The Computation of Design Vocabulary:
Prototyping, Variation, and Composition

by Lee, Hee Won

Master of Architecture in Architectural Design,
University of California at Los Angeles
June, 1992

Submitted to the Department of Architecture
in partial fulfillment of the requirements for the degree of
Master of Science in Architecture Studies
at the Massachusetts Institute of Technology
June, 1997

Signature of the Author
Lee, Hee Won
Department of Architecture, May 22, 1997

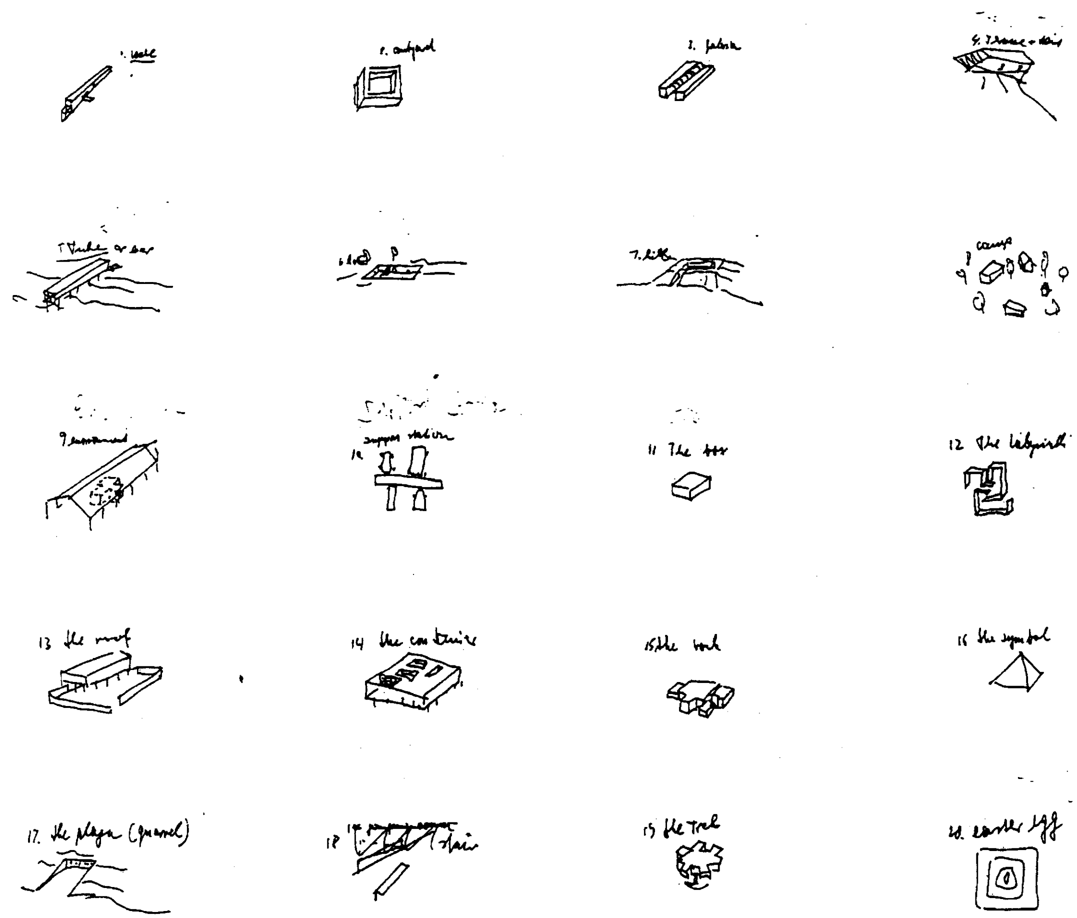
Certified by
William Mitchell
Thesis Advisor, Professor of Architecture

Certified by
Takehiko Nagakura
Thesis Advisor, Assistant Professor of Architecture

Accepted by
Roy Strickland
Chairman, Departmental Committee on Graduate Students

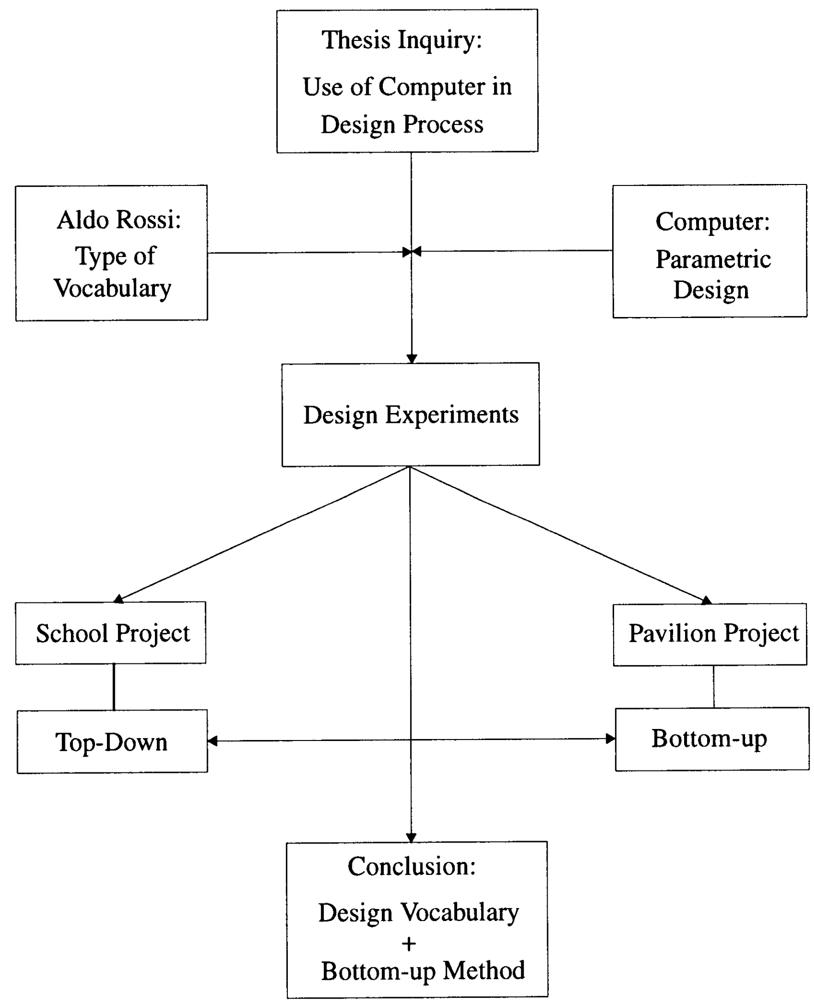
Thesis Reader: George Stiny
Title: Professor of Architecture

Thesis Reader: Andrew Scott
Title: Associate Professor of Architecture



O.M. Ungers' sketch for defining 'house types' from *Architecture as Theme*

Thesis Flowchart



The Computation of Design Vocabulary:
Prototyping, Variation , and Composition

by Lee, Hee Won

Submitted to the Department of Architecture on May 9, 1997 in partial fulfillment of the requirements for the degree of Master of Science in Architecture Studies

Abstract

The range of one's vocabulary affects the solution of architectural problems. The more extensive vocabulary one has, the wider array of solutions one can provide for the needs of man. Thus when we design a building, we must know how vocabulary elements are manipulated and composed for the making of space, structure, and enclosure in response to program and context.

My research interest is in the computation of design vocabulary for transformations and combinations in the design process as well as the shape rules involved in the composition of vocabulary elements into an architectural entity. This thesis investigates how an architectural type can be defined parametrically to fit into an architectural context, how vocabulary elements can be assembled and transformed by top-down and bottom-up methods, and how those forms can be composed by a set of rules. Design experiments follow the theoretical investigation and demonstrate the proposed theory.

The thesis begins with the study of 'typology' in architectural history which establishes a theoretical framework for understanding the historical importance of 'type'. Next follows the analysis of Aldo Rossi's design methodology to understand how elemental forms are manipulated and composed by a set of rules, and how the design method can be encoded into a computer program by shape grammar.

After the historical and theoretical study of type, the computational background of the thesis is examined for possible application of computers in the design process. In Computer-Aided-Design, form-assemblage is the crucial idea for communication of design concepts as well as spatial exploration. The ability to assemble a form from individual graphic elements is the most useful feature in CAD. Shape grammar can take advantage of this CAD capability in classifying basic forms and laying out their transformations for spatial exploration. To illustrate this idea, a knowledge-based system is described with graphic examples.

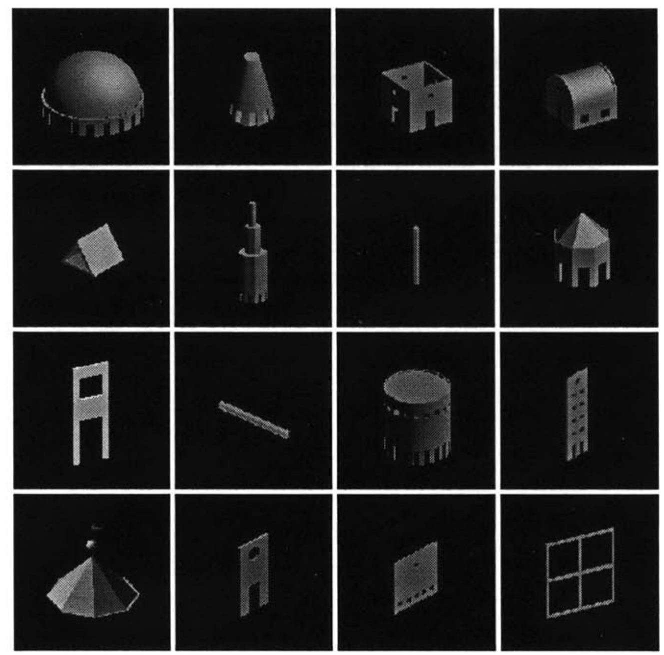
Subsequently, two composition techniques: bottom-up and top-down design methods are studied through computational design experiments which implements the idea of typing vocabulary. Furthermore, two sets of building components are encoded through computer programs to illustrate how vocabulary elements are manipulated in top-down and bottom-up fashions.

In the end, this thesis concludes that the bottom-up design method is a more efficient method of exploring form and space than the top-down design method. It concludes further that the concept of typing vocabulary can play an important role for using a computer efficiently in the design process with the bottom-up method.

Thesis Co-Advisor: William Mitchell
Title: Professor of Architecture

Thesis Co-Advisor: Takehiko Nagakura
Title: Assistant Professor of Architecture

Contents



Rossi Vocabulary

Ideology and Methodology

My fascination with Aldo Rossi's architecture comes from my personal obsession with simplicity in life. Architecture is like life; neither is mastered by man. For this reason, I have consciously and unconsciously fallen in love with everything simple in all aspects of life – hobby, fashion, car, and people, to name a few. Architecture is not an exception; I cannot resist preoccupation with simple architectural forms. This preoccupation has led me to search for 'simplicity' in the principle of architecture, and finally Aldo Rossi's architecture provided me with a religious foundation for the generation of idea and method in design throughout my architectural education.

Aldo Rossi's design methodology inspired my combined interest in design and computation. His buildings are created by repeated use of his own architectural prototypes which he transforms into new design vocabularies in response to site and program. Rossi's prototypal vocabulary is not transient with a stylish trend, but they are meant to be reused when he creates new ideas and new forms. Whenever he designs a building, he uses his existing vocabulary as a base for form-generation; he transforms them into new vocabulary in response to new conditions; and he assembles them into an architectural identity with small shifts. This idea of 'repeatability of vocabulary' provides great potential for use of computers in the design process. Once a designer establishes his or her own vocabulary, he or she can store and retrieve its elements whenever necessary, and manipulate them to fit into space with the capability of parametric variation. Shape grammar plays an important role in manipulating these elemental shapes for spatial composition. It helps to compose basic forms for organizations of space.

