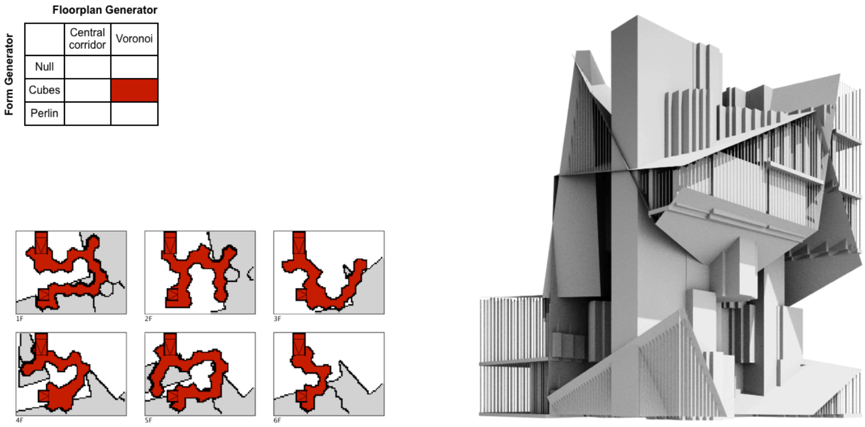
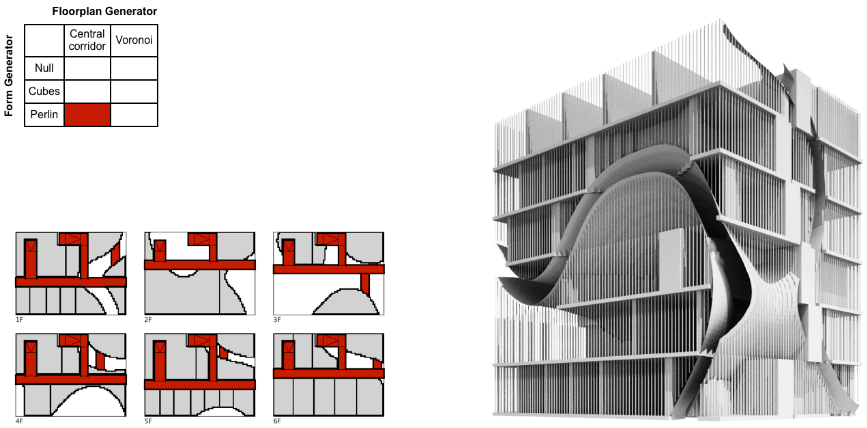
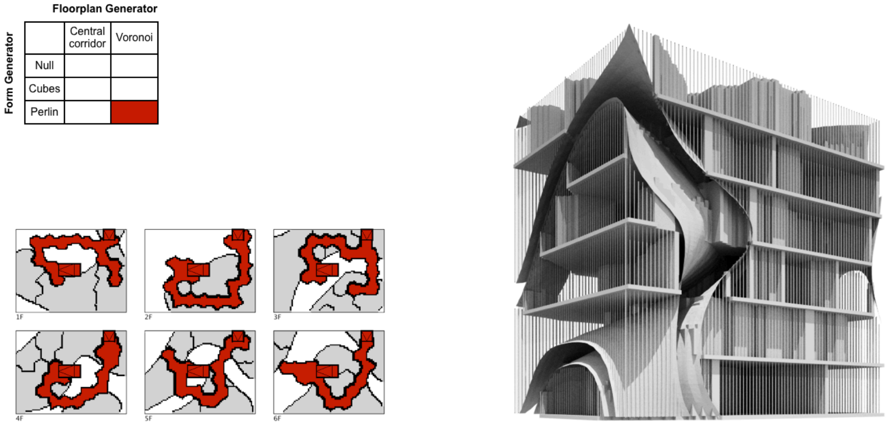


Fig 5.32: The layouts and the form generated by Voronoi – Cubes



	Central corridor	Voronoi
Null		
Cubes		
Perlin	🟥	

Fig 5.33: The layouts and the form generated by Central Corridor – Perlin



	Central corridor	Voronoi
Null		
Cubes		
Perlin		🟥

Fig 5.34: The layouts and the form generated by Voronoi – Perlin

The results imply that the interplay of the two families of models can produce a wide range of structures and shapes. Each pair of models constitutes a contingent generator. The generated forms can be read in two ways (corresponding to the two models) simultaneously. Though there are ambiguities in the forms, the two underlying generators are more or less transparent. It suggests that combining heterogeneous models in one generative process could be valid and productive.

5.4 Case-Based Reasoning and Fictional Aggregation

5.4.1 Case-based reasoning and domain knowledge

Case-based reasoning (CBR) or case-based design (CBD) is an important branch of intelligent systems. CBR usually requires an adequate representation of the cases, or, a formal language describing the experiences in the domain. Yet, representing architecture is a critical task. This experiment tries to combine the CBR model with a novel model of aggregation, in order to enhance the productivity of CBR. In contrast to the approaches that formulate as much domain knowledge as possible, this project suggests minimizing the domain knowledge in the case-based design.

Compared to many scientific fields and engineering fields, the domain of architectural design is less structured. There is no general model of architecture that is widely accepted. Because of that, different strategies have been developed for applying CBR to architecture. These strategies fall into two categories: retrieval-oriented and adaptation-oriented. The first strategy usually follows a conventional CBD procedure of indexing, retrieval, evaluation and revision/adaptation (Aamodt and Plaza 1994; Kolodner 1992). CADRE (Hua and Faltings 1993), Archie-II (Domeshek and Kolodner 1993), and SEED-layout (Flemming and Chien 1995) are some important experiments in architecture. Yet the second strategy utilizes cases without a standard CBD framework; rather, it needs additional methods to decompose and recombine cases into new designs. For example, Rosenman (2000) and de Silva Garza and Maher (2000) both used evolutionary algorithms to manipulate cases and to generate new instances. These programs prefer the task-specific models to the general models of architecture. Yet this experiment employs the elements of both strategies - combining a model for processing cases and a specific model for constructing new instances. It integrates the CBR with an aggregation model.

It is critical to apply domain knowledge to the cases in CBD. The raw cases usually contain unstructured data, since the low-level of information in these original cases is not adequate for retrieval or other operations that require high-level information. To fill such a semantic gap (Smeulders 2000), feature extraction or pattern recognition is often required. Hence, certain domain knowledge will impose a model on the unstructured data to make sense of them. During this process, the “data” becomes the “information.” This process could be carried out manually, or by algorithms of clustering, classification, segmentation, feature extraction, and pattern recognition. Some CBD theories (Maher and de Silva Garza 1997) hold that rich domain knowledge benefits the reasoning and the operations on the cases. In contrast to that, for this experiment we employ a minimal body of domain knowledge. Though this CBD model contains very limited domain knowledge, it could be productive with an aggregation model that specifies how the cases could be deconstructed and re-assembled into new designs.

5.4.2 Recognition model + aggregation model

The research takes the 3D mesh models of architecture as the sources of cases. The computer program is also supposed to produce new designs in 3D mesh models. Most CBD researches on architecture focused on the 2D drawings of architecture (Heylighen and Neuckermans 2001; Richter et al. 2007). However, 3D models could catch more information of the cases than 2D drawings. Some CAD systems focus on the geometries of architecture (e.g. AutoCAD, SketchUp and Rhinoceros), while some integrate the semantics of architecture (e.g. Revi and ArchiCAD). In our case, the geometrical models without architectural information are more versatile. For convenience, we choose the SketchUp models because Google/Trimble has provided an online repository of SketchUp models (3D Warehouse 2013). We pick the models that contain interior structures such as interior walls and stairs.

The program works fine with a very small repository of modernism architectures[8]. The 3D models are converted into Wavefront obj files by SketchUp. The geometries are triangulated so that the buildings are actually represented by a set of triangles without any additional semantic information. Then the geometries are loaded in the Java program. The program has two stages governed by two models:

1. Recognition. Categorizing the triangles of the geometry into several sets (wall, floor, stairs/slopes). Besides, the connectivity between the floors and stairs, between floors and walls are also calculated.
2. Aggregation. Constructing new designs by combining the elements extracted in the first step, according to user-defined topologies.

The goal of recognition is to categorize the triangles of the geometry into meaningful categories such as walls and floors. The categorization meets the requirements of the aggregation model. The recognition includes three major tasks: recognizing floors, stairs/slopes, and walls. The recognition of floors contains five steps:

1. Grouping all the triangles into four sets according to their normal vectors: $\mathrm{A}\{\mathrm{t}{:}\ n(t) < 0.1\}$, $\mathrm{B}\{\mathrm{t}{:}0.1 \le n(t) < 0.95\}$, $\mathrm{C}\{0.95 \le n(t) < 0.995\}$, $\mathrm{D}\{\mathrm{t}{:}0.995 \le n(t)\}$(horizontal pieces).
   t: triangle made of three Euclidean vectors (vertices).
   n(t): the absolute value of the z component of the normal vector of triangle t.
2. Making set E=$\{e\}$, $\bigcap_{e\in E} e = \emptyset$, $\bigcup_{e\in E} e = \bigcup_{t\in D} t$. If share($t_i$, t_j) is true, t_i and t_j must fall into the same set e.
   e: a set of triangles, actually a continuous face made of triangles.

---

[8] Twenty-six models are used in the tests below (Fig 5.38-5.41).

share(t_i, t_j): test if two triangles share two vertices.

3. Constructing set F=$\{f: f \in E, area(f) > a_1\}$.
   f: a set of triangles, actually a continuous face made of triangles.
   a_1: a threshold, constant.
   $area(f)$: the sum of the areas of the triangles in set f.
4. Calculating set G=$\{g\}$, $\bigcap_{g \in G} g = \emptyset$, $\bigcup_{g \in G} g = \bigcup_{f \in F} f$. If *close*($f_i, f_j$) is true, all the triangles in f_i and in f_j must fall into the same set g.
   close(f_i, f_j)=$\exists t_p, t_q$: closeZ(t_p, t_q) $\wedge$ closeXY(t_p, t_q), $t_p \in f_i$, $t_q \in f_j$.

   g: a set of triangles.
   closeZ(t_p, t_q): true if the minimum difference between the z components of the vertices of the two triangles are below a threshold; otherwise false.
   closeXY(t_p, t_q): true if the minimum distance between the vertices of the two triangles are below a threshold, or the two triangles overlap in the XY plane; otherwise false.
5. Making set H=$\{h: h \in G, area(h) > a_2\}$.
   h: a set of triangles, actually a floor.
   a_2: a threshold, constant.

The set H is the set of floors (Fig 5.36, top). The algorithm is designed for handling "bad meshes" (e.g. a single floor might be made of two unconnected surfaces) and modeling variations (e.g. a floor can be modeled either as a single surface or as two surfaces).

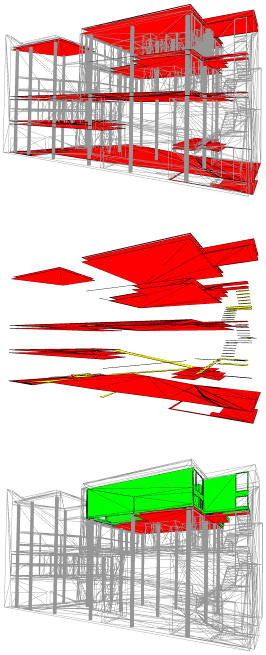


Fig 5.36: One result of recognition. top: floors in red; middle: floors connected by stairs/slopes (in yellow); bottom: walls (in green) with corresponding floors.