There have been numerous studies on the use of computers in the design process, but few explorations on actual design application. The most of the research is limited to an analysis or theory of developing software tools which generally does not demonstrate how the tools are actually going to be used in design. This lack of demonstration inspired me to experiment the use of computers in actual design projects. By actual design explorations, I can learn the advantages and disadvantages of computer use, and thus provide new and better suggestions for educating designers about use of computers both in schools and in professional world. Furthermore I hope that the results of these design explorations will lead to the development of a better software tool.

There is a well-known method in computer use called 'Parametric Design' with two techniques: top-down and bottom-up methods. A knowledge-based system for the top-down design was developed and extensively analyzed for use in design, but the bottom-up design has not been researched or possibly avoided for some reason. Two design projects will be explored with both design methods, particularly with focus on the bottom-design method, in order to find out their effectiveness in the process of making a building. The intention of this investigation on top-down and bottom-up methods is to bring me to a level of confidence where I could lay a theoretical foundation for a continuous search for an architectural principle. And I hope this thesis will also show that it is possible to use computers from " scratch to finish".

Design is the computation of shape information that is needed to guide fabrication or construction of an artifact...The process of design takes different forms in different contexts, but the most usual computational operations are transformations and combinations of shapes in a two-dimensional drawing or a three-dimensional geometric model. An initial vocabulary of shapes, together with a repertoire of shape transformation and combination operators, establishes the shape algebra within which the computation takes place. The computation terminates successfully when it can be shown that certain predicates are satisfied by a shape produced by recursively applying the transformation and combination operators to the initial vocabulary.

William J. Mitchell, The Electronic Design Studio, 1989

1. Type

By definition, 'type' means the general form, structure, plan, and style characterizing the members of a class or group (*Webster's New World Dictionary*). But there have been various interpretations about 'type' among designers and theorists. According to Rafael Moneo in *On Typology* (1978), 'type' in architecture can be defined as "*a concept which describes a group of objects characterized by the same formal structure.*" Moneo thinks that 'type' is fundamentally based on the idea of grouping objects by the similarities of structures. In his article, *On the Typology of Architecture* (1963), Giulio Carlo Argan defines 'type' as a common, original form abstracted from numerous variants. He writes that "*in the process of comparing and superimposing individual forms so as to determine the 'type', particular characteristics of each individual building are eliminated and only those remain which are common to every unit of the series.*" Argan believes that 'type' must have a capability of infinite variations and modification of 'type' itself. U.M. Ungers also talks about a 'type' in another term: a 'model'. He states, in his book *City Metaphors* (1982), that "*a model is commonly understood as something that poses as a prototype representing an ideal form.*" In other words, a model is a basic, ideal structure from which forms of variations are generated. Ungers implemented this concept in his various projects. Figure 1.1 shows the thematic models of plan which serve as a base for all his design. This concept of 'type' has been used and discussed for a long time in the design and theory of architecture. Designers have developed ways of 'typing' to produce and reproduce our built world. They have constructed human places by using certain types of shapes and spaces, reusing them in a certain context when required.

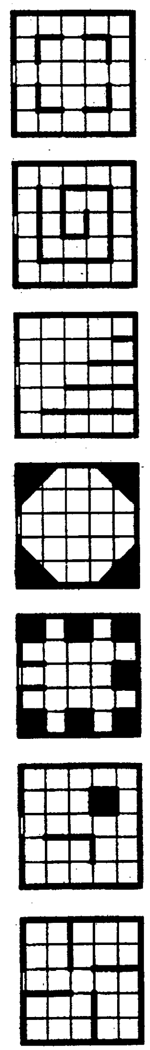


Figure 1.1 Thematic models of plan

1.1 History and Theory

Quatremere de Quincy, a theoretician of neo-Classicism, proposed the first comprehensive, architectural theory of type in his book *Encyclopèdie Methodiquè* by the early nineteenth century. He believes that everything must be generated from an antecedent since nothing comes from nothing. Thus in his opinion, this concept must apply to all human inventions as well as architecture. He thinks all buildings must be born of something ideal from the past, for example, the Parthenon. The most well-known theory of typology

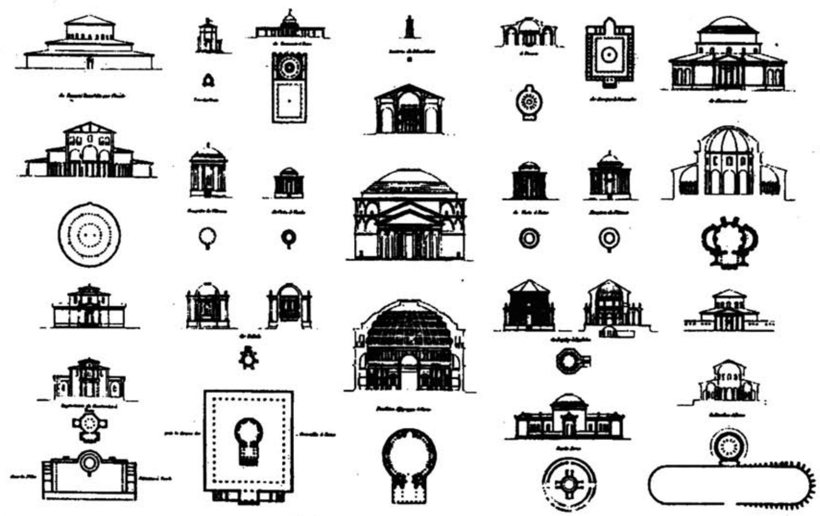


Figure 1.1.1 Classification of round temples from Durand's *Recueil,* 1801

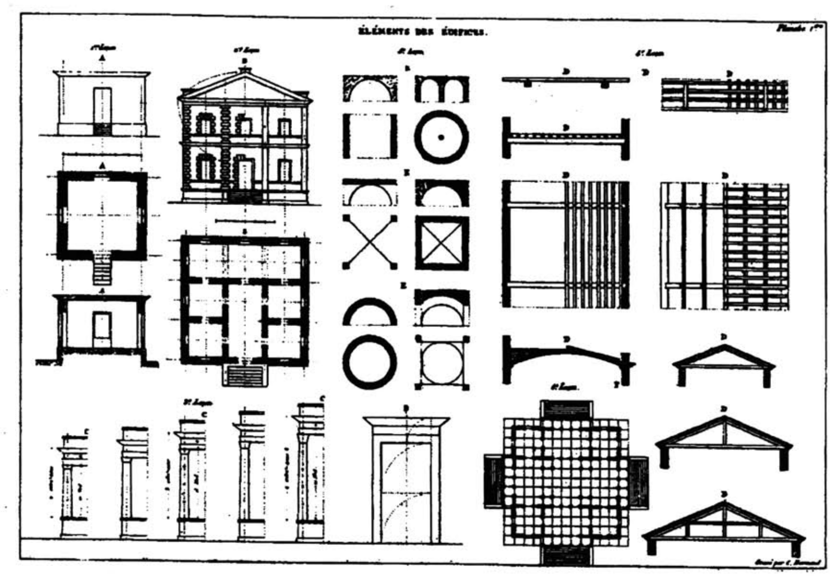


Figure 1.1.2 Building elements from Durand's *Partie Graphique,* 1821

was put forth by J.N.L. Durand shortly after Quatremere de Quincy's theory. Durand developed a highly systematic theory of typology, particularly about a vocabulary of architectural types derived from existing classical forms.[Figure 1.1.1] For Durand, to establish the basic types of building components is the point of departure for architectural design since these types can be found in any building, regardless of style, i.e., walls, windows, columns, and roofs. [Figure 1.1.2] Once the basic elements of architecture are defined, parties of buildings i.e., lobbies, lounges, and courts can be composed by the combination of the basic elements. Finally the integration of the parties results in producing a building.

According to Thomas Thiis-Evensen, types are comprised of 3 main parts: the floor [Figure 1.1.3], the walls [Figure 1.1.4], and the roof [Figure 1.1.5]. He writes that these can be divided into four categories: the elements' major forms, the construction system (massive or skeletal), the surface treatment of the major forms, and the openings in the major forms. Thiis-Evensen believes that on each of these four levels, fundamental formal types must exist to represent universal solution to problems of various forms that remain the same anywhere throughout time. To him, creativity depends on how these basic forms are combined and transformed.

William Mitchell , in *The Logic of Architecture* (1990), emphasizes the 'type' concept for the use of architectural vocabularies in the description of buildings. He says that *" formation of a type is an act of abstraction; we identify what is the same about all member of some class of objects and disregard what varies."* What he means here is that 'type' is an object of class abstracted by the elimination of idiosyncratic variations while maintaining common characteristics. Mitchell believes that defining 'type' is necessay for developing an architectural language because it provides *"a strategy for segmenting architectural compositions into parts and a way to refer to parts by name in order to develop a discourse about some body of architectural work."* Further, he believes that by classifying vocabularies into different types, a designer can establish the meanings of nouns, such as column, beam, wall, door, bedroom, and house so that he or she can organize and incorporate them into design knowledge.

When we develop design vocabularies, it is important to define 'type' vocabularies which serve as a basis for all later variations and combinations. These 'type' vocabularies are essential to the composition of an architectural

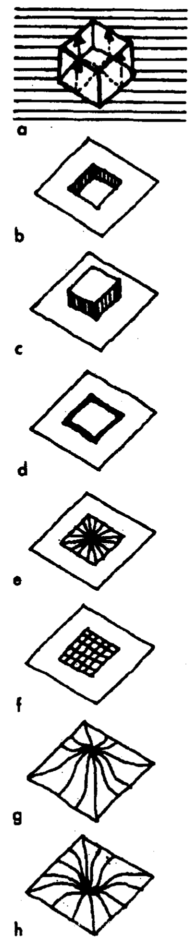


Figure 1.1.3 Types of floor: a) continuous, b) sinking, c) raising, d) framed, e) centralized, f)equal directionnal, g,h) undulating

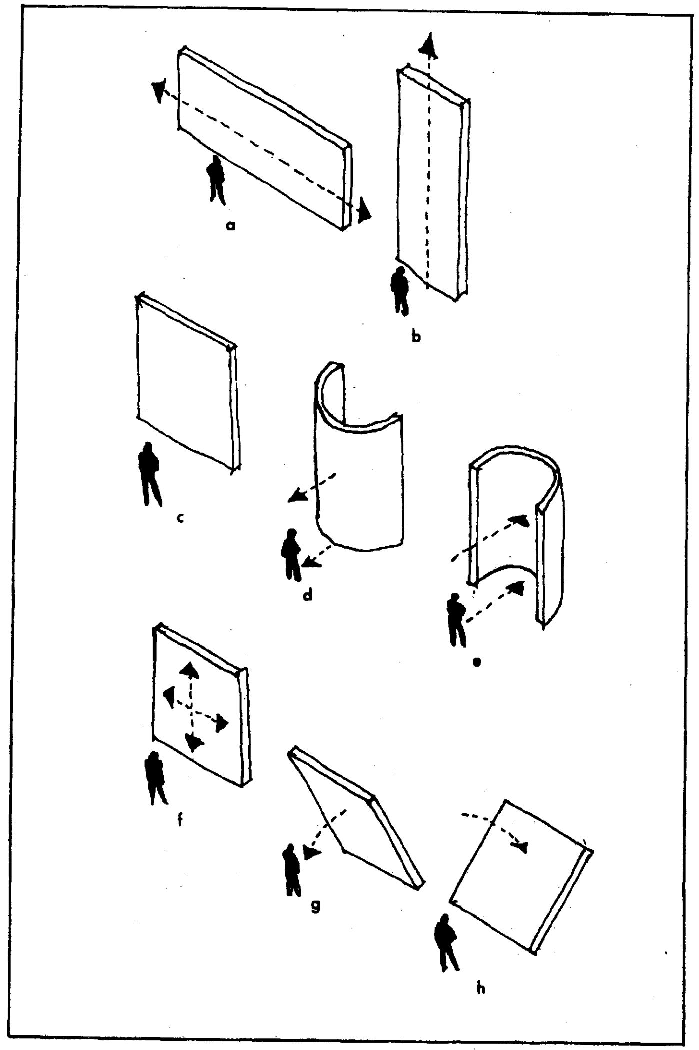


Figure 1.1.4 Types of wall: a) horizontal, b) vertical, c) flat, d) convex, e) concave, f) straight, g) leaning toward, h) leaning away

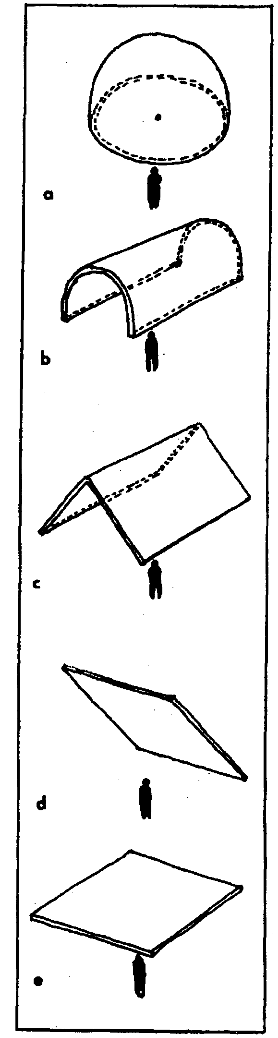


Figure 1.1. 5 Types of roof a) done, b) barrel vault, c) gable, d) shed, e) flat

order and adaptable to the plurality of the many conditions regardless of time and place. However, the use of type should not limit a designer's creativity. 'Type' is intended to establish a framework for a designer's process of thinking within which he or she can do change operations. Here 'change' does not mean to be arbitrary or transient, but to be contextual and programmatic. In the transformation process, the designer can create the subtypes from the type, and overlap different types to produce new ones. He or she can use a known type or create new types for new inventions depending on context and program.

1.2 Aldo Rossi: Shape and Rule

The Italian rational architect Aldo Rossi is known for his use of 'typing' concept in architectural design. His architecture is composed of types of objects - towers, steps, and corridors - that are used again and again in different contexts. Rossi always uses these same objects of design vocabularies in order to find something new for the composition of an architectural space. He generates new forms by reproducing what is already there with small modifications. Rossi himself says :

Looking at these projects all together, it seems to me that today I could still redesign them one by one: perhaps the final result would be different, if only because of small changes and shifts in proportion. [1]

[1] Peter Arnell and TeBickford, eds. *ALDO ROSSI: Buildings and Projects* (New York, 1984), p. 10.

The design vocabularies of Aldo Rossi's architecture are derived from pure, primitive forms, such as cube, cone, cylinder, and pyramid which are primary factors for structural and functional elements of his architecture [Figure 1.2.1]. The employment of such simple, stereometric forms resulting from analogies to historical building forms, plays a central role in the work of Aldo Rossi. Through reference to history as well as memory from personal experience, he tries to derive architectural elements that provide a new basis for design. At the same time, Rossi always endeavors to find "prototypes" for each particular use. This process of defining architectural prototypes provides him with a unique freedom of design; this freedom of design allows him to obtain different results by making the same thing over and over again. Basically he creates his buildings by repeated use of his own architectural prototypes which he transforms into new design vocabularies in response to site and

Origins of Prototypes

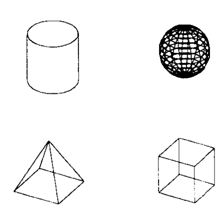


Figure 1.2.1

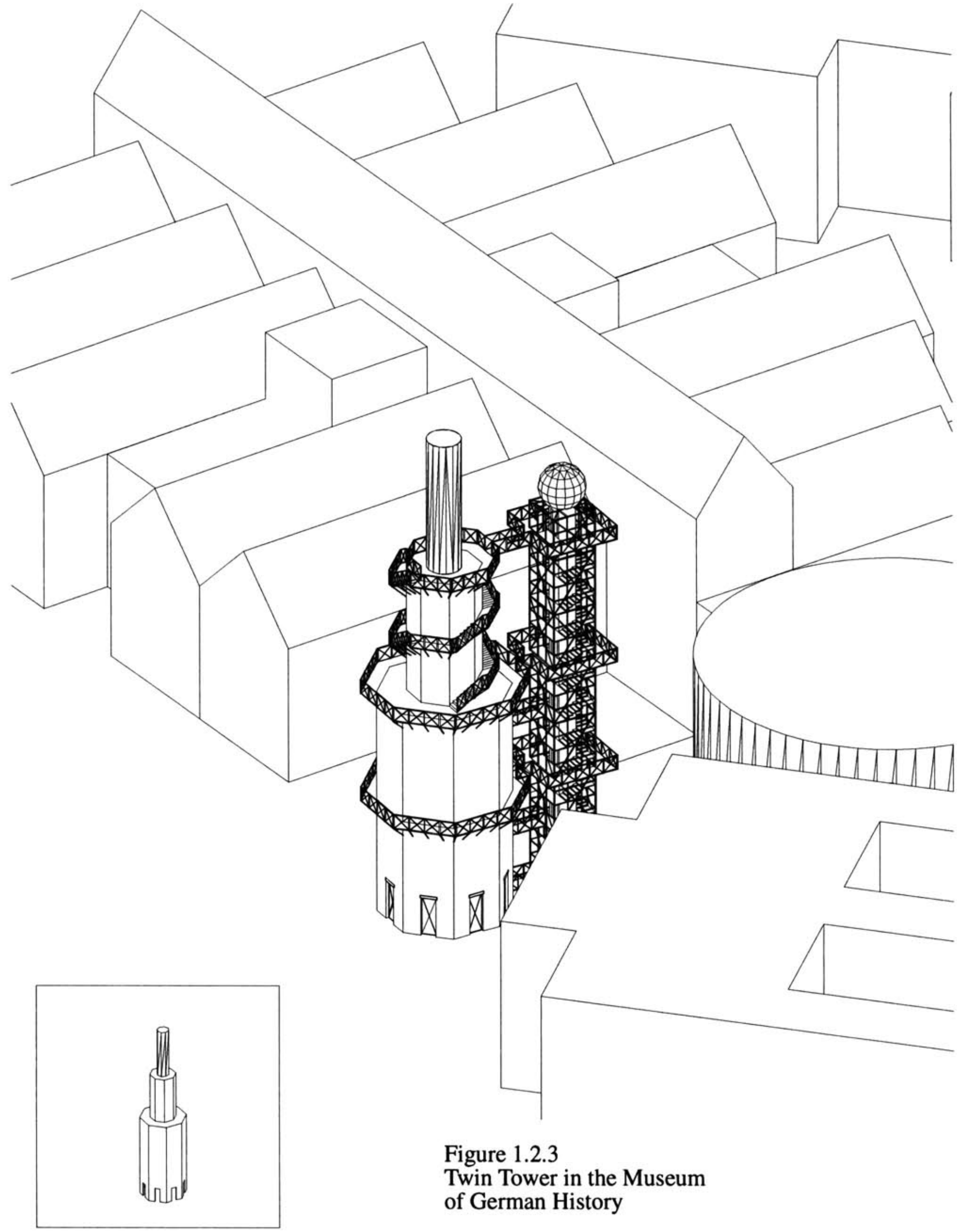


Figure 1.2.3
Twin Tower in the Museum
of German History

Prototypes of Vocabularies

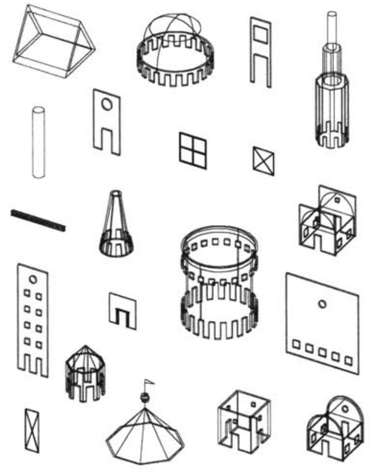


Figure 1.2.2

Transformations and Variations

1. Thesis: Cultural Center and Theater, 1959
2. Secondary School, 1979
3. Floating Theater, 1979
4. A Landmark for Melbourne, 1979
5. Entrance Portal, 1980
6. Tower in Lighthouse Theater, 1987
7. Yatai, 1989
8. Urban Monument, 1989
9. Union Bank of Switzerland, 1991

Figure 1.2.4

program. These new vocabularies evolve into a physical work of architecture through the transformation of scale, volume, and texture. Figure 1.2.2 shows Rossi's own various prototypes which are found in most of his projects.

A design element that can be analyzed to obtain a better understanding of Rossi's method of 'typing', is a tower in the Museum of German History. This tower, a competition-winning project in 1988[Figure 1.2.3], is part of the twin towers and is comprised of 3 volumes- two octagons and one cylinder. The two basic shapes(octagon and cylinder) and the tripartite composition of nesting one element on top of another are found in his design projects from the early years to the recent. Figure 1.2.4 shows variations of his tripartite forms in chronological order - how they were transformed into different shapes in different contexts throughout the years.

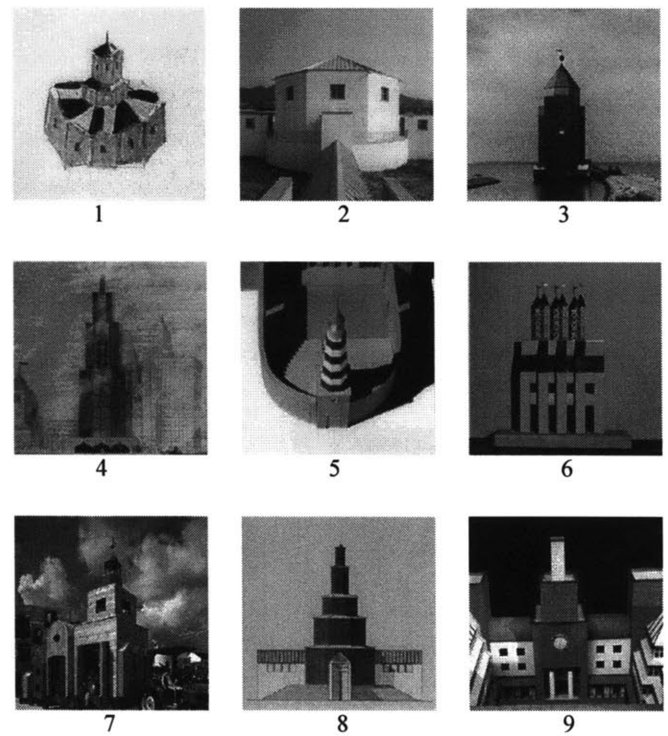
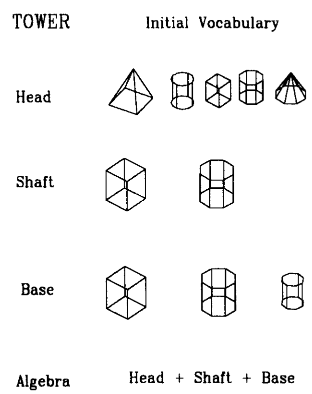


Figure 1.2.4 Transformations and Variations

[2] George Stiny states that a shape grammar consists of rules and an initial shape. The rules apply to the initial shape and to shapes produced by previous rule applications to generate designs. He views the generation of designs as the final production resulting from adding and subtracting shapes. According to his theory, a design begins with a single shape and a new design can be produced from the current design by adding a shape to it or subtracting a shape from it.