The stairs/slopes are the vertical components connecting a pair of floors (Fig 5.36 middle). Therefore, the stairs/slopes are searched between any pair of two floors. The search procedure for floor h_i and h_j is as follows:

1. Constructing set K={t: $t \notin \bigcup_{h\in H} h$, $t \in C \cup D$, $minz(t) > h_j$, $maxz(t) < h_i$}.
   minz(t): the minimum z coordinate of the triangle t.
   maxz(t): the maximum z coordinate of the triangle t.
   h_i: the upper floor.
   h_j: the lower floor.
2. Calculating set M={m}, $\bigcap_{m\in M} m = \emptyset$, $\bigcup_{m\in M} m = \bigcup_{t\in K} t$. If close($t_i, t_j$) is true, t_i and t_j must fall into the same set m.

   close(t_i, t_j)=$share(t_i, t_j) \vee closeZ(t_i, t_j) \vee closeXY(t_i, t_j)$

   m: a set of triangles.
3. Making set N={n: $n \in M$, area(n)>a_3}
   n: a set of triangles.
   a_3: a threshold, constant.
4. Making set S={s:$s \in N$, $\exists t$: $c(t, h_i) \wedge c(t, h_j)$, $t \in s$}.

   $c(t, h) = \exists t_i$: $closeZ(t, t_i) \wedge closeXY(t, t_i)$, $t_i \in h$

   $c(t, h)$: test if triangle t is close to floor h.
   s: a set of triangles

As a result, there is a set of stairs/slopes (set S) connecting two floors, i.e., it is possible that more than one staircase/slope connects two floors. Finally, a pair of floors and the corresponding set of stairs/slopes make an element p ={h_i, h_j, S} for the aggregation stage. All elements extracted from the input models institute the pool P={p}. Besides, the walls (Fig 5.36 bottom) are recognized after the recognition of floors and stairs/slopes. Each floor saves a reference to its own walls (if there are any). The details are not shown here since the walls are secondary structures in this experiment, compared with the floors and the stairs/slopes as the essential elements for aggregation.

The aggregation model combines the elements {p} extracted from cases into new compositions. It consists of two steps: first, specifying topology and second, implementing the topology by the elements of {p}. The topology of the 3D multi-story buildings is represented by a graph. Each floor is modeled as a node, the set of vertical connections (stairs or slopes) between a pair of floors as a link/edge (Fig 5.37). Thus, an element p is represented by two nodes and one link in the graph. Two jointed nodes (e.g. the node B, C in Fig 5.37 left) means that the two floors connect horizontally. Such graph encodes minimal domain knowledge for making valid compositions of multi-story buildings. To some degree, the construction of topology is arbitrary due to the purposes of designers, yet the three topologies are tested in the experiment: LINEAR, CIRCLE, and BRANCH (Fig 5.37).

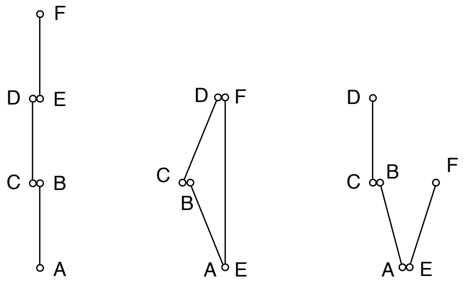


Fig 5.37: Three topologies: LINEAR, CIRCLE, and BRANCH (from left to right).

After the topology is fixed, the computer program randomly retrieves elements from {p} to implement the topology. For example, if we are going to put element p_i on the top of p_j to implement the LINEAR topology, the procedure is:

1. Translating all the geometries in p_i by vector v=up(p_j) - dn(p_i).
   up(p): the center point of the upper floor of element p.
   dn(p): the center point of the lower floor of element p.
2. Running a "trial and error" process until there is no error.

The error function:

$$f(v) = \text{overlap}(\text{v}, l(p_i), h(p_j)) \wedge \text{as}(\text{v}, \text{s}(p_j), \text{l}(p_i)) \wedge \text{as}(\text{v}, \text{s}(p_j), \text{h}(p_i))$$

$l(p)$: the lower floor of element p
$h(p)$: the upper floor of element p
$s(p)$: the set of stairs/slopes of element p
v: the 2d transformation vector of three components: angle (rotation), x and y (translation).

Actually the error function contains two sub-error functions, $overlap(v, h_i, h_j)$ tests if floor h_j overlaps floor h_i (translated by v); $as(v, s, h)$ tests if stairs s keeps away from the obstacle from floor h (translated by v).

$$overlap(v, h_i, h_j) = \exists t_p, t_q: overlap(t_p, t_q), t_p \in h_i, t_q \in h_j$$

$$as(v, s, h) = \nexists t_p, t_q: intersect(t_p, t_q), t_p \in s, t_q \in h$$

$intersect(t_p, t_q)$: true if triangle t_p intersects with the volume made by extruding (up) triangle t_q by 1.8m; otherwise false.

As we can see here, the major task of aggregation is error elimination. The errors occur in the vertical/horizontal connectivity between the floors, and in the collisions between the stairs/slopes and floors. The error functions for different topologies (Fig 5.37) are slightly different.

5.4.3 Results and discussion

The three topologies in Fig 5.37 can be quickly achieved by the Java program (usually after dozens of iterations of "trial and error"). The program outputs 3D mesh models in Autodesk .dxf files. The information calculated in the recognition phase enables the aggregation process to make valid compositions. Here the topologies for aggregation are manually defined; however, it is possible to generate the topologies under certain constraints by codes in the future work. Fig 5.38-5.40 shows two results for each topology. The results imply that a small collection of samples (26 cases) can lead to a great diversity of new compositions.

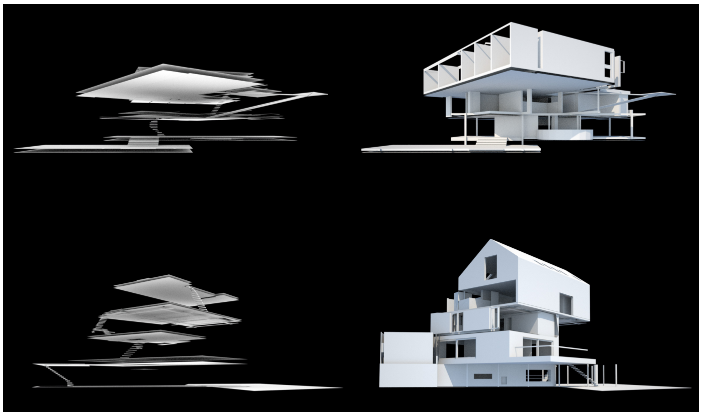


Fig 5.38: Two results for the LINEAR topology. For each result, both the skeleton (left, without walls) and the building (right, with walls) are shown.

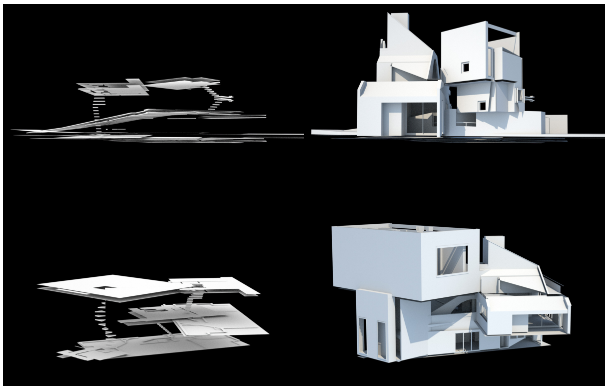


Fig 5.39: Two results for the CIRCLE topology.

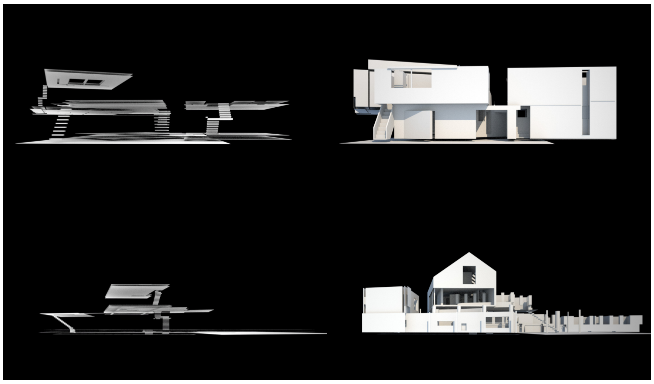


Fig 5.40: Two results for the BRANCH topology.

The program takes building models as inputs and produces new building designs. Besides, it's interesting to make a composition of the generated designs. A simple scenario was tested: first, creating several roads in a virtual 360m×360m construction site (the roads divide the site into parcels); second, employing a 2-d packing algorithm to place the generated buildings (according to their convex hulls) in the parcels in compact manner (Fig 5.41). Although very few criteria of urban design are considered here, more criteria could be added to the packing algorithm to have more control on the positions and the orientation of the buildings in the future work.

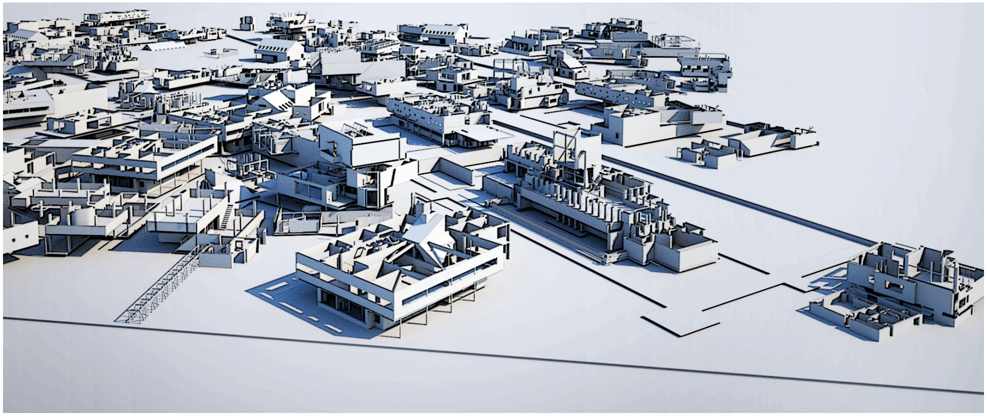


Fig 5.41: Packing the generated buildings in the parcels of a virtual construction site.

The results indicate that the combination of recognition model and aggregation model successfully implements the general task of CBD: “creating a new design solution by combining and/or adapting previous design solution(s)” (Watson and Perera 1997). The modification of either model would lead to different results. Minimal domain knowledge instead of general domain knowledge is coded in this program. Thus, there are some “flaws” in the results. For example, some generated instances lack structural feasibility since this aspect has not been taken into account. Moreover, some interior spaces are chaotic due to the conflicts between the walls from distinct elements. The future work can be developed in several directions. First, the topology for aggregation can be generated automatically. Second, the recognition-aggregation process could consider more criteria. Third, it is interesting to apply this CBD method (mainly focusing on individual buildings) to urban design.