This phenomenon of shape composition depicts Aldo Rossi's unique methodology of designing a building. This methodology is a design knowledge of comprising various types of architectural elements by shape rules. Figure 1.2.5 is a diagrammatic analysis of shape composition which is conceptually analogous to Georgy Stiny's shape grammar: rules of addition and subtraction.[2] The rule of composition for a tower is the algebra of the sum of base, shaft, and head. Each component can have two subshapes as shown in example1, 5, 7, and 8. The Base has initial shapes of cube, polycube (generally octa-cube), and cylinder. These elements are changed into new geometries of forms by scale, proportion, and texture to be harmoniously assembled with shaft and head which also have their own prototypes. By composing and manipulating these forms with rules of addition and subtraction (replacement), Aldo Rossi created various types of towers with small shifts in geometry.



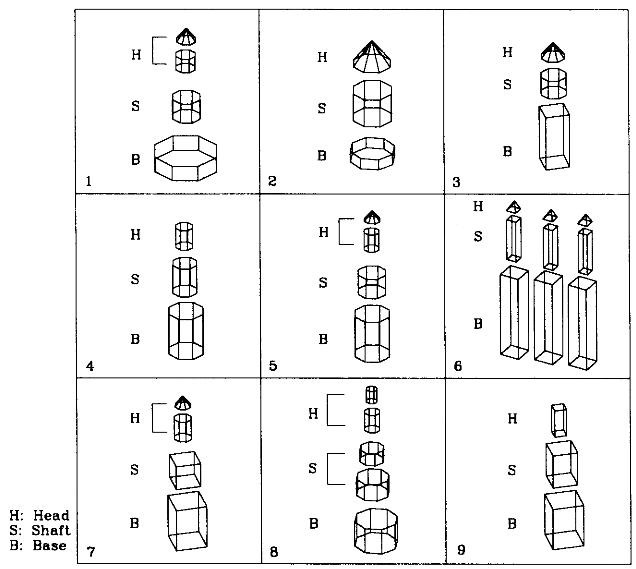


Figure 1.2.5

Rossi generates a typological form of architecture with shape rules of architectural grammar. He uses types of objects to plan his design projects, and composes them according to a simple rule of algebra: addition and subtraction. Even though his design methodology is a far cry from the actual computer application, it supports the idea that a design can be computed with shape information through transformation and combination operators. In his book *The Electronic Design Studio,* William J. Mitchell talks about a similar idea of shape computation. He believes that the most usual computational operations are transformations and combinations of shapes. Further, he believes that the computation can terminate when a designer recursively applies those operators to the initial vocabulary until certain predicates are satisfied by a shape produced by the operators.

Once design vocabularies are defined, we can manipulate them with few but powerful sets of rules, and combine them into a building with the algebra of addition and subtraction. But these vocabularies and rules should not limit the description of design (the creative design) , nor predetermine the final design. They must be open, flexible, and comprehensive so that they can be changed anytime when necessary.

2. Computation of Design Knowledge

Design is the making of a building through knowledge structure. In the process of making, designers collect, analyze, and implement their knowledges acquired from direct and indirect experiences. Robert Oxman and Rivka Oxman, in the article *The Computability of Architectural Knowledge* [1], explains that design knowledge is a collection of information and data necessary for integrating object descriptions with the formalization of knowledge in design. The design knowledges are useful when they can be represented as a form which employs its generic characteristics. Such a form is called 'Prototype'. The concept of prototype is the generalization of typological knowledge in the process of design. It helps to define proper knowledge of form for solving particular problems. In this process, however, there is the necessity of acquiring procedural knowledge of the modification of the prototype - 'Parametric Design' [2] as well as compositional technique for controlling the knowledge of the procedures - 'Shape Grammar'.

[1] McCullough, Malcolm. et al. *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era.* (Cambridge, 1990), pp. 187-199.

[2] John Gero calls it 'Routine Design' in the article *Structural Design Knowledge as Prototypes.* For the computation of design, he states that types of design should be differentiated into 3 categories: routine design, innovative design, and creative design.

2.1 Shape Grammar

The rules of a grammar are not limiting prescriptions, but tools for constructing a path from the known to the unknown—tools that can be changed if they do not seem to get you to the right place.

William J. Mitchell, The Electronic Design Studio, 1989

Grammars have been used in programming languages to design algorithms. They facilitate the compilation process by simplifying computational complexities. In architectural domain, shape grammar is developed to determine how elements are combined to produce the form of an artifact by a set of rules. George Stiny [3] explains that shape grammar defines parts of shapes with a set of rules and changes the shapes recursively to conform to given spatial relations. He believes that a shape grammar is made up of an initial shape and a set of rules; formal computation starts with a rule applied to the initial shape, and then other rules are applied to a series of shapes generated by previous rule applications until desired spatial conditions are satisfied. Figure 2.1.1 shows an example of a shape grammar application. There are two initial shapes-- a small square and a large square. Two of four rules are addition rules and the others subtraction rules [4]. They are created

[3] Stiny, George. "*Computing with Form and Meaning in Architecture*," JAE, Fall, 1985, p. 7-19.

[4] In his article *What Designers Do That Computers Should* from the book *The Electronic Design Studio,* Stiny believes that shapes can be manipulated by sum and difference with transformations which are the underling concept of shape grammar. He explains that the algebra of sum and difference allow for shapes to be decomposed in any situation and make it possible to replace the traditional design (drawing and erasing) with the computational design. The sequence of algebra operations is defined by the following transformation rule:

$$C$$
$$A \rightarrow B$$
$$t(A) <= C$$
$$(C - t(A)) + t(B)$$

C is an initial shape. *A* is replaced by *B* if transformed shape *A* - *t(A)* is part of *C*. The final result comes from subtracting *t(A)* from *C*, then adding transformed shape - *t(B)*.

[5] Stiny, G. and Mitchell, W.J. *"The Palladian grammar,"* Environment and Planning B 5:5-18, 1978. The analysis of paper is based on *The Four Books of Architecture* by the Italian Renaissance architect, Andrea Palladio in 1570.

by spatial relation of nesting one square on another. By applying these rules to the initial shape, one can generate different types of interesting nested-squares.

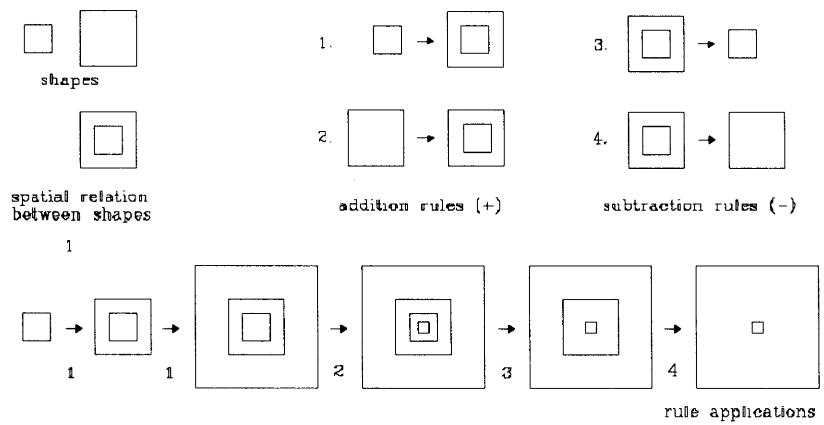


Figure 2.1.1 Shape rules of addition and subtraction

Shape grammar has been used to define an architectural language. One of the most well-known analyses was the Palladian grammar presented by Stiny and Mitchell [5]. They investigate Palladio's villa ground plans to define Palladio's architectural grammar – a Parametric Shape Grammar. The Villa Malcontenta is chosen to describe the parametric shape grammar with a set of composition rules [Figure 2.1.2]. They think that there are six sets of rules to generate the villa plan: 1) grid rules , 2) wall pattern rules, 3) room layout rules, 4) portico and wall inflection rules, 5) column rules, and 6) window and door rules. It is their belief that the power of the shape rules is in generating numerous villa plans of Palladian type [Figure 2.1.3]. They successfully formulated the Palladian grammar as a parametric shape grammar with illustrations of a step-by-step transformation of the Villa Malcontenta plan.

There have been numerous studies of architectural styles by shape grammar, but few actual explorations of design. This may result from not only a lack of interest in shape grammar among designers, but also negative and passive attitudes toward computer use in the design process. However, shape grammar become more acceptable to designers as CAD (Computer-Aided-Design) systems are more available and easy to use. The nature of the computer requires the logic

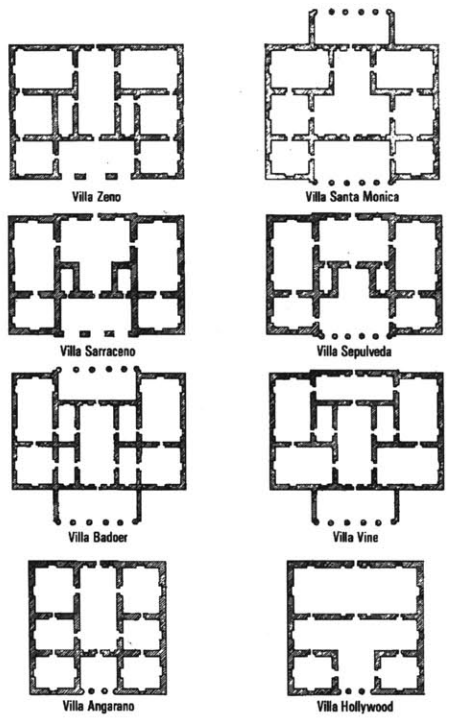


Figure 2.1.3 Palladian villa plans generated by six sets of parametric shape grammar

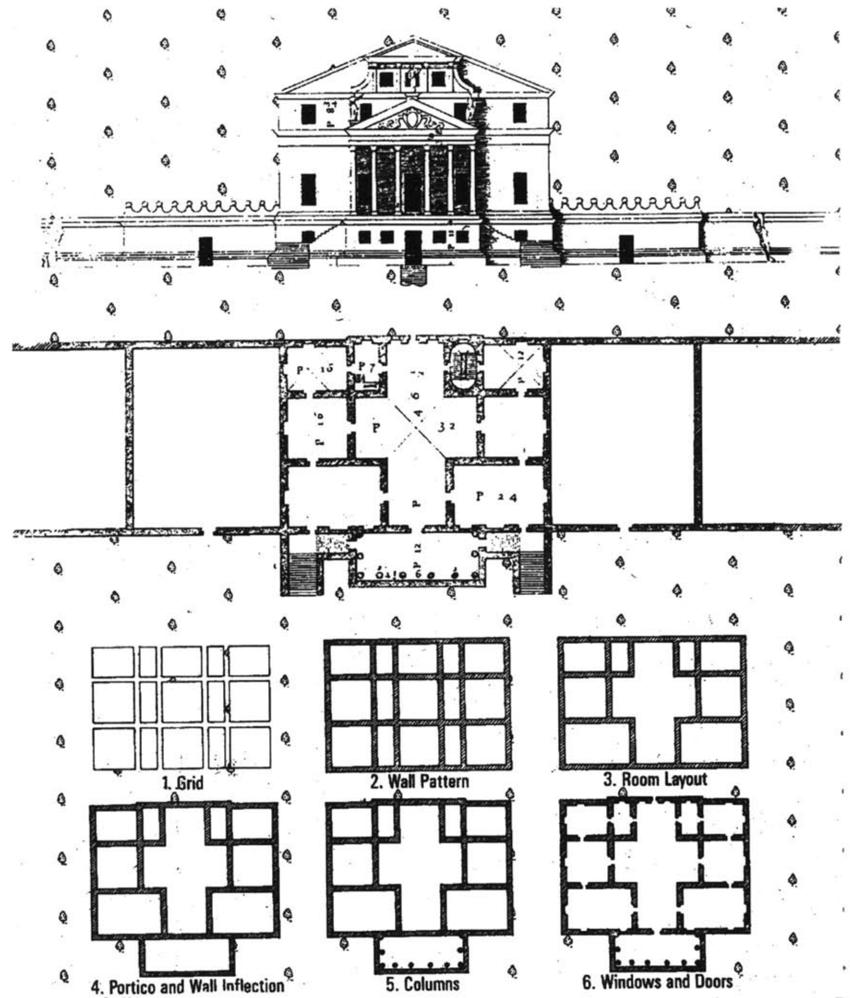


Figure 2.1.2 Six set of rules to generate the plan for the Villa Malcontenta

of shape grammar. In order to use a computer in design, one must know the coordinates of point to draw lines, the geometrical relationship between lines to construct planes, and the spatial inter-dependence between planes to make volumes. Manipulating these coordinates and relationships does not satisfy the needs of man. One must know how to manipulate them for building humanly-supportive spaces. In this process of constructing lines, planes, and volumes, a designer needs a shape rule to place lines, planes , and volumes in a CAD environment. Without a defined knowledge of architecture—shape grammar, using computers in design will be no more than automatic production by a frozen mechanism [6]. Initial shapes with a set of flexible rules will guide designers to get to the right track of design processes. This will enable them to simplify their thinking processes from complex design issues, speed up the design process, and come up with a better design.