Chapter 6

Conclusions

When a professional programmer types a few lines of codes and presses the “Run” button, the programmer hopes the codes are correct. However, when an architect presses the “Run” button, the architect is expecting something novel unfolding from the codes. It reflects the difference between modeling and the fictional use of models. The former follows specifications, while the latter involves imaginations. Yet, this thesis has shown that fictional manipulation of models actually serves as an alternative solution to the various problems in modeling.

As mentioned in the introduction, the nature of models leads to the coexistence of multiple inconsistent models. In other words, there is no “correct” model for any subject matter[1], and there is no best model for a given task[2]. Therefore, some people suggested choosing the useful or the cost-effective one[3] for the given task. Besides traditional modeling approaches, metamodeling tries to make abstraction of particular models in order to overcome the inconsistency of diverse models. Unfortunately, metamodels are still models and subsequently cannot solve the intrinsic problems of models. Thus, this thesis proposes an alternative of modeling—combined modeling. First, it is operationally valid. Second, it could be productive for architectural design. The proposal is not only applicable for computational design, but also has a few implications on design and planning in general.

6.1 Design with Models: A Point of View

Models are pervasive today. Thus, the ways that people choose and use models become critical. The thesis has shown that there are dilemmas in modeling, especially the problem of multiple inconsistent models. Many people are not aware of the dilemmas and their consequences, though they are familiar with certain particular models. For instance, most computer users know the RGB, CMYK, and HSB

---

[1] “Essentially, all models are wrong” (Box and Draper 1987, 424)

[2] “There is no such thing as the best model for a given phenomenon. The pragmatic criterion of usefulness often allows the existence of two or more models for the same event, but serving distinct purposes.” (Pinsky and Karlin 2011,1)

[3] Pinsky and Karlin (2011, 1) stated that: “In the final analysis, a model is judged using a single, quite pragmatic, factor, the model's usefulness”. Rothenberg (1989) articulated that: “modeling in its broadest sense is the cost-effective use of something in place of something else”.

model of colors. Yet, shall we ask which model is the most faithful representation of the colors or which model is the best? Unfortunately, many studies have found that such questions will not end up with a definite answer; rather, different models of the same subject matter will coexist. Moreover, the model sometimes does not strictly follow its subject matter, but has to meet the purpose of modeler and to fit the task environment. For example, suppose we are going to prepare some images that will appear in color in electronic files and will be printed in black and white. How do we make the color image vivid and the corresponding black/white image intelligible? It is not a bad idea to take account of both the color model and the grayscale model, so that the two models can take care of the image quality in the two media respectively. This case indicates that the task environment finally determines the values and the roles of the models. Furthermore, we often need multiple models instead of a single model in a task. Dealing with multiple models could be painful; however, it could also be interesting and productive. If people hold a single coherent view on a certain subject matter[4], then multiple models are quite disturbing. When people try to orchestrate a set of competing views, multiple models become the base of a new position. We can term the latter approach with combined modeling.

Few studies have paid attention to the fictional use of multiple models; however, the relevant situation is common in everyday life. For instance, when two people argue with each other, the argument will be justified (or refuted) several times by different sets of facts with different kinds of reasoning. In other words, different models are contingently fabricated for argumentation. Moreover, it is interesting that the subject matter may shift many times while the conversation keeps going. It is the activity of inventing new subjects based on previous subjects. It seems that creating contingent (verbal) models and making fictional combination of them to argue is a normal skill of everyday conversation (argumentation). The status of such activity, not their role, is analogous to design activity. As Rittel (1988) gave his opinion on the reasoning of designers: “A design problem keeps changing while it is treated…He finds himself in a field of positions with competing arguments which he must assess in order to assume his own position.”

This work has demonstrated that the fictional combination of multiple models could be fruitful in architectural design. Design needs multiple models because a single coherent model can hardly cover all issues in a given task. Besides, new issues might be introduced when the design proceeds. Design leads to the fiction of models when the designer uses the models to describe new objects that are not subject to the epistemic spaces of the individual models. Thus, we can draw some con-

---

[4] For instance, Dave and Woodbury (1990) said that: “We feel that it is important to present a single coherent view, and we thus have to embrace a single overall model.”

clusions in this context: Design uses models, but design is not modeling; Fictional combination of models involves multiple models, but it is not a model; Architectural design could be carried out by such combined modeling.

As shown in the introduction, the hypothesis of the work is that the coexistence of multiple inconsistent models is useful for architectural design. Through the investigation on modeling, computing, and design problems, we find that combined modeling is better than modeling for design. Design would inevitably suffer from the dilemmas of modeling if the design involves modeling. The method of combined modeling is a special response (not a final solution) to the dilemmas; it could be fruitful in the context of architectural design. The fiction of models makes little sense for well-defined problems, but it is quite helpful for architectural design whose subjects and objectives are neither fixed nor well defined.

6.2 Implications for Computational Design

Computational design is highly associated with modeling and computing. There have been many subfields[5] in computational design; they roughly fall into two categories from a point of view of modeling. The first is "one model for all". It anchors the model to a fixed subject matter. Hence, the more general the model is, the better the model is. Optimization models and the Building Information Modeling (BIM) pertain to this category. The second is "one model for one task." The subject matter varies with the task, so the resulting model is only good for "this" task instead of for any other. Usually, parametric design and generative design fall into this category, since the designers often build idiosyncratic models for particular tasks. The former is problematic due to the problem of multiple models[6]. Yet in the latter the models cannot be effectively accumulated since the modeler always has to make new model (or modify an available model) for a new task. Nonetheless, the two approaches are not substantially different within the framework of combined modeling. No matter whether a model is general or particular, it serves as a part of the fictional combination of models. The particularity of tasks would be embodied in the interrelationships between the models. Combined modeling does not seek the generality as with the "one model for all" approach, and it is better at reusing models than the "one model for one task" approach.

It is rewarding to revisit some methodologies in computational design from the viewpoint of fiction. We can identify the fictional part of some standard methodologies. For instance, in optimization programs, the objective function (or cost func-

---

[5] Parametric design, generative design, case-based design (or expert system), CAD systems, BIM, digital fabrication, interactive design and so forth.

[6] Essentially, there would be more than one model for the same original, due to the nature of models.

tion, fitness function) is more or less fictional since it is determined by the designer's purposes. The fictional character of the objective function is most obvious in the multiobjective optimization that combines several objective functions in a form of:

$$C(x) = \sum w_i C_i(x)$$

Such objective function is a linear combination of several objective functions. Each $C_i(x)$ takes account of distinct issues, and its "influence" on the overall objective function is weighted by value w_i. In fact, it is unreasonable that the trade-off between distinct models (objective functions) is reduced into a linear combination. Thus, the linear combination is fictional, and is actually one of the simplest version among enormous combinatorial schemes. We can see that optimization programs combine the output of the models (objective functions) rather than combine the models themselves. By contrast, combined modeling quests for the direct communications between different models. It allows the individual models to exhibit their own behaviors, and the designer can re-contextualize each model through its connections to other models. The optimization programs could be regarded as a special case of combined modeling, for they combine models in one dimension[7] while there are naturally many other dimensions for combination.

Many architectural designs involve a sort of digital chain. For instance, the architect employs an idiosyncratic model in the early stage of design, and then the result is passed to the structural engineer who applies a structural model to it in order to make it constructible, and the manufacture companies apply their own models to it so that they can materialize the building. Roughly put, the models are successively combined during the design process. Some information of the model could be lost or misunderstood when it enters another design stage in that the participants use other models. To solve this problem, people have suggested making compatible models or building a general model of models (e.g. BIM). However, both methods are quite rigid and very difficult to implement in reality. From a point of view of combined modeling, the models in one project don't need to be completely consistent because the participants just want their models to address (or solve) the problems adequately and not to be substantially interrupted by other models. In other words, combining models is not eliminating inconsistency, but making the inconsistency trivial and unharmful. Furthermore, the inconsistency could be meaningful in design. For instance, Venturi (1966) stated, "Through unconventional organization of conventional parts he is able to create new meanings within the whole." He encouraged architects to make use of the contradictions of architectural

---

[7] The output value of the objective function (or cost function) that is usually a weighted sum of several objective functions.

objects, yet the fictional combination of models can make use of the great diversity of architecture models.

6.3 Fictions Powered with Models and Computers

The work is based on the contemporary background that more and more models are available for architectural design. Yet, there have been confusions and difficulties in dealing with many models. There are two common strategies: First, constructing a particular model (or modifying an available one) for the given task. Second, developing a general model (or language) as a high-level abstraction of various models involved in the task. The first is modeling while the second is metamodeling (making model of models). However, both methods can't benefit a lot from the increasing number of available models, because the strategy of absorbing new models is absent (especially when the new models don't fit the ontology of the current models). Thus, this thesis proposes the method of combining models. It leads to the fiction of models.

People make fiction because sometimes they are better than "reality." As Ankeny (2009) put it: "Hence from fictions we can learn about and achieve a deeper understanding of the actual world around us." The fiction of models as a whole is not real, but it is made of the genuine models of reality. On one hand, the fiction still benefits from the models; on the other hand, it can convey information that contradicts the models.

One merit of model is that it packs the abstraction of the subject matter into an intelligible form (e.g. logics or mathematics). The model makes the idea on the subject matter explicit and exact. In the Notes on the Synthesis of Forms, Alexander (1964) developed a constructive diagram, a network of various requirements and possible forms, in order to solve complex design issues. The fiction of models also spans a network of design issues, but it has several distinctions from Alexander's diagram. First, the constructive diagram itself is a model, while the fiction combines models. Second, the diagram studies (the requirements and the form of) the artifact, while the fiction studies the models and the relationships between them. Here, it is not difficult to distinguish two mindsets on design: 1) The design world (the objectives, the constraints, the possible forms, the causal links and so on) can be modeled for a given task. Therefore, a systemic procedure searching for the optimal state(s) is feasible; 2) The design issues are essentially contingent and incoherent. Thus, the designer has to manage various solutions with multiple models and thus the concept of design will be embodied in the combination of models.

By running the models, the fiction of models becomes “real” or observable. As Godfrey-Smith (2009) spoke about fictions in science: “The world of a novel is something that does not actually exist, but would be concrete if real.” The designer can test, evaluate, and modify his fiction of models by running it in computers. In this context, the “fiction” has a twofold character: it is imaginary (not real) but it could be realized in the real world (or in a computer). The computer plays an indispensable role in the fictional use of models. Although mathematical models (or other formal models) are open to formal reasoning, their actual behaviors cannot be observed without running the models. Six decades ago, Turing (1952) developed the mathematical equations of morphogenesis, however, he conceded that it is impossible to get concrete results without the aid of computer and “one only gets results for particular cases.”[8] The same to the fiction of models, we have to run the models and to see how they actually work. Very often computing leads to imaginable but more or less unpredictable results. That makes the fiction of models especially interesting.

The architects are free to use all kinds of available models, or to construct new models by themselves. However, there is much more freedom in combining models. We call such combination fiction, since they refer to nothing in the real world and they may contradict with models. It is in contrast with scientific modeling that looks for the “objective” correspondence between the representation and the target phenomena. This thesis suggests that the architects should look for their own way of using models. Powered with models and computers, the modeling approaches of architects could be vivid and productive.