[6] Moneo, Rafael. *"On Typology"*, Oppositions 13, 1978. Moneo talks about the concept of 'type' as an important aspect of design in architecture. He believes that types are intended to play a role of instrument with a capability of coping with diversity of forms. Further he believes that the use of 'type' does not mean the mass-production by a frozen mechanism; types should be repeated and transformed to accommodate new conditions of program and context.

2.2 Parametric Design

By definition, a parameter is a constant whose value varies with the circumstances of its application. The process of changing the parameter's values is called 'parametric variation'. Parametric design is a computational method of using parametric variation for creating various sub-shapes from one shape (prototype) to fit into a certain design environment through the implementation of a computer program. In the architectural realm, the parametric design has been studied for form-generation and spatial exploration. It can be a useful design method, particularly, in the design process when a shape has to be changed to fit into different spaces in different situations. By assigning specific values to the parameters of a shape, it can be transformed into a particular geometry of form for a particular situation.

There is a consensus among CAD researchers that parametric design has been successfully applied in many areas including architecture. In *Classes of Design - Classes of Tools* (1990) [7] Gerhard Schmitt states that parametric design assumes the following three conditions: 1) *The design problem is well defined and client requirements are well understood*; 2) *There exists a parameterized prototype for the type of design that is to be developed and a data base of parameter variations*; 3) *The final design may be derived by refining, but not fundamentally changing the prototype.* He states further that the parametric

[7] McCullough, *op. cit.*, pp. 77-90

design is useful in phases of schematic design, preliminary design, design development, contract documents, and shop drawings except for program development and construction drawings.

Parameterized shapes are advantageous for their adaptability to many different conditions. For example, a parameterized window can fit into any opening of width and height inside a wall. Once the window is parameterized with a programming language, such as Autolisp, it can be reused again and again by simply inputting different values of parameters for width and height whenever sizes of window openings change. This flexible capability provides a designer with a wide range of alternative forms in the process of making a building which ,in the end, helps him or her to come up with a better architectural solutions.

There are two advanced approaches to parametric design: top-down and bottom-up design methods. They have a common characteristic in that they deal with more than two shapes at once and the parameters of one shape are dependent on the parameters of the other. For instance, if you change a parameter of one of the shapes, the shape with the changed parameter not only transforms itself into a new form of geometry, but also affects the other shape so that the latter changes in relation to the former. On the other hand, the difference between top-down and bottom-up methods is the structuring sequence of parametric shapes. In top-down design method, smaller shapes (sub-shapes) are parameterized to fit within a larger shape so that they can be correlated properly when any of the shapes changes. Take a classical column as an example. In general, a classical column consists of a base, a shaft, and a capital. If a parameter of the base is changed, these sub-shapes have to be parameterized with correct coordination of dimensions and proportions. The geometry of the shaft and the capital as well as the base itself, are affected while the overall figure of the column is maintained.

On the other hand, the sub-shapes do not have to be parameterized within a larger frame of shape in the bottom-up design method. They are parameterized independently in the beginning, but they are adapted , when necessary, to create a desired shape. In this bottom-up fashion, the base, shaft, and capital are independent elements. They can be used in a combination of pieces that achieves any desired result. A combination of a base and a shaft or a base and a capital can be adopted to create different kinds of shapes in different contexts. There is a general consensus that top-down and bottom-up design methods are complimentary. When some shapes are inter-dependent, they can be parameterized to fit within a geometry resulting from their combination in top-down fashion. Otherwise sub-shapes are designed using the bottom-up method to be compatible with any other shapes parametrically so that their combination can produce a desired shape for a particular situation.

As design becomes more complex, different conditions at different stages of design demand different characteristics of architectural shapes. In this respect, parametric design with top-down and bottom-up design methods can be an efficient method of manipulating and transforming shapes and provide design alternatives that can rapidly respond to the changing needs of design situations.

2.3 Knowledge-Based System

Computer programs, in which human knowledges are represented in symbolic forms, are called 'knowledge-based systems'. They are based on a relationship between types of structured knowledge. From the architectural point of view, the knowledges are encoded as pre-defined structures which are delivered to the user (designer) through the computer program. The knowledge-based systems are concerned with the manipulation and modification of design knowledges , while they provide assistance to designers in the process of decision-making.

One of the best-known knowledge-based systems is *Top-Down* developed by William Mitchell, Robin Liggett, and Milton Tan.[8] They demonstrate that the knowledge system can be useful to a designer when he or she develops a particular element or subsystem of a building. The system has two graphic windows – the peek window and the poke window. [Figure 2.3.1] The peek window displays the design state in use; the poke window shows the result from the input in the peek window. As can be seen in Figure 2.3.2, one can choose the desired object in a dialog box to perform a transformation. Actual parametric variation can be accomplished by using the slide bar in the poke window. [Figure 2.3.3] Mitchell, Liggett, and Tan use classical columns to illustrate the operation of *Top-Down* with several types of capitals, shafts,

[8] Mitchell, William J. et al. *The Art of Computer Graphics Programming: A Structured Introduction for Architects and Designers.* New York: Van Nostrand Reinhold, 1987. The book explains extensively the top-down method by PASCL program.