---

[8] Turing (1952) wrote, “One would like to be able to follow this more general process mathematically also. The difficulties are, however, such that one cannot hope to have any very embracing theory of such processes, beyond the statement of the equations. It might be possible, however, to treat a few particular cases in detail with the aid of a digital computer… The essential disadvantage of the method is that one only gets results for particular cases.”

References

3D Warehouse, 25 Match, 2013, http://sketchup.google.com/3dwarehouse/?hl=en&hl=en.

Aamodt, A. and Plaza, E.: 1994, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, in AI Communications. IOS Press, Vol. 7: 1, pp.39-59.

Aarts, E. H. L. and Laarhoven, P. J. M.: 1989, Simulated: An introduction. Statistica Neerlandica 43(1) 1989 31-52.

Ackoff, R. L., Gupta, S. K. and Minas, J. S.: 1962, Scientific method: optimizing applied research decisions, Wiley, New York.

Adami, C., Ofria, C. and Collier T.C.: 2000, Evolution of biological complexity, in PNAS April 25, 2000, vol. 97 no.9 p4463-4468

Adleman, L. M.: 1994, Molecular Computation of Solutions to combinatorial problem. in Science, New Series, Vol. 266, No. 5187(Nov. 11, 1994), pp.1021-1024.

Aguilar, R. J.: 1968, A generalized linear model fro optimization of architectural planning

Alberti, L. B.: 1443-52, On the art of building. in: H. F. Mallgrave (ed) 2006, Architectural theory, Blackwell publishing. pp.30-35

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel., S.:1977, A Pattern Language. Oxford University Press, New York.

Aires Mateus 2002-2011, 2011, EL Croquis 154 Alberti, L. B.: 1755(1965), Ten books on Architecture. translated by James Leoni, UK Alexander, C.:1964, Notes on the Synthesis of Form. Harvard University Press.

Alfaris, A.: 2009, Emergence through conflict : the Multi-Disciplinary Design System (MDDS), Ph.D Thesis, MIT.

Aliaga, D.G.: 2008, Interactive Example-Based Urban Layout Synthesis

Amidon, E. L., Elsner, G. H.: 1968, Delineating landscape view areas: a computer approach, Forest Research Note PSW-180, US Department of Agriculture, Washington, DC

Angélil, M. 2008, Inchoate: an experiment in architectural education. ETH Zurich. Aranda, B and Lasch, C.: 2006, Tooling. Princeton Architectural Press. New York.

Ankeny, R. A.: 2009, Model Organisms as Fictions. in: M. Suarez (ed.) Fictions in science: philosophical essays on modeling and idealization, Routledge, New York. pp.193-204.

Aristotle, 350 BC, Metaphysics, translated by W.D. Ross. http://classics.mit.edu/Aristotle/metaphysics.1.i.html, 04.07.2012

Armour, G. C. and Buffa, E. S. 1963, A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities

Arvin, S. A. and House, D. H.:1999, Making designs come alive: using physically based modeling techniques in space layout planning, in: G. Augenbroe, C. Eastman Eds.., Computers in Building: Proceedings of CAAD Futures ’99, Kluwer Academic Publishing, Boston, 1999, pp. 245–262.

Arvin, S. A.: 2002, Modeling architectural design objectives in physically based space planning. in Automation in Construction 11(2002) p213-225.

Ashby, W. R.: 1957, An Introduction to Cybernetics, Chapman & Hall, London.

Ashby, W. R.: 1962, Principles of the self-organizing system, in Principles of Self-Organization: Transactions of the University of Illinois Symposium, H. Von Foerster and G. W. Zopf, Jr. (eds.), Pergamon Press: London, UK, pp. 255-278.

Augenbroe, G.: 2005, A framework for rational building performance dialogues, in: B. Kolarevic and A. Malkawi (eds.) 2005, Performative Architecture: Beyond Instrumentality, Routledge, New York. pp. 97-110.

Azhar, S., Hein, M. and Sketo, B.: 2008, Building Information Modeling (BIM): Benefits, Risks and Challenges, Leadership Manage. Eng., 11(3) 241–252.

Balci, O.: 1986, Credibility assessment of simulation results: the state of the art, Proceeding of the conference on simulation methodology and validation to be held as part of the 1987 Eastern Simulation conferences.

Barberousse, A. and Ludwig, P.: 2009, Models as Fictions. in: M. Suarez (ed.) Fictions in science: philosophical essays on modeling and idealization, Routledge, New York. pp.56-73.

Barros, L., Valente, A. and Benjamins, R.: 1996, Modeling planning tasks. In Third In- ternational Conference on Artificial Intelligence Planning Systems, AIPS-96, AAAI, 1996, pp.11-18.

Bäck, T.: 1996, Evolutionary algorithms in theory and practice. Oxford, USA Badiou, A.: 1990, Number and Numbers, polity press, UK. Balkin, J. M.: 2009, Constitutional crises. 157 University of Pennsylvania Law Review 707 (2009).

Belkin, N. J., Oddy, R. N. and Brooks, H. M.: 1982, ASK FOR INFORMATION RETRIEVAL: PART I. BACKGROUND AND THEORY, Journal of Documentation, 38 (2) 61 – 71.

Bender, E. A.: 1978, An introduction to mathematical modeling, John Wiley & Sons.

Bengio, Y.: 2009, Learning Deep Architectures for AI, Foundation and trends in machine learning, 2(1), January 2009, 1-127.

Bentley, P.: 1999, An Introduction to Evolutionary Design by computers, in Evolutionary Design by Computers, P. Bentley (ed.), 1999, Morgan Kaufmann, California. pp.1-73.

Bentley P. J. and Corne D. W.: 2002: An introduction to creative evolutionary system. in D.W. Corne and P. J. Bentley (eds.) 2002, Creative evolutionary systems, Elsevier, pp.1- 75.

Bézivin, J.: 2005, On the unification Power of models, Software and Systems Modeling, 4(2) May 2005, 171 - 188.

Bickle, J. 2006, Multiple Realizability. in Encyclopedia of cognitive science, 2006 - Wiley Online Library.

Björk, B.: 1992, A conceptual model of spaces, space boundaries and enclosing structures, Automation in construction, 1(3) December 1992,193-214.

Blaikie, N.: 2010, Designing social research, Polity Press, Cambridge, UK.
Block, H. D.: 1962, The Perceptron: a model for brain Functioning. I, Review of modern physics, 34 (1) Jan 1962, pp.123-135.

Boden M.A.: 1990, The creative mind: myths and mechanisms. Routledge, New York.

Bokeloh, M. 2010, A Connection between Partial Symmetry and Inverse Procedural Modeling

Boole, G.: 1854 (2005), An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities. Project Gutenberg, www.gutenberg.net. 24.12.2012

Boullée, B. L.: 1794(1976), Architecture, essay on art, translated by Sheila de Vallée , in Boullée & Visionary Architecture, Helen Rosenau, New York: Harmony Books, 1976

Box, G. E. P. and Draper, N. R.: 1987, Empirical model-building and response surfaces. John Wiley & Sons, New York.

Broadbent, G.: 1973, Design in architecture. John Wiley & Sons Ltd. England.

Brotchie, J. F. and Linzey, M. P. T.: 1971, A Model for Integrated Building Design. in Build. Sci. 6, 89-96. .

Bühlmann, V.: 2012, Continuing the Dedekind Legacy today: Some ideas concerning architectonic computation.

Buchanan, R.: 1992, Wicked problems in Design Thinking, in Design Issues, Vol. 8, No.2, MIT press, p. 5-21.

Burry, J. and Burry, M.: 2010, The new Mathematics of Architecture. Thames&Hudson. New York.

Caldas, L. G.: 2001, An evolution-based generative design system: using adaptation to shape architectural form. Phd dissertation, MIT.

Camazine, S., Deneubourg, J., Franks, H. R., Sneyd, J., Theraula, G. and Bonabeau, E.: 2001, Self-Organization in Biological Systems, Princeton University Press, Princeton, NJ.

Carrara G., Kalay, Y. E. and Novembri, G.: 1992, Multi-modal representation of design knowledge. Automation in construction 1 (1992) pp.111-121.

Carrara G., Kalay, Y. E. and Novembri, G.: 1994, Knowledge-based computational support for architectural design, in: G. Carrara and Y. E. Kalay (eds.) Knowledge-based computer-aided architectural design, Elsevier Science, The Netherlands. pp.147-201.

Carson II, J. S.: 2004, Introduction of modeling and simulatin, WSC'04 Proceedings of the 36th conference on Winter simulation, 9-16.

Cartwright, N.: 1983, How the laws of physics lie, Oxford University press, New York, US.

Cerovsek, T.: 2011, A review and outlook for a 'Building Information Model' (BIM): A multi-standpoint framework for technological development, Advanced Engineering Informatics, 25(2), 224-244.

Chandrasekaran, B.: 1990, Design Problem Solving: a task analsysi. in AI Magazine Volume 11 Number 4 (1990) p.59-71.

Chase, S. C.: 2005, Generative design tools for novice designers: Issues for selection, Automation in Construction 14, 689-698.

Chiou, S.C.: 1995, The grammar of Taiwanese traditional vernacular dwellings. Chomsky, N (1957). Syntactic Structures.

Chouchoulas, O.: 2003, Shape Evolution, PhD thesis, University of Bath.

Cotes, P. and Makris, D.: 1999, Genetic Programming and Spatial Morphogenesis. AISB Symposium on Creative Evolutionary Systems, Edinburgh College of Art and Division of Informatics, University of Edinburgh. pp.105-114.

Coates, P., Healy, N., Lamb, C. and Voon W. L..: 1996 The use of cellular automata to explore bottom – up architectonic rules,Paper presented at Euro graphics UK Chapter 14th Annual Conference held at Imperial College London

Colajinmi, B., de Grassi, M., di Manzo, M. and Naticchia, B.: 1991, Can planning be a research paradigm in architectural design? in Gero, J. S.(eds). Artificial intelligence in design '91, p.23-48.

Cook, V. J.: 2007, Chomsky's universal grammar: an introduction Corbusier, L. 1931(1985), Towards a new architecture. Dover publications.

Cortés, J. A.: 2011, Building the Mould of Space. Concept and Experience of Space in the Architecture of Francisco and Manuel Aires Mateus, in Aires Mateus 2002-2011, 2011, EL Croquis 154.

Coyne, R.: 2005, Wicked problems revisited, in Design Studies 26 (2005), Elsevier. Great Britain. p.5-17. Darwin, C.: 1859, The origin of species, D. Appleton and company, New York.

Cross, M. C.: 1993, Pattern formation outside of equilibrium, Reviews of Modern Physics, Vol. 65, No. 3, July 1993, 851–1112

Da Costa, N., and Steven F.: 2000, Models, Theories, and Structures: Thirty Years On, Philosophy of Science 67, Supplement, S116–127.

Dave, B. and Woodbury, R.: 1990, Computer Modeling: A first course in design computing. in M. McCullough, W. J. Mitchell and P. Purcell (eds.), 1990, The Electronic Design Studio, MIT Press, London. pp.61-p76

Dawkins, R.: 1976/1989, The Selfish Gene. Oxford. Great Britain.

Dawkins, R.: 1986, The Blind Watchmaker. Longman, Harlow.

Deleuze, G.: 1968(1994), Difference and Repetition. Translated by Paul Patton, Athlone Press, Great Britain.

Deleuze, G. and Guattari, F.:1980, A Thousand Plateaus. Trans. Massumi, B. University of Minnestota Press. London and New York.

de l'Orme. P, 1626, Architecture de Philibert de l'Orme, Paris.

Dedekind, R.: 1901, Essays on the theory of numbers, the open court publishing company, Kegan Paul, Trench, Trubner & Co., Ltd.

Dembski, W. A.: 1999, Intelligent Design: the bridge between science & theology. InterVarsity Press.

Dennett, D. C.: 1995, Darwin's dangerous idea: evolution and the meaning of life. Penguin, England

de Silva Garza, A. G. and Maher, M. L.: 2000, A process model for evolutionary design case adaptation, in Gero, J. S. (ed.), Artificial Intelligence in Design '00, Kluwer Academic, Dordrecht. p. 393-412.

de Silva Garza, A.G. and Maher, M. L. 2001, Using evolutionary methods for design case adaptation. Proceedings of the Twenty First Annual Conference of the Association for Computer-Aided Design in Architecture / Buffalo (New York) 11-14 October, 2001, pp.180-191

De Wolf, T. and Holvoet, T.: 2005, Emergence Versus Self-organisation: Different conceptes but promising when combines. in: Engineering Self-organising systems, Brueckner, Sven A.; Di Marzo Serugendo, Giovanni; Hales, David (eds.), Springer.

Descartes, R.: 1637, La Géométrie, translated by D. E. Smith and M. L. Latham, Dover Publications, 1954, USA.

Domeshek, E. and Kolodner, J.: 1993, Using the points of large cases, in AI EDAM, 7(2) 87-96.

Doulgerakis, A.: 2007, Genetic Programming + Unfolding Embryology in Automated Layout Planning, Masters thesis, UCL.

Durand, J.N.L.: 1802-5, Précis of the Lectures on Architecture. translated by David Britt. Getty Research Institute, 2000, USA.

Duygulu, P., Barnard, K., de Freitas, J. F. G. and Forsyth, D. A.: 2002, Object recognition as machine translation: learning a lexicon image vocabulary, ECCV 2002.

Eastman, C. M.: 1979, The representation of design problems and maintenance of their structure

Eastman, C. M.: 1975, Spatial Synthesis in Computer-Aided Building Design.

Eastman, C. M. and Siabiris, A.: 1995, A generic building product model incorporating building type information, Automation in construction 3 (1995) 283-304.

Eastman, C., Teicholz, P., Sacks, R. and Listion, K.: 2011, BIM handbook: a guide to building information modeling for owners, managers, designers, enginners and contractors, 2nd edition, John Wiley & Sons, New Jersey.

Edgerton, S. Y.: 1974, Florentine Interest in Ptolemaic cartography as background for Renaissance painting, architecture, and the discovery of America. Journal of the society of architectural historians, Vol. 33, No.4, pp.275-292.

Eigen, M.: 1992, Steps towards life: A perspective on evolution. Oxford University Press, USA.

Eigen, M. and Schuster, P.: 1977, The Hypercycle: A principle of natural self-organization, in Springer, in Naturwissenschaften, 1977-11-01, p541-p565, Springer Berlin/Heidelberg

Einstein A.: 1921, Geometry and Experience, Lecture before the Prussian Academy of Sciences, January 27, 1921.

Eisenhart, L. P. 1997, Riemannian geometry. Princeton University press. Elam, K.: 2001, Geometry of design. Princeton Architectural Press, New York.

Euclid, c.300BC (2011), Elements. http://www.gap-system.org/~history/HistTopics/Euclid_definitions.html, 2/12/2011

Fasoulaki, E.: 2008, Integrated design: a generative multi-performative design approach. Master of Science in Architecture Studies at the Massachusetts Institute of Technology June 2008.

Fine, A.: 2009, Fictionalism. in: M. Suarez (ed.) Fictions in science: philosophical essays on modeling and idealization, Routledge, New York. pp.19-36.

Flemming, U. 1990, Syntactic Structures in Architecture: Teaching Composition with Computer Assistance, in: M. McCullough, W. J. Mitchell and P. Purcell (eds.), 1990, The Electronic Design Studio, MIT, London.

Flemming, U.: 1994, Case-based design in the SEED system, in Automation in Construction 3, 1994, pp.123-133.

Flemming, U. and Chien, S. F.: 1995, Schematic Layout Design in Seed Environment, in Journal of architectural engineering, December, p.162- 169.

Fodor, J.: 1974, Special Sciences: Or the Disunity of Science as a Working Hypothesis. Synthese, 28: 97-115

Fogel, L. J., Owens, A. J. and Walsh, M. J.: 1966, Artificial Intelligence through Simulated Evolution, John Wiley&Sons

Fogel. D. B.: 1995(2006), Evolution Computation: Towards a new philosophy of machine intelligence. Wiley, New Jersey.

Frampton, k.: 1983, Towards a Critical Regionalism: Six Points for an Architecture of Resistance", in The Anti-Aesthetic. Essays on Postmodern Culture (1983) edited by Hal Foster, Bay Press, Seattle.

Francis, L.: 2001, Neural networks demystified, Casualty Actuarial Society Forum, Winter 2001, 253-320.

Frazer, J. 1995. An Evolutionary Architecture. John Frazer and Architectural Association, London.

Frigg, R. and Hartmann, S.: 2012, Models in Science, in The Stanford Encyclopedia of Philosophy (Fall 2012 Edition), Edward N. Zalta (ed.) URL = <http://plato.stanford.edu/archives/fall2012/entries/models-science/>.

Gardner, M.: 1970, Mathematical Games-The fantastic combinations of John Conway's new solitaire game "life", Scientific American 223 October 1970, 20-123

Gen, M. and Cheng, R.: 1997, Genetic Algorithms&Engineering design, John Wiley&Sons, USA.

Gero, J. S.: 1975, Architectural optimization – a review, Engineering optimization, 1:3, 189-199.

Gero, J. S.: 1990, Design Prototypes: A knowledge representation schema for design, AI Magazine Volume 11 Number 4. pp.26-36.

Gero, J. S.: 1996, Creativity, emergence and evolution in design, in Knowledge-based systems 9 (1996) p. 435-448.

Gero, J. S.: 2002, Computational models of creative designing based on situated cognition. in C&C '02 Proceedings of the 4th conference on Creativity & cognition, New York.

Godfrey-Smith, P.: 2009, Models and Fictions in Science, Philosophical Studies 143 (2009) 101-116.

Goel, V. and Pirolli, P.: 1992, The structure of design problem spaces, in Cognitive Science 16, p.359-429.

Gold, S. and Rangarajan, A.: 1996, A Graduated Assignment Algorithm for Graph Matching, in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 18, NO. 4, APRIL 1996, pp. 377,-388.

Goldberg, D. E.:1989, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-wesley, US.

Grant, D. P.: 1993, Housing location for low income residents: an architectural case study of simulating conflicts of interest and generating compromise proposals. in: M. J. de Vries, N. Cross and D. P. Grant (eds.) Design methodology and relationships with science, 1993, Kluwer Academic Publishers, The Netherlands.pp.63-101.

Gropius, W.: 1931, Flach-, Mittel-oder Hochbau?, Schweizerische Bauzeitung.

Halevy, A., Norvig, P. and Pereira, F.: 2009, The unreasonable effectiveness of data, IEEE Intelligent systems, 24 (2), 8-12.

Hamming, R. W.: 1980, The Unreasonable Effectiveness of Mathematics. In American mathematical monthly, 87 (2), 81-90.

Hammond, K. J.: 1990, Case-based planning: a framework for planning from experience. Cognitive science 14 (1990) 385-443.

Han, J. and Kamber, M.: 2006, Data mining: concepts and techniques, Elsevier, USA.

Hand, D. J., Mannila, H. and Smyth, P.: 2001, Principles of data mining, MIT, USA.

Hart, V. and Hicks, P.: 1996, Sebastiano Serlio on Architecture, Volume one, Yale University Press, Great Britain.

Hawking, S. W. and Penrose, R.: 1996, The nature of space and time, Princeton University Press, Princeton, New Jersey.

Haymaker, J., Fischer, M. and Kunz, J.: 2002, Perspectors: inferring spatial relations from building product models. in: J. S. Gero (ed.), Artificial Intelligence in Design'02, Kluwer Academic Publishers, Dordrecht. Netherlands. pp.593-615

Heath, T.: 1993, Social aspects of creativity and their impact on creativity modeling, in: J. S. Gero, J. S. and M. L. Maher, M (eds.), Modeling creativity and knowledge based creative desing, New Jersey: Lawrence Erlbaum Associates, pp.9-24.

Hensel, M., Menges, A., and Weinstock, M.: 2004, Finding exotic form: an evolution of form finding as a design method. in Hesel, M., Menges, A., and Weinstock, M (eds.) Emergence: morphogenetic design strategies. Wiley-Academy, London.

Hensel, M., Menges, A., and Weinstock, M. (eds.): 2004, Emergence: morphogenetic design strategies, Wiley-Academy, London.

Hensel M. and Menges A.:2008, Versatility and Vicissitude: Performance in Morpho- Ecological Design, Wiley-Academy, London.

Herr C. M.: 2007, Adapting cellular automata to support the architectural design process, in Automation in Construction 16 (2007) p61-69.

Heylighen, A. and Neuckermans, H.: 2001, A case base of case-based design tools for architecture. in Computer-aided design 33(2001) 1111-1122

Hillier, B. and Hanson, J.: 1984, The Social Logic of Space. Cambridge University Press.

Hiller, B.: 1986, Ideas are in things: an application of the space syntax method to discovering house genotypes

Hillier, B, 1996, Space is the machine.

Holland, W. D.: 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

Hovestadt, L. 2010. Beyond the Grid - Architecture and information technology, Birkhäuser, Germany.

Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., Twigger, S., White, O. and Rhee, S. Y.: 2008, Big data: the future of biocuration. Nature, 455, 47–50.

Hua, H.: 2012, Decoupling Grid and Volume: a Generative approach to architectural design. in Achten, Henri; Pavlicek, Jiri; Hulin, Jaroslav; Matejdan, Dana (eds.), Digital Physicality - Proceedings of the 30th eCAADe Conference - Volume 1, pp. 311-317.

Hua, H.: 2013, Multiplying Architectural Layouts and 3D Forms: Interplay of Necessity and Contingency in Architectural Modeling. Global Design and Local Materialization, Communications in Computer and Information Science Volume 369, 2013, pp.338-346.

Hua, K. and Faltings, B.: 1993, Exploring case-based building design-CADRE, Artificial Intelligence for engineering, design, analysis and manufacturing, 7 (02) 135-143.

Kahn C. H.: 2001, Pythagoras and the Pythagoreans: a brief history, Hackett Publishing, US.

Kari, L. and Rozenberg, G.: 2008, The many facets of natural computing, Communications of the ACM, 51(10) 72-83.

Kalogerakis, E., Chaudhuri, S., Koller, D. and Koltun, V.: 2012, A probabilistic model for component-based shape synthesis. ACM Trans. Graph. 31, 4, 55:1–55:11.

Kauffman, S.A.: 1993: The origins of order: Self-organization and selection in evolution, Oxford, New York.

Kennedy, J. and Eberhart, R.: 1995, Particle swarm optimization, IEEE international conference on neural networks, 4, 1942-1948.