Figure 2.3.1
Peek, Poke, and
message windows

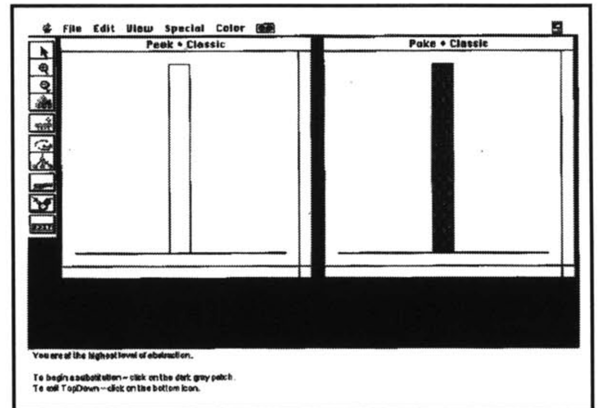


Figure 2.3.2
Transformation
in Poke window

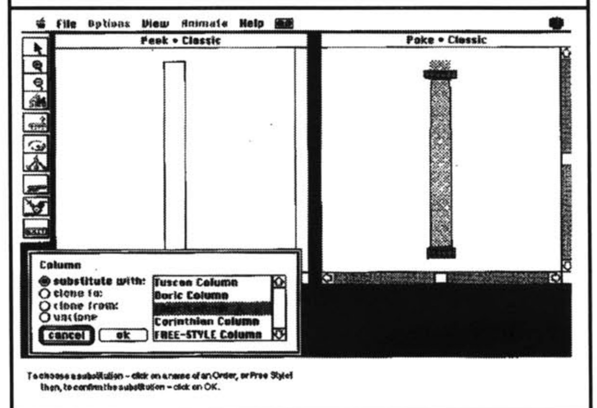


Figure 2.3.3
Parametric variation
in Poke window

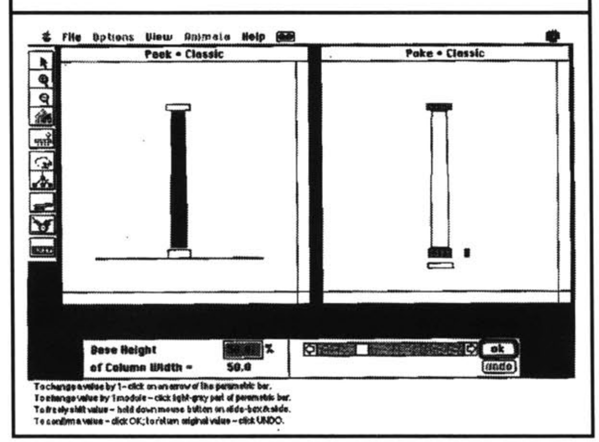


Figure 2.3.7
Variations of out-of-proportioned columns

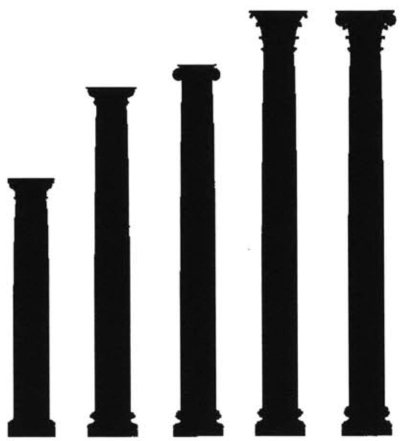


Figure 2.3.6
Variations of proportioned columns

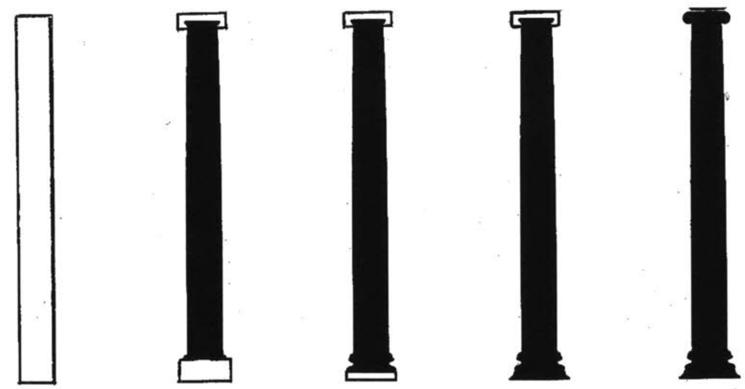


Figure 2.3.5 Steps in elaboration of an Ionic column

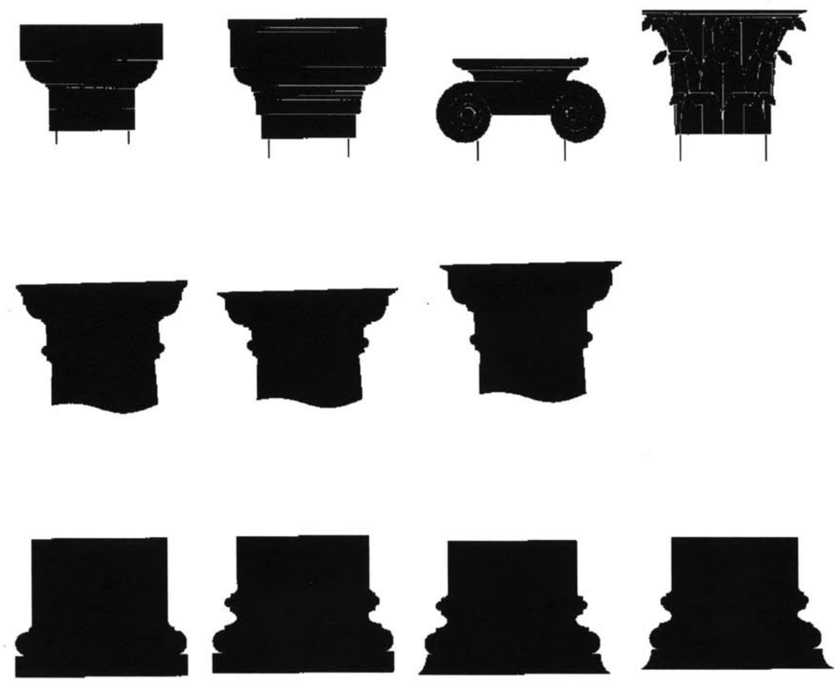
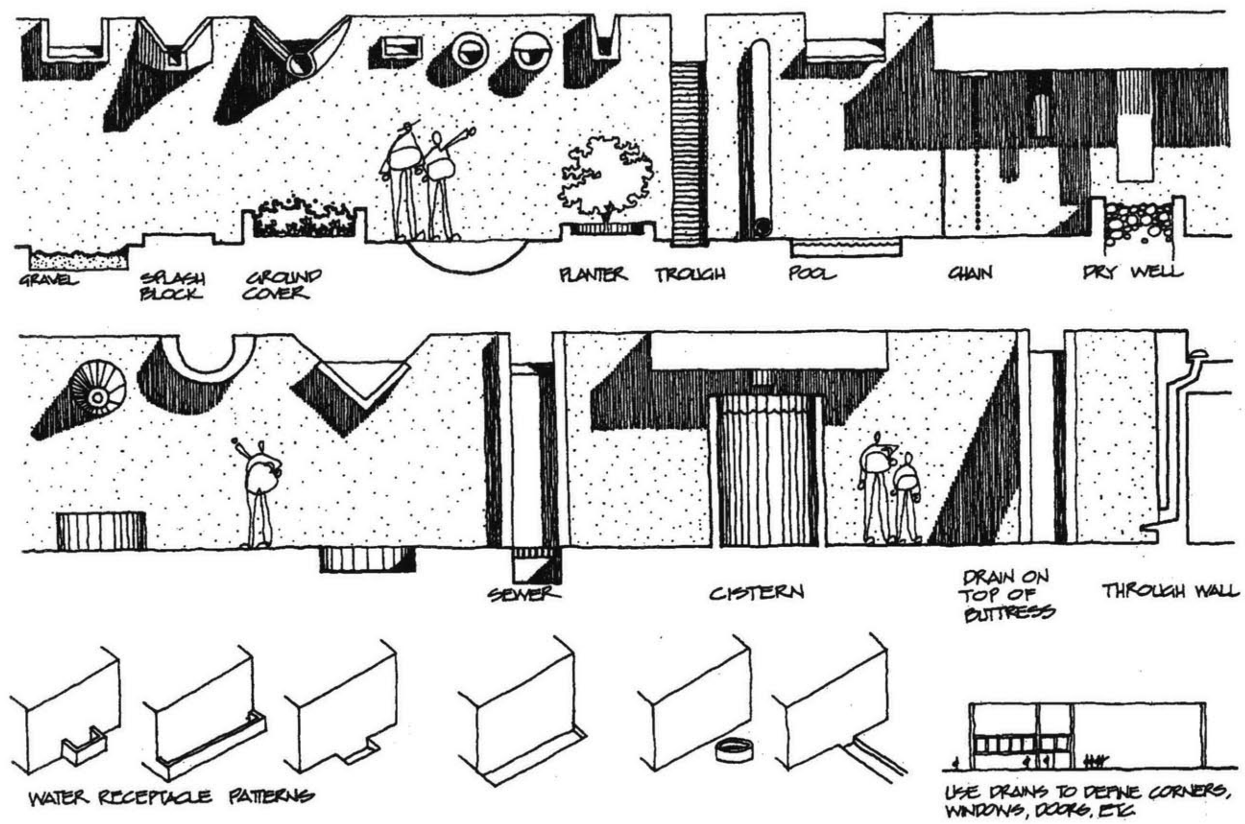


Figure 2.3.4 A vocabulary of capitals and bases

and bases. [Figure 2.3.4] Then they show how a long rectangle can be transformed into an classical Ionic column as shown in Figure 2.3.5. Through such an operation, the knowledge system can create various classical columns [Figure 2.3.6] as well as interesting geometries of objects [Figure 2.3.7].

The *Top-down* knowledge system demonstrates how an object can be manipulated to get a desired shape in a top-to-bottom fashion, and how a knowledge encoded by shape grammar can be implemented to perform a design operation. But Mitchell, Liggett, and Tan do not explain how the system can be actually applied to the design process. Furthermore there is a lack of examples of architectural vocabularies, such as doors, windows, roofs, and walls. Most of them are two-dimensional and abstract. However, while there are some limitations to the system, the top-down system seems to provide great potential for the future development of knowledge-based systems.

The use of computer requires that a design knowledge be encoded to represent and symbolize architectural forms. The computation of design necessitates the formalization of the design knowledge – 'shape grammar' to construct compositional structures. By generalizing and classifying design knowledges – 'prototyping knowledges', one can establish a link between design and computer, thus executing manipulative operations of architectural forms. The parametric design provides a wide variety of flexible tools for the transformation of the architectural forms in various programs and contexts. With the theory of shape grammar and the computation of parametric design, the design knowledges can be used effectively and creatively with the talent of a designer.



Vocabularies of canales and water bins from *CONCEPT SOURCEBOOK: a vocabulary of architectural forms* by Edward T. White

The buildings I want to design are assemblages of forms put together to create a figure reminiscent of synthetic cubism. They are composed of individual forms joined together as a single architectural identity by means of shape rules. Architectural order is created by applying rules of a grammar recursively to these forms until they are successfully linked to the aspect of place and the need of man.

Lee Hee Won, The Computation of Design Vocabulary, 1997

3. Investigation and Exploration

Figure 3.1
Rossi's sketches

This thesis investigates and explores the idea of using computer in design process based on Aldo Rossi's 'type' concept of design vocabulary. [1] Two composition techniques -- top-down and bottom-up methods -- are evaluated to determine the advantages and disadvantages of both techniques through actual design explorations in an effort to answer the question of how a computer can be used efficiently and creatively in design. By the design exploration, the role of shape grammar is also investigated to test its possible application in the design process away from the conventional theory and analysis. Furthermore, the top-down and bottom-up methods are encoded through a program to demonstrate their capability of parametric variation. Up to today, a program has been one thing; design has been another. There has been no inter-dependent relationship between them other than analytical kinship. For this reason, two program examples of building components are analyzed from my design experience and illustrated as if they were in actual design situation. This investigation and exploration is an endeavor to learn how a designer can use a computer in design process, but not a story to tell how a computer can support a designer in design process.

3.2 Experimental Design

Based on Aldo Rossi's 'Type' concept of design vocabulary, two design projects were experimented for using a computer in the design process from the conceptual idea to the final production. Each design was executed with the following three questions: 1) Is it possible to use a computer from "scratch to finish"? 2) Which design method is better in design process-- top-down method or bottom-up method? Or do they compliment each other? 3) Is it possible to lay a computational foundation for an architectural principle? Due to the experimental nature of this process, mistakes were to be expected.

3.2.1 The Summer Academy

This project was executed in Antonakakis studio (Fall, 1995). The design approach was carried out in a top-down fashion. This was my first attempt to use a computer in design process from "start to finish".

[1] Figure 3.1 shows Aldo Rossi's sketches through which he repeatedly used his vocabulary elements to generate ideas and create architectural spaces.

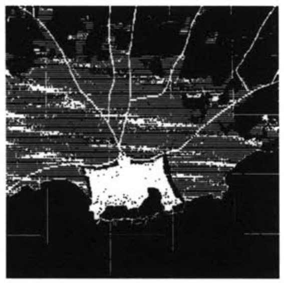


Figure 3.2.1.1 Site

[2] Figure 3.2.1.2 shows my own vocabulary called 'Cubic Vocabulary' developed for design experiments. I admire architectures of Richard Meier, Charles Gwathmey, O.M. Ungers, Tadao Ando, and Aldo Rossi. The conceptual scheme in Figure 3.2.1.3 was generated from cubic vocabulary elements. When I generate the vocabulary elements, I consider the architectural forms of five architects as 'prototypes' of my vocabulary. Then I try to derive my vocabulary elements from them by analyzing their geometry and sometimes borrowing some of their vocabularies.

[3] From the beginning, it was my intention not to use a physical model to experiment with the use of computer in design process, but the studio requirement hindered the computational process.

The design goal of the Summer Academy was to promote architecture and its relation to other arts and cultural activities in a cultural environment. The site is located at Chanea, the north-western part of Crete, in Greece, in a recess of the Bay of Chanea flanked by the Akrotiri peninsula and the cape of Spathi. [Figure 3.2.1.1] The building of the Summer Academy was to be built on the queue of the old harbor. The program includes four main spaces: administration, teaching Area, performance area, classrooms, and guest rooms. The total area was about 25, 000 square feet.

The parti resulted from understanding the nature of program and context. The idea was to create a courtyard scheme to provide a variety of public spaces by placing the courtyard scheme in the site. The public spaces include small alleys, a man-made plaza, and open space to be left as it is. The program required that the museum including theater, exhibition space, and reception function independently from the school. The solution was to separate two major programs which are connected by courtyard to promote the inter-activity between the school and the museum.

Initially my design vocabulary [2] was used to generate the courtyard scheme, but the scheme had to be given up because no compositional rule could be developed to manipulate the vocabulary elements due to my lack of experience in computational design. Geometric volumes were created in response to the program requirement. [Figure 3.2.1.4] They were thrown into the site to create a courtyard scheme with scale to a surrounding environment as can be seen in Scheme 1. Then the physical model in Scheme 2 of Figure 3.2.1.5 was made to represent the overall scheme three-dimensionally [3]. This led me to go back to the computer modeling to articulate each volume in detail. From this stage, the problem of the top-down design method became evident as geometries were complex. It was difficult to understand the inside space because of the overlapping of lines. It took a lot of time to see the rendered image which still did not reveal the quality of the interior space. Another problem was that a new form had to be redesigned from "scratch" whenever a new design state was changed. This was not an efficient method. The biggest problem was to communicate ideas with the studio instructor. The abstract computer model as seen in Scheme 3 did not convince him because I was unable to represent design ideas with a three-dimensional model. After Scheme 3, I had to depend heavily on two-dimensional drawings [Figure 3.2.1.6] to communicate ideas better as well as to satisfy the studio requirement with a series of superficial physical models. But the