Khabazi, Z.: 2010, Generative Algorithms: using Grasshopper, http://www.grasshopper3d.com/page/tutorials-1. 15.12.2011

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P.: 1983, Optimization by Simulated Annealing. Science 220 (4598) 671–680.

Kleene, S. C.: 1967(2002), Mathematical Logic. Dover.

Kleppe, A., Warmer, S. and Bast, W.: 2003, MDA Explained. The Model Driven Architecture: Practice and Promise, Addison-Wesley.

Knight, T. and Stiny, G.: 2001, Classical and not-classical computation, arq: architectural research quarterly, 5(04) 355-372.

Knorr, W. R.: 1975, The Evolution of the Euclidean Elements. D. Reidel Publishing, The Netherlands.

Knight, T. 1994. Transformations in design. : A Formal Approach to Stylistic Change and Innovation in the Visual Arts. Cambridge University Press.

Kolarevic, B.: 2003a, Digital Morphogenesis, in: B. Kolarevic (ed) 2003, Architecture in the digital age: design and manufacturing. Spon press, New York, pp. 11-28

Kolarevic, B.: 2003b, Architecture in the digital age: design and manufacturing. Spon press, New York

Kolarevic, B. and Malkawi, A. (eds.) 2005, Performative Architecture: Beyond Instrumentality, Routledge, New York.

Kolodner, J. L.: 1992, An Introduction to Case-based Reasoning, in Artificial Intelligence Review 6, 3-34.

Kolodner, J. L.: 1993, Case-based reasoning. Morgan Kaufmann, New York.

Koning, H. 1981, The language of the prairie: Frank Lloyd Wright's prairie houses.

Koonin, E. V. and Wolf, Y. I.: 2009, Is evolution Darwinian or/and Lamarckian? Biology Direct 2009, 4:42

Koopmans, T.C and Beckmann, M.: 1957, Assignment problems and the location of economic activities. Econometrica, 25 (1) 52-76

Koza, J. R.: 1992, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press

Koza, J. R. and Roli, R.: 2005, Genetic Programming, in Search Methodologies, Burke, E. K. and Kendall, G.(eds), Springer, p127-164.

Kühne, T.: 2006, Matters of (meta-) modeling, Software & systems modeling, 5(4) 2006, 369-385.

Kurpjuweit, S. and Winter, R.: 2007, Viewpoint-based meta model engineering, in EMISA, ser. LNI, M. Reichert, S. Strecker, and K. Turowski, Eds., vol. P-119.GI, 2007, pp. 143–161.

Krejcirik, 1969, Computer-aided plant layout, Computer-Aided Design, 2, 7-19.

Laird, J. E., Newell, A. and Rosenbloom, P. S.: 1987, SOAR: An architecture for general intelligence, Artificial Intelligence 33(1987) p.1-64.

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

Ledoux, C. N.: 1084, L'Architecture, Premier volume, 1981,URL Verlag.

Leeuwen, J. P. van: 1999, Modelling Architectural Design Information by Features, Ph.D. Thesis, Eindhoven University of Technology/Faculteit Bouwkunde, Bouwstenen 54, Eindhoven.

Leeuwen, J. van, Hendricx, A. and Fridqvist, S.: 2001, Towards dynamic information modeling in architectural design, in: Proceedings of CIB W78 2001, Mpumalanga, South Africa, 10 May-1 June, 2001.

Lehnerer, A. and Braach, M.: 2010, Stadtraum Hauptbahnhof (Zurich, CH) in Hovestadt, L.: 2010, Beyond the Grid - Architecture and information technology, Birkhäuser, Germany. p.94-101.

Li, J. and Wang, J. Z.: 2003, Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 9, SEPTEMBER 2003, pp.1075-1088.

Libeskind, D.: 2000, Daniel Libeskind: the space of encounter. Universe. New York.

Liddell, H. G. and Scott, R.:1940, A Greek-English Lexicon, 08.05.2013. URL=http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0057:entry=meta/

Logan, B. and Smithers, T.: 1993, Creativity and design as exploration. in: J. S. Gero, J. S. and M. L. Maher, M (eds.), Modeling creativity and knowledge based creative design, New Jersey: Lawrence Erlbaum Associates, pp.139-176.

Lorensen, W.E. and Cline, H. E.:1987, Marching Cubes: A high resolution 3D surface construction algorithm. In: Computer Graphics, Vol. 21, Nr. 4, July 1987.

Lorenz, E. N.: 1963 Deterministic nonperiodic flow. Journal of Atmospheric Sciences. Vol.20, pp.130-141.

Lovelace, A.: 1842, notes upon the Memoir, sketch of the Analytic Engine invented by Charles Babbage. 08.04.2013. http://www.fourmilab.ch/babbage/sketch.html

Madec, P.: 1989, Etienne-Louis Boullée, Birkhäuser Verlag Basel, Germany.

Madrazo, L. 1994, Durand and the Science of Architecture. in Journal of architectural education. Vol.48, No. 1, pp. 12-24.

Maher, M. L. and de Silva Garza, A. G.: 1997, Case-based reasoning in design, IEEE expert.Mar/Apr 1997, 12(2) 34-41

Malin, S.: 2001, Nature loves to hide: quantum physics and reality, a western prespective. Oxford university press. New York.

Mallgrave, H. F. (ed) 2006, Architectural theory, Blackwell publishing.

March.L and Steadman. P, 1971, The Geometry of Environment: An Introduction to Spatial Organization in Design. MIT Press.

Mark, E., Gross, M. and Goldschmidt, G.: 2008, A Perspective on Computer Aided Design after Four Decades, eCAADe 2008, pp.169-176.

McCarthy, J. and Hayes, P. J.: 1969, Some philosophical problems from the standpoint of artificial intelligence, in: Machine Intelligence, 4, pp.463-502.

Menges, A.: 2006, Instrumental geometry, in: Techniques and Technologies in Morphogenetic Design, Architectural Design, 76(2) 42-53.

Menges, A.: 2012, Biomimetic design processes in architecture: morphogenetic and evolutionary computational design, Bioinspir. Biomim. 7 (2012) 015003.

Meredith, M. 2008. From Control to Design: Parametric/Algorithmic Architecture

Merrell, P., Schkufza, E. and Koltun, V.: 2010, Computer-generated residential building layouts, ACM Transactions on Graphics, 29 (6) 2010, No. 181.

Miranda, P.:2005, http://armyofclerks.net/ArchiKluge/index.htm, 2012.2.3

Minsky, M.: 1986, The Society of Mind. Simon and Schuster, New York.

Minsky, M.: 1961, Steps toward artificial intelligence, Proceedings of the IRE, 49 (1) pp.8-30.

Mitchell, W. J. 1977, Computer-Aided Architectural Design. John Wiley & Sons, Inc. NY, USA.

Mitchell, W. J.: 1990, The logic of architecture: design, computation, and cognition. MIT press, Cambridge, MA, USA.

Mitchell, W. J. and Dillon, R. L.: 1972, A polyomino assembly procedure for architectural floor planning. Environmental Design: Research and Practice W. Mitchell (ed.) (UCLA, 1972), 2, pp. 23/51-11.

Mora, R., Rivard, H. and Bédard, C.: 2004, Computer-aided conceptual design of building structures: geometric modeling for the synthesis process, in: J. S. Gero (ed), Design Computing and cognition'04,Kluwer Academic Publishers, Dordrecht, Netherlands, pp.37-55.

Morris, H. M.: 2009, Scientific creationism. Master Books, USA.

Morrison, M.: 2011, One phenomenon, many models: Inconsistency and complementarity, studies in history and philosophy of science, 42 (2011) 342-351.

Motta, E.: 1999, Reusable components for knowledge modeling: case studies in parametric design problem solving. IOS Press, Netherlands.

Müller, P., Wonka, P., Haegler, S., Ulmer, A. and Van Gool, L.: 2006, Procedual modeling of buildings. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2006, Vol. 25, Issue. 3. pp.614-623.

Mylopoulos, J and Levesque, H.: 1983, An overview of knowledge representation, Informatik-Fachberichte, 76, 1983, 143-157.

Naylor, T.H., Finger, J. M., McKenney, J. L., Schrank, W. E. and Holt, C. C.: 1967, Verification of computer simulation models, Management Science, 14(2) B92-B106.

Newell, A. and Simon, H. A.: 1956, The logic theory machine: a complex information processing system, IRE Transactions on information theory, 2(3) 61-79.

Newell, A. and Simon, H. A.: 1972, Human problem solving, Prentice-hall, Englewood Cliffs. N.J. US.

Nock, R. and Nielsen, F.: 2004, Statistical Region Merging, in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 26(11), NOVEMBER 2004, 1452-1458.

Newell, A. and Simon, H. A.: 1976, Computer science as empirical enquiry. Communications of the ACM 19(3):113–26.

Newell, A.: 1981, The knowledge level, AI MAGAZINE, AI Magazine 2(2) 18 (1981) 1-33

O'Connor, J. J. and Robertson, E. F.: 2006, Quotations by Galileo Galilei, URL= http://www-history.mcs.st-and.ac.uk/Quotations/Galileo.html, 2.5.2013

OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2, November 2007, URL=http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF

Oreskes, N., Shrader-Frechette, K. and Belitz, K.: 1994, Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences, Science, New Series, 263(5147) (Feb. 4, 1994), 641-646.

Osborne, M. J.: 2009, An introduction to game theory. Oxford University press. URL= http://www.ius.edu.ba/sfadda/An_Introduction_to_Game_Theory.pdf

Oxman, R.: Performance-based Design: 2008, Current Practices and Research Issues, International Journal of Architectural computing. Vol 6, No.1 pp.1-17.

Oxman, R. and Oxman, R.: 1990, The computability of Architectural Knowledge. In: McCullough, M. Mitchell, W. J. and P. Purcell (eds.), 1990, The Electronic Design Studio, MIT Press, London. pp.171-p185

Oxman, R. and Oxman, R.: 1994, Case-based design: cognitive models fro case libraries, in: G. Carrara and Y. E. Kalay (eds.) Knowledge-based computer-aided architectural design, Elsevier Science, The Netherlands. pp.45-68.

Padovan, R.: 1999, Proportion: Science, philosophy, architecture. Spon press, London.

Paley, W.: 1809 (1857), Natural Theology: or, Evidences of the Existence and Attributes of the Deity. Gould and Lincoln, New York.

Papalambros, P. Y. and Wilde, D. J.: 2000, Principles of optimal design: modeling and computation, 2nd edition, Cambridge university press, USA.

Parish, Y. I. H. and Müller, P. 2001, Procedural modeling of cities. Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 301-308.

Pazlar, T. and Turk, Z.: 2008, Interoperability in practice: geometric data exchange using the IFC standard, ITcon Vol. 13, Special Issue Case studies of BIM use, 362-380.

Pérez-Gómez, A.: 1983, Architecture and the Crisis of Modern Science. MIT press, US.

Penttilä, H.: 2007, Early architectural design and BIM, CAADFutures 2007, Springer, 291-302.

Perrault, C.: 1683, Ordonnance for the Five Kinds of Columns after the Method of the Ancients. in: H. F. Mallgrave (ed) 2006, Architectural theory, Blackwell publishing. pp.78-80.

Perlin, K.: 2002, Improving noise, in ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2002, Vol. 21, issue 3.