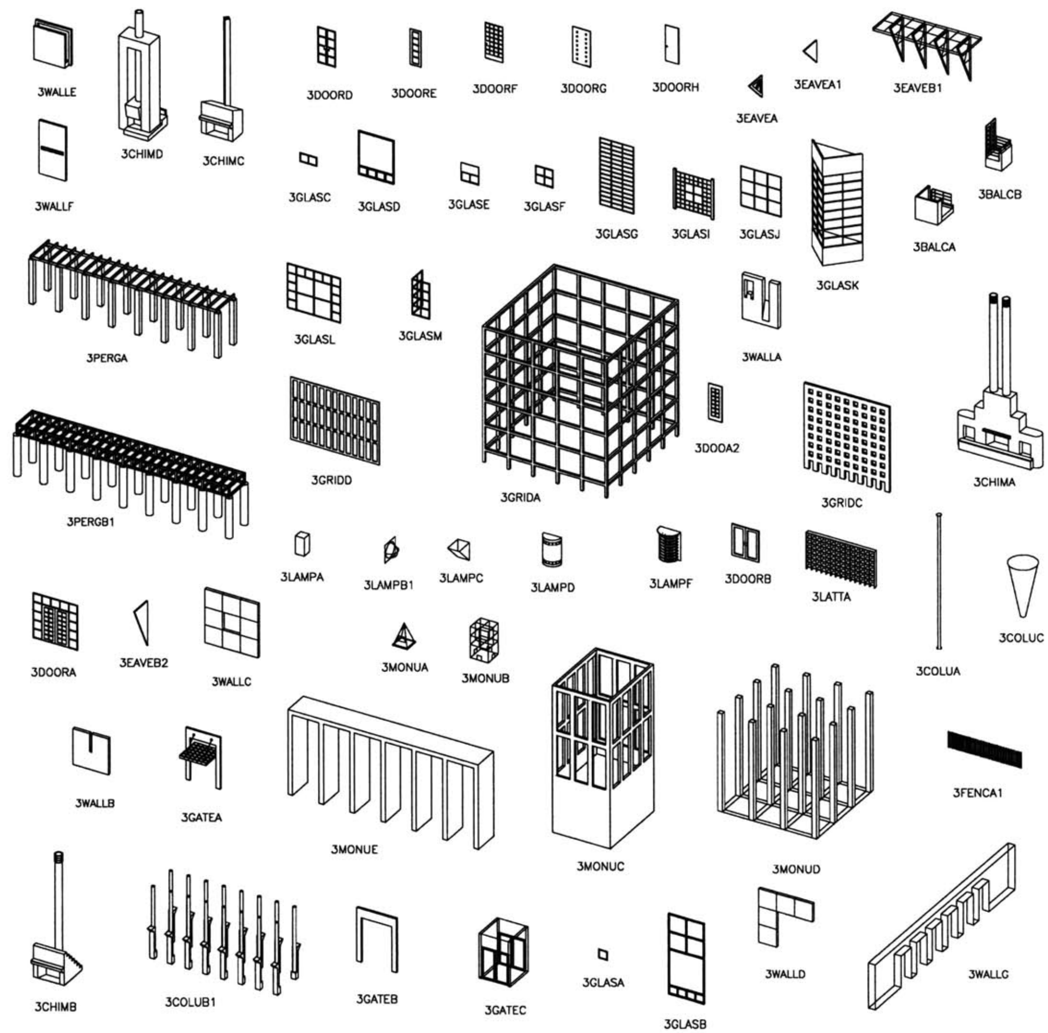


Figure 3.2.1.2 Cubic Vocabulary

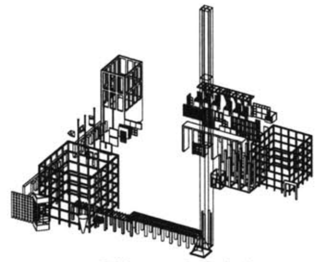


Figure 3.2.1.3
Courtyard scheme
from cubic vocabulary

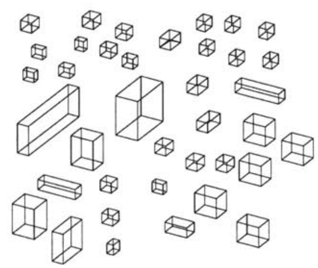
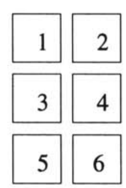


Figure 3.2.1.4
Geometric Volumes



Schemes

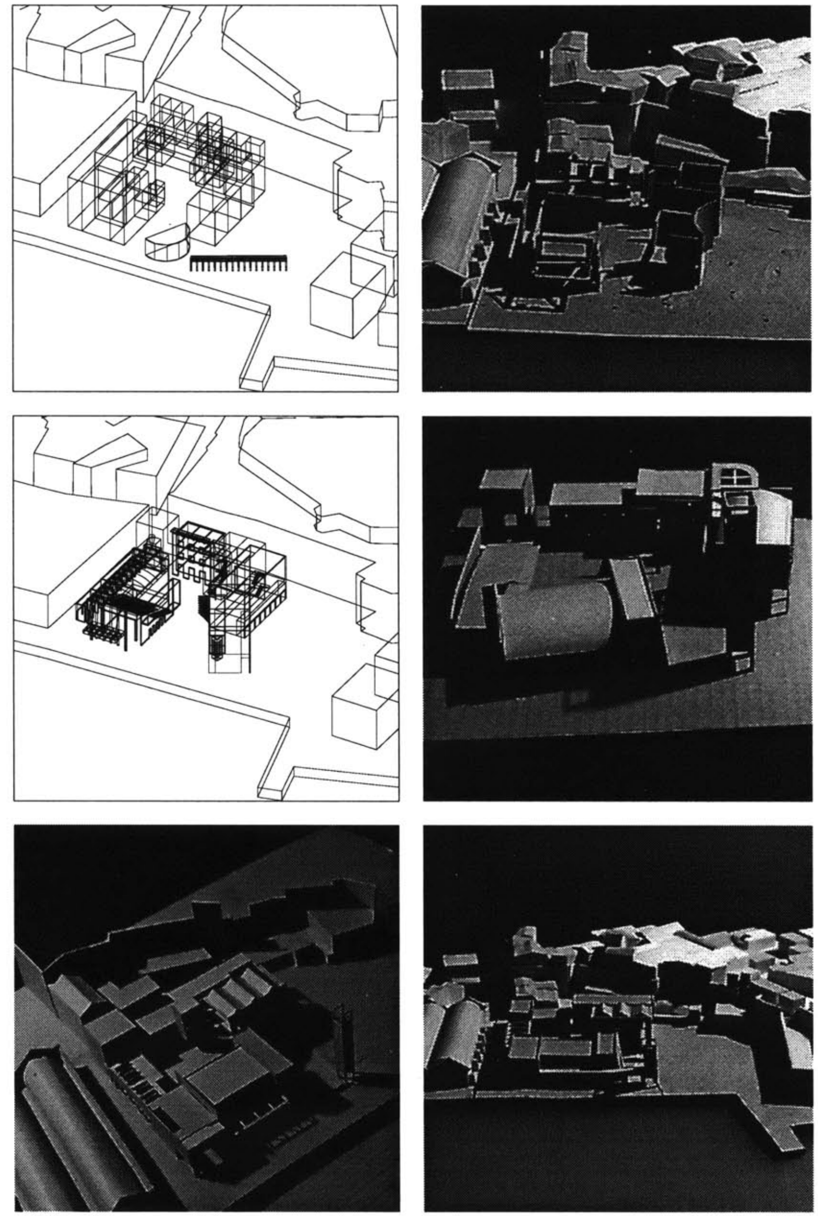


Figure 3.2.1.5 Development of Schemes

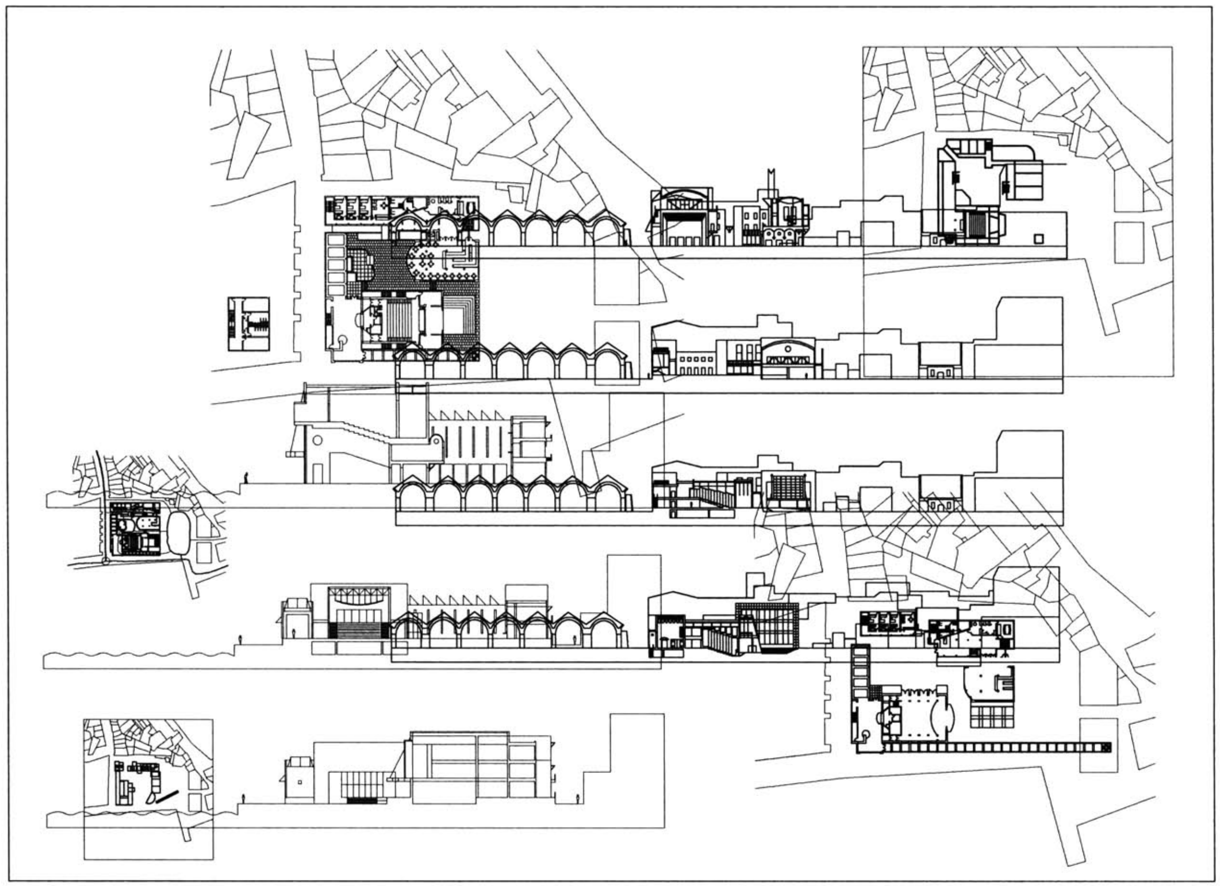
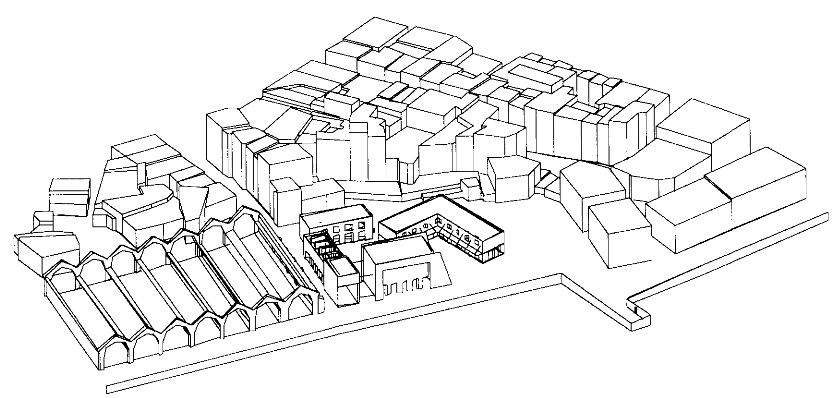


Figure 3.2.1.6 Sketch of plans, elevations, and sections

process of drawing plans, sections, and elevations as well as making the physical models was not helpful for understanding and exploring the dynamic space between objects.

Top-down design is supposed to start with a set of rules to articulate vocabulary elements, but this rule of thumb was not used in this project due to my present lack of experience in using computer in the design process. This seemed to result in the unsatisfactory design process and product. Thus it might be not fair to say that the top-down design is not an efficient design method. However, from my limited experience, the method seems not to be a good design tool for exploring the dynamic, flowing space between three-dimensional volumes. Furthermore to start from scratch whenever a design state is changed, seems to hinder the continuous design process with other design conditions.



3.2.2 The Visitor Pavilion

This project was executed in the William Mitchell/Andrew Scott studio (Spring, 1996). The design approach was carried out in a bottom-up fashion. The intention of the project was to engage in the act of design using a computer as a medium of exploration. The primary focus was on understanding the role of computer-controlled prototyping and fabrication in the creative act of design and making form.