Peschard, I.: 2011, Making sense of modeling: beyond representation. in European Journal for Philosophy of Science , 1(3): p.335-352.

Pidcock, W.: 2013, What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model? 30. 04. 2013, URL=http://infogrid.org/trac/wiki/Reference/PidcockArticle

Pinsky, M. A. and Karlin, S.: 2011, An introduction to stochastic modeling, fourth edition, Elsevier, USA.

Plato.: 350 B.C, Sophist, translated by Benjamin Jowett, http://classics.mit.edu/Plato/sophist.html, 16.09.2012

Plato.: 2008, Timaeus, translated by Jowett B. The project Gutenberg Ebook. http://www.gutenberg.org/files/ 1572/1572-h/1572-h.htm, 24.11.2012

Popper. k 2012 http://en.wikipedia.org/wiki/Karl_Popper

Posada, D. and Crandall, K. A.: 1998, MODELTEST: testing the model of DNA substitution.

Pottmann, H., Asperl, A., Hofer, M. and Kilian, A.: 2007, Architectural Geometry. Bentley Institute Press. Prigogine, I. 1984, Order out of Chaos, Bantam

Prokopenko, M.: 2008: Design vs. Self-organization, in Advances in applied self-organizing systems, Prokopenko, M(ed.), Springer. p3-17.

Putnam, H.: 1967, Psychological Predicates. In W.H. Capitan and D.D. Merrill (eds.), Art, Mind, and Religion. Pittsburgh: University of Pittsburgh Press, pp.37-48.

Quigley, E. J. and Debons, A.: 1999, Interrogative Theory of Information and Knowledge", in Proceedings of SIGCPR '99, ACM Press, New Orleans, LA., 1999, pp.4-10.

Radford, A. and Gero, J. S.: 1988, Design by optimization in architecture, building, and construction.

Rechenberg, I.: 1973: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog.

Reynolds, C. W. :1987, Flocks, herds, and schools: a distributed behavioral model. in Computer graphics, vol. 21, number 4, July 1987.

Richter, K., Heylighen, A. and Donath, D.: 2007, Looking back to the future: an update case base of case- based design tools for architecture. 25th eCAADe Conference Proceedings, Frankfurt am Main (Germany) 26-29 September 2007, pp. 285-292

Rittel, H. W. J.: 1988, The reasoning of designers, working paper A-88-4, Institut für Grundlagen der Plannung, Stuttgart.

Rittel, H. and Webber, M.: 1973, Dilemmas in a General Theory of Planning, in Policy Science, Elsevier scientific Publishing, Amsterdam.

Rocha, L. M.: 1999, Complex Systems Modeling: Using Metaphors From Nature in Simulation and Scientific Models. BITS: Computer and Communications News. Computing, Information, and Communications Division. Los Alamos National Laboratory.

Rosenblatt, F.:1958, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408.

Rosenman, M. A. and Gero, J. S.: 1996, Modelling multiple views of design objects in collaborative cad environment. in Computer-Aided Design, Vol. 28, 1996, p.193-205.

Rosenman, M. A.: 1996, The generation of form using an evolutionary approach, in Gero, J. S. and Sudweeks, F. (eds), Artificial Intelligence in design'96, Kluwer Academic, Dordrecht. pp.643-662.

Rosenman, M.: 2000, Case-based Evolutionary Design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 14, 17-29.

Rothenberg, J.: 1989, The Nature of Modeling, in: L. E. Widman, K. A. Loparo and N. R. Nielso (eds.), Artificial Intelligence, Simulation, and Modeling, John Wiley & Sons, pp.75-92.

Rouse, J.: 2002, How Scientific Practices Matter. Reclaiming Philosophical Naturalism. The University of Chicago Press.

Rowe, C.: 1947, The Mathematics of the Ideal Villa: Palladio and Le Corbusier Compared, Architectural Review, 1947, 101-104.

Rowe, C. and Slutzky, R .: 1963, Transparency: Literal and phenomenal. in , Vol. 8. (1963), p.45-54.
Rozenberg, G and Salomaa, A.: 1997, The mathematical theory of L systems. Academic press. New York.

Rumelhart, D. E., Geoffrey E. H., and Williams, R. J.: 1986, Learning Internal Representations by Error Propagation. in Rumelhart, D.E., James L. M. and the PDP research group. (eds.), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.

Saad, M. and Maher M.L.: 1993, A computational model for synchronous collaborative design. in AAAI Technical Report WS-93-07.

Savage, J. E.: 1998, Models of computation: exploring the power of computing, Addison-Wesley.

Scheurer, F. and Stehling, H.: 2011, Lost in parametric space? in: Mathematics of space, Architectural Design, 81(4) 70-79.

Schlüter, A.: 2010, Model-based, performance-oriented building design employing distributed service systems. Ph.D thesis, ETH, zuirch.

Schmitt, G.:1993, Case-based Design and creativity. in Automation in Construction 2 (1993) 11-19. pp. 11-19

Scholfield, P. H.: 1958, The theory of proportion in architecture. Cambridge university press, London.

Schumacher, P.: 2008, Parametricism as Style - Parametricist Manifesto, 11th Architecture Biennale, Venice 2008.

Schumacher, P.: 2009, Parametricism: A New Global Style for Architecture and Urban Design. in Leach, N (ed.) Digital cities, AD: Architectural Design, 79 (4) 14-23.

Seehof, J. M. 1966, Automated facilities layout programs

Shanks, N. and Joplin, K. H.:1999, Philosophy of Science, 66 (June 1999) 268-282.

Shea, K. and Cagan, J.: 1999, Languages and semantics of grammatical discrete structures. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), 13, 241–251.

Simon, H. A.: 1996, The Sciences of the Artificial. MIT Press, Cambridge, Massachusetts.

Simulated Annealing, 31.03.2013, http://en.wikipedia.org/wiki/Simulated_annealing

Slade, S.: 1991, Case-based reasoning: A research paradigm. AI Magazine, 42-55.

Smeulders, A. W. M.: 2000, Content-Based Image Retrieval at the End of the Early Years, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 22 (12) 1349-1380.

Smith, B. C.: 1985, The limits of correctness, ACM SIGCAS Computers and Society, 14 (4) 18-26.

Spek, R. V. D. and Spijkervet, A.:1997, Knowledge Management: Dealing Intelligently with Knowledge, CIBIT, Utrecht.

Steadman, P.: 1975, Models in our heads, models in the material world, and models in the world of objectives knowledge, in: Hawkes, D.(ed.) Models and systems in architecture and building. The Martin centre for architectural and urban studies. The construction press, England. p. 29-34.

Steels, L.: 1994: The artificial life roots of artificial intelligence. Artificial Life 1, (1994) pp.75-110.

Stenmark, D.: 2002, Information vs. Knowledge: the role of intranets in knowledge management. in Proceedings of the 35th Hawaii International Conference on System Sciences.

Stevens, G.: 1990, The Reasoning Architect: Mathematics and Science in Design. McGraw-hill, Singapore.

Stachowiak H.: 1973, Allgemeine Modelltheroie, Springer-Verlag.

Št'ava, O.: 2010, Inverse Procedural Modeling by Automatic Generation of L-systems

Stiny, G.1972, Shape Grammars and the Generative Specification of Painting and Sculpture. In: O. R. Petrocelli (ed.) The Best Computer Papers of 1971 (Auerbach, Philadelphia, 1972) pp.125-135.

Stiny, G. and Mitchell, W. J.: 1978, The Palladian grammar, in Environment and Planning B. vol.5, p.5-18.

Studer, R., Benjamins, V. R. and Fensel, D.:1998, Knowledge engineering: principles and methods. Data and Knowledge Engineering, 25(1-2),1998,161–197.

Styhre, A. and Sundgren, M.: 2003, Creativity as connectivity: a rhizome model of creativity, Int. J. Internet and Enterprise Management, Vol. 1, No.4, 2003, p.421-436.

Suppes, P.: 2002, Representation and Invariance of Scientific Structures. Stanford: CSLI Publications.

Sutherland, I. E.: 1963, Sketchpad, a man-machine graphical communication system. PhD thesis, MIT.

Tabor, P.:1976, Analysing communication patterns, In March. L.:1976, The architecture of Form.

Terzidis, K.: 2006, Algorithmic Architecture, Elsevier, Oxford.

Takala, T. 1993.: A neuropsychologically-based approach to creativity. in: J. S. Gero, J. S. and M. L. Maher, M (eds.), Modeling creativity and knowledge based creative desing, New Jersey: Lawrence Erlbaum Associates, pp.91-108.

Thompson D. W. 1917, On growth and form, Cambridge Press, 2004,UK.

Toxiclibs, 31.03.2013, http://toxiclibs.org/docs/core/toxi/geom/SutherlandHodgemanClipper.html

Turk, M. and Pentland, A.: 1991, Eigenfaces for recognition. in Journal of Cognitive Neuroscience, Vol 3. No.1

Turing, A.: 1950, Computing machinery and intelligence. Mind, 59 (236) 436-460.

Turing, A.: 1952, The chemical basis of morphogenesis, in Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 237 (641) 37-72.

Ullmann, J. R.: 1976, An algorithm for subgraph isomorphism. in Journal the ACM. Vol. 23, Issue 1, Jan. pp. 31-42.

van Doesburg, T, 1966, Grundbegriffe der Neuen Gestaltenden Kunst

van Fraassen, B.: 1980, The Scientific Image , Oxford Univ. Press, New York.

van Gigch, J. P.: 1991, Metamodeling: the epistemology of system science, system practice, 6(3) 1993, 251-258.

Venturi, R.: 1966, Complexity and Contradiction in Architecture,The Museum of Modern Art Press, New York.

Verebes, T.: 2009 Experiments in associative urbanism. Architectural design, 79 (4), 24-33.

Vitruvius.: 1960, The ten books on architecture. Translated by Morgan, M. H. Dover publication. US.

Von Storch, H. and Zwiers, F. W.: 2004, Statistical analysis in climate research, Cambridge university press, Cambridge, UK.

Watanabe, S.: 1994, Knowledge integration for architectural design, in: G. Carrara and Y. E. Kalay (eds.) Knowledge-based computer-aided architectural design, Elsevier Science, The Netherlands. pp.123-146.

Watson, I. and Perera, S.: 1997, Case-based design: A review and analysis of building design applications. Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 11 (1) 59-87.

Weber, M., Langenhan, C., Roth-Berghofer, T., Liwicki, M., Dengel A. and Petzold F.: 2010, a.SCatch: semantic structure for architectural floor plan retrieval. ICCBR'10 Proceedings of the 18th international conference on Case-Based Reasoning Research and Development. pp.510-524.

Weinzapfel, G.: 1975, IMAGE: Computer Assistant for Architectural Design, in Charles M. Eastman e.d , 1975, Spatial Synthesis in Computer-Aided Building Design

White, J.: 1949, Developments in Renaissance Perspective: I, Journal of the Warburg and Courtauld Institutes, 12 (1949), 58-79.

Whitehead, and M.Z.Eldars, 1965, The planning of single-storey layouts

Wiger, E. P.: 1959 The Unreasonable Effectiveness of Mathematics in the Natural Sciences, Richard Courant Lecture in Mathematical Sciences delivered at New York University, May 11,1959

Winograd, T. A. and Flores, C. F.: 1986, Understanding computers and cognition: a new foundation for design. Ablex Publishing, USA.

Wittkower, R.: 1960, The changing concept of proportion, in Kepes, G. (ed) The visual arts today, Wesleyan University press. p.203-219.

Wittkower, R.: 1949, Architectural principles in the age of humanism. St. Martin's press, Great Britain.

Wolfram, S.: 1984a, Cellular automata as models of complexity, Nature vol. 311 4 October 1984. 419- 424

Wolfram, S.: 1984b, Universality and complexity in cellular automata. Physica 10D (1984) 1-35

Wolfram, S.: 2002, A new kind of science, Wolfram Media, Inc.

Woodbury, R.: 2010, Elements of Parametric design Wright. L.: 1983, Perspective in perspective. Routledge & Kegan Paul, London.

Wright, S.: 1932, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in Proceeding of the sixth international congress of genetics, Vol.1, 1932, p. 356-66

Zeeman, E. C.:1976, Catastrophe theory, in Scientific American, April 1976, pp.65-70, 75-83.

Zhu, L., Chen, Y. and Yuille, A.: 2009, Unsupervised Learning of probabilistic grammar-markov models for object categories. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 31(1), January 2009, 114-128.

Zwicky, F.: 1969. Discovery, Invention, Research - Through the Morphological Approach.

Figures

Fig 1.1: The structure produced by the generative process, Reptile project, John Frazer, 1966 onwards. (Frazer 1995)
Fig 1.2: Multiple views and models, Rosenman and Gero (1996).
Fig 2.1: The columns of the forum in Pompeii.
23. 04. 2012, http://commons.wikimedia.org/wiki/File:Pompeii_forum2007.JPG
Fig 2.2: Comparison of the illustrations of Vitruvian man (Wittkower 1949).
Fig 2.3: The two pediments “overlap” in the facade of S.Francesco della Vigna, Palladio. (Wittkower 1949).
Fig 2.4: Palladio’s proportion system (Stevens 1990).
Fig 2.5: The proportion in the floor plan of Villa Emo at Fanzola, Palladio (Wittkower 1949).
Fig 2.6: Section, Claude Nicolas Ledoux (1804).
Fig 2.7: Section, Étienne-Louis Boullée (Madec 1989).
Fig 2.8: Comparison of Greek theatre(left) and Roman theatre(right). (Vitruvius 1960).
Fig 2.9: Alberti’s method of perspective (Wright 1983.)
Fig 2.10: Drawing arches, Sebastiano Serlio (Hart and Hicks 1996, 65)
Fig 2.11: Locating the positions of the points on the curve, with x (AB) and y (BC). Descartes (1637).
Fig 2.12: Transformation of shapes through variable coordinate system. Thompson (1917, 301).
Fig 2.13: Elements of building, J. N. L. Durand (1802-5).
Fig 2.14: Developing plan by a series of corquis, J. N. L. Durand (1802-5)
Fig 2.15: Developing plan by a series of corquis, No.2, J. N. L. Durand (1802-5).
Fig 3.1: Turing’s morphogen system (reaction-diffusion system). Turing (1952).
Fig 3.2: Computer, model and the real world, Smith (1985).
Fig 3.3: Model, system and language, Kleppe (2003).
Fig 3.4: Using binary string to represent spatial configurations of 4×4×4 grid, ArchiKluge, Miranda (2005).
Fig 3.5: A floor plan produced by a series of functions (Doulgerakis 2007)
Fig 3.6: Two graphs of interior spaces, Hillier (1984).
Fig 3.7: A graph of casual linkage (Alexander 1964, 43).
Fig 3.8: Model of design as a process (Gero 1990).
Fig 3.9: A linear model of designing (Grant 1993).
Fig 3.10: All possible locations of the new cell (no.13), (Whitehead and Eldars 1965).
Fig 3.11: One of the generated floor plan layouts (Whitehead and Eldars 1965).
Fig 3.12: Crossover the genotypes (Rosenman and Gero 1996).
Fig 3.13: Floor plans generated by Rosenman’s program (Rosenman and Gero 1996).
Fig 3.14: Crossover the genotypes in genetic programming (Doulgerakis 2007).
Fig 3.15: Three layouts produced by the Doulgerakis’s program (Doulgerakis 2007).
Fig 3.16: An exploration-based model of design (Logan and Smithers 1993).

Fig 3.17: A process model for case-based design with evolutionary case adaptation (de Silva Garza and Maher 2001).
Fig 3.18: A cycle of CBR (Kolodner 1992).
Fig 3.19: Comparison of CBD applications in architectural design (Weber et al. 2010).
Fig 3.20: The user interface of the Archie-II system (Domeshek and Kolodner 1993).
Fig 3.21: The method of encoding cases: the 3×3 Grid applied to the landscape level, house level and room level. (de Silva Garza and Maher, 2000).
Fig 3.22: One solution produced by case adaptation/combination (de Silva Garza and Maher, 2000).
Fig 3.23: A parametric model of three rooms (Woodbury 2010, 25).
Fig 3.24: Parametric modeling with Grasshopper (Khabazi 2010, 12).
Fig 3.25: Morphologies in the evolutionary process (Menges 2012).
Fig 3.26: Multiple Views and Models (Rosenman and Gero 1996).
Fig 4.1: A single wake behind a cylinder (left). Double wakes behind a pair of cylinders (right), Peschard (2011).
Fig 4.2: A morphological box for three variables.
Fig 4.3: System, model and metamodel, Bézivin (2005).
Fig 4.4: System, model, metamodel, and meta-metamodel, Kühne (2006).
Fig 4.5: Model and Metamodel in UML, OMG UML (2007).
Fig 5.1: The interplay of volume and grid, Eisenman. 08.05.2012, http://www.deax.it/sei/images/stories/sezion/B2/B2-2b.jpg
Fig 5.2: The grid is in red, the cuboid volume in black.
Fig 5.3: Search for the proper composition based on the volume and the grid.
Fig 5.4: Solution A, perspective.
Fig 5.5: Solution A, floor plans.
Fig 5.6: Solution B, perspective.
Fig 5.7: Solution C, perspective.
Fig 5.8: The four “verbs” of the grammar.
Fig 5.9: Collections of generated compositions, set A.
Fig 5.10: Collections of generated compositions, set B.
Fig 5.11: The specification of rooms.
Fig 5.12: One shape model specifying a set of parametric shapes.
Fig 5.13: The parametric “T” shape.
Fig 5.14: Selecting detected polygons as the rooms in the floor plan.
Fig.5.16: Four results of specification I (Fig. 5.11-5.12).
Fig 5.17: Room specification II.
Fig 5.18: Shape specification II.
Fig 5.19: Four results of specification II (Fig 5.17-5.18).
Fig 5.20: The depth analysis.
Fig 5.21: Variations of dividing the blob.
Fig 5.22: Assigning rooms to blobs, iteration 400 (840 iterations in total).
Fig 5.23: One solution generated from the image of natural objects.

Fig 5.24: One solution generated from image of painting (Paul Klee). The image is from: 21.09.2011, http://jesseturri.com/wordpress/wp-content/uploads/2010/06/klee17.jpg
Fig 5.25: Another solution generated from image of painting (Paul Klee). The image is from: 21.09.2011, http://29.media.tumblr.com/tumblr_lf8xenMxSP1qzultro1_r1_500.png
Fig 5.26: One solution generated from the image of collage (Kurt Schwitters). The image is from: 21.09.2011, http://api.ning.com/files/nIFQFJ9rwrBJET9i0nhAQLlw6nQnWdbPsgm0V-RFOyK0HH*o57AriKB7CMm-lfSKOkx7UAeyxIenDKTHDZAUi14pcWYl6vlE/schwitters.jpg
Fig 5.27: The 3d geometry generated by the "Perlin" model (left); the geometry generated by the "Cubes" model (right).
Fig 5.28: The 2d planes resulted from the "Perlin" model.
Fig 5.29: One layout generated by the "Central Corridor" model, when all the areas on the plane are positive.
Fig 5.30: Analysis of the central corridor layouts.
Fig 5.31: A corridor generated by the "Voronoi" model (a); Grouping cells into rooms (b).
Fig 5.32: The layouts and the form generated by Central Corridor – Cubes.
Fig 5.33: The layouts and the form generated by Voronoi – Cubes.
Fig 5.34: The layouts and the form generated by Central Corridor – Perlin.
Fig 5.35: The layouts and the form generated by Voronoi – Perlin.
Fig 5.36: One result of recognition.
Fig 5.37: Three topologies: LINEAR, CIRCLE, and BRANCH (from left to right).
Fig 5.38: Two results for the LINEAR topology.
Fig 5.39: Two results for the CIRCLE topology.
Fig 5.40: Two results for the BRANCH topology.
Fig 5.41: Packing the generated buildings in the parcels of a virtual construction site.

Acknowledgements

Professor Ludger Hovestadt created a research team that have been attracting and inspiring me a lot. I am particularly grateful for his guide during my doctoral study. As my supervisor, he gives me great freedom on research topics on one hand; and pushes me implicitly but intensively on the conceptual level on the other hand. He often starts where I am going to stop; thus, it is often both confusing and exciting to take a new step. I would like to express my deep gratitude to Professor Li Biao for his continuous encouragement for my study. He has given me constructive suggestions and useful critiques on my work. It is very enjoyable to discuss with him on research topics and on technical details. I wish to thank Professor Urs Hirschberg for his suggestion and support on my research. Thank Professor Zhang Hong for his long-term support since my master study.

I wish to express my appreciation to the facilities and the services from ETH Zurich, especially the brilliant service of the library. My special thinks to China Scholarship Council, Education Affairs Office of the Embassy of China (Switzerland), and Southeast University. They make my study in Switzerland possible.

I would like to thank all the people in the Chair of Ludger Hovestadt. Tom Powlofsky, Mathias Bernhard, Christoph Wartmann gave me very good introductions to their own research fields during the first year of my study. Benjamin Dillenburger has given me a lot of support on computational design. His own works are full of brilliant ideas and thus always inspire me. Thank Michael Hansmeyer for his help in computational geometry and his guide to European cities. Thank Vera Bühlmann and Klaus Wassermann for the introductions to the number theory and the information theory. Special thanks to Manuel Kretzer and Knut Brunier for inviting many interesting lecturers to the chair. I would like to thank Hironori Yoshida, Li Li and Matthias Standfest for sharing their research ideas with me. Thanks to Mario Guala for his great help during my stay. I would like to extend my thanks to all MAS and doctoral students since we have a good time in and outside the chair. Special thanks should be given to Sibylla Spycher for her special care of me; I cannot continue my work smoothly without her assistance and advice.

Great appreciation to my friends Wang Yu, Zhang Yuzhen and Chen Tao. Finally, I am very grateful for the great support from my mother and my father. Thank Jia Tingli who likes chatting with me online and lately founds out that she's my wife.

Curriculum vitae

Personal details

Name: Hao Hua
Date of Birth: 07.02.1983
Place of Birth: Wuxi, Jiangsu
Nationality: China

Education

Since 2009	Doctoral study Chair of Computer-Aided Architectural Design, ETH Zurich Prof. Dr. Ludger Hovestadt
2007-2009	Master School of Architecture, Southeast University, Nanjing Supervisor: Prof. Dr. Zhang Hong
2001-2006	Bachelor School of Architecture, Southeast University, Nanjing