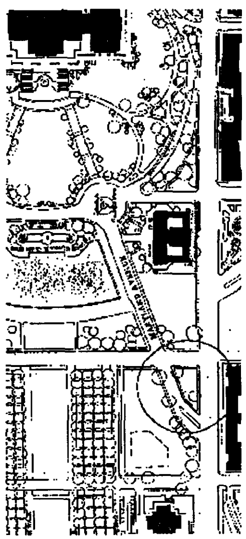


Figure 3.2.2.1 Site

The pavilion is a Visitor pavilion celebrating sports in America. It will be located in Washington, on Maryland Avenue close to the Capitol Building. [Figure 3.2.2.1] It will provide up-to-date information about sporting events as well as enable visitors to learn about American Sports. The program requires a public space, information center, exhibition space, space for audio visual display, and lockable storage totaling 4000 to 6000 square feet. The Visitor Pavilion is a permanent structure , but is only intended to be used seasonally, from May through September, so that it will not be heated or cooled. However, it will rely on passive based ventilation and cooling together with a self sufficient energy source-photovoltaic systems.

The design concept was derived from considering two major program issues-fabrication of components and control of natural lights. Initially 'a kit of parts' idea was developed to investigate how building components can be manipulated for prototyping and fabrication. Their architectural forms are generated from responding to the program and site. Then to organize and articulate the space for controlling natural light, the 'Five Faces' concept was introduced; east face was created for admitting the light in half and rejecting in in half; west face for denying the light; south face for shutting off the light but allowing indirect light; north face for celebrating the light; and finally, sky face for welcoming the sunlight to utilize the photovoltaic panels. Aesthetically these elements meant to create five distinctive facades, but the challenge was to assemble them harmoniously as one building entity.

Inside space was divided into two zones by a wide bridge which functions as both gallery and circulation path. One is a space for celebrating light, serving as an entry and lobby; the other is a space for blessing darkness where major artifacts are displayed. The entire circulation system was designed to experience four different walls so that the circulation starts with the northern wall

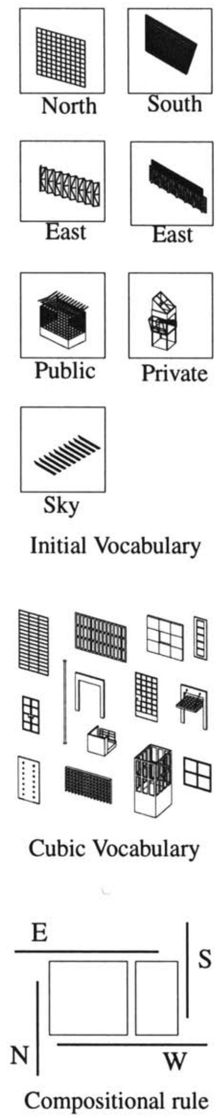


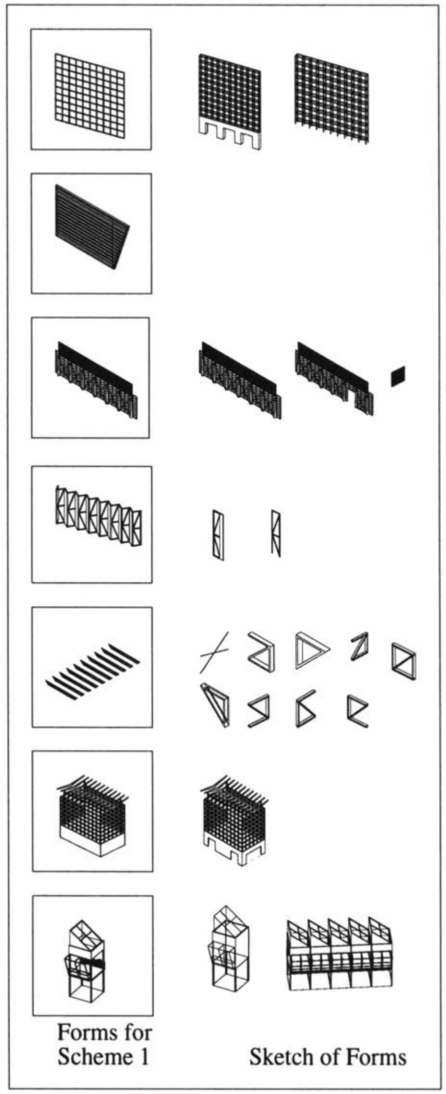
Figure 3.2.2.2

(leading visitors to the second floor), moves to the eastern wall, crosses the major pavilion space to arrive at the western wall, and then ends with southern wall (taking visitors to the ground level). Spaces under the bridge accommodate a video display area, office, restrooms, and storage. Steel was used for structural components- columns and beams; aluminum was used for non-structural elements, such as window, exhibition panel, and fins for shading. The ground floor was made of black marble so that it could reflect surrounding components, thereby casting interesting shadows with a combination of direct and indirect lights.

At the initial design stage, 4 distinctive wall elements, a roof form, and two interior volumes were created with some of my existing vocabulary introduced. A compositional rule was set up to manipulate and organize the vocabulary elements. [Figure 3.2.2.2] The first scheme of the project was composed by the intial shapes based on this composition rule. [Figure 3.2.2.3] After this scheme, there were three computatinal transformations as can be seen from Figure 3.2.2.4, Figure 3.2.2.5, Figure 3.2.2.6). In this process of making a building, the shape rule of 'addition and deletion' was applied for the transformation of the vocabulary elements. Figure 3.2.2.7 shows how initial shapes are manipulated and transformed by adding new shapes and deleting old shapes in response to changing needs of the design states.

The bottom-up design approach which I attempted here is not a perfect demonstration, but an investigative experiment. There was some inconsistency in the way vocabulary elements are manipulated and replaced. Some initial shapes maintained their original characteristics of geometry with some changes in scale, proportion, and texture throughout the design development, but other shapes were discarded to meet the changing design conditions with the generation of new shapes. Figure 3.2.2.8 show numerous sketches of building components in the design process; some of them were gone and others survived in the final stage of design.

Through this experimentation, the bottom-up design approach was found to be an efficient method of exploring form and space to take advantage of the computer capability (that is, inserting, deleting, moving, rotating, and copying). From the beginning, a few objects were manipulated to compose a space by moving and rotating them, and to build up complex spaces by the simple functions of inserting and copying with deleting unnecessary objects. The bottom-up method did not seem to pretermine the final form of design unlike the



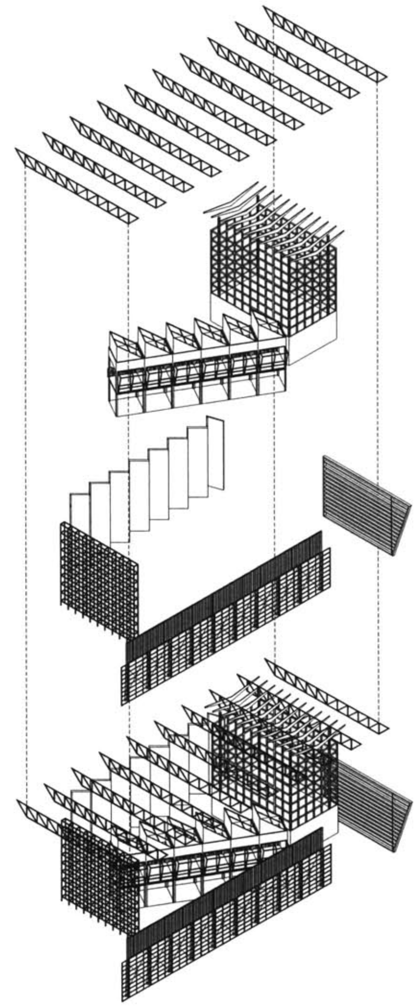
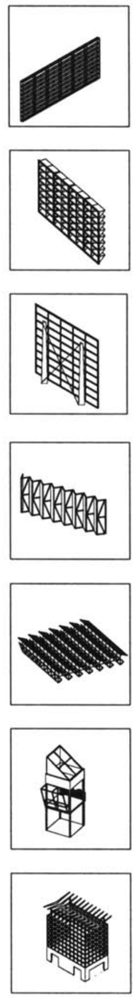


Figure 3.2.2.3 Scheme 1



Forms for
Scheme 2

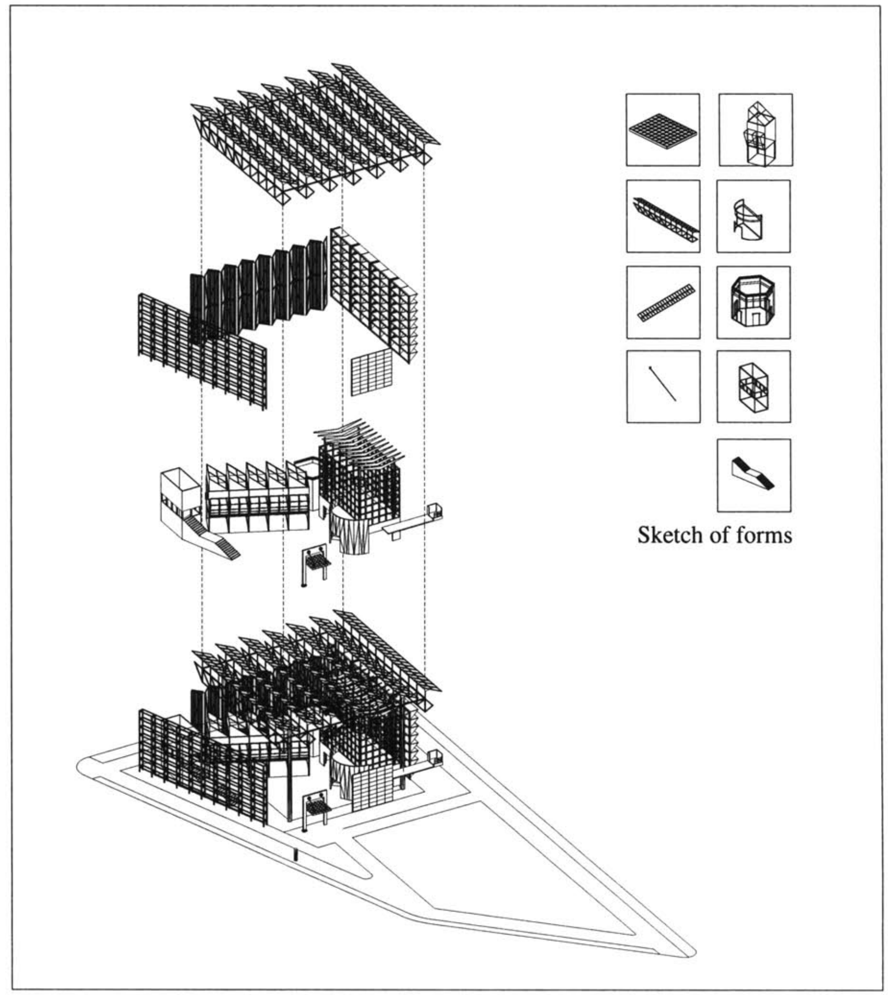
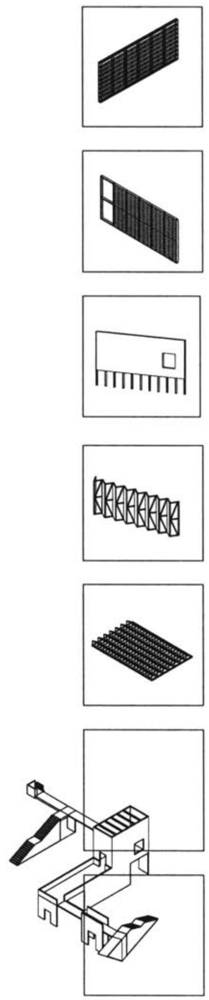


Figure 3.2.2.4 Scheme 2



Forms for
Scheme 3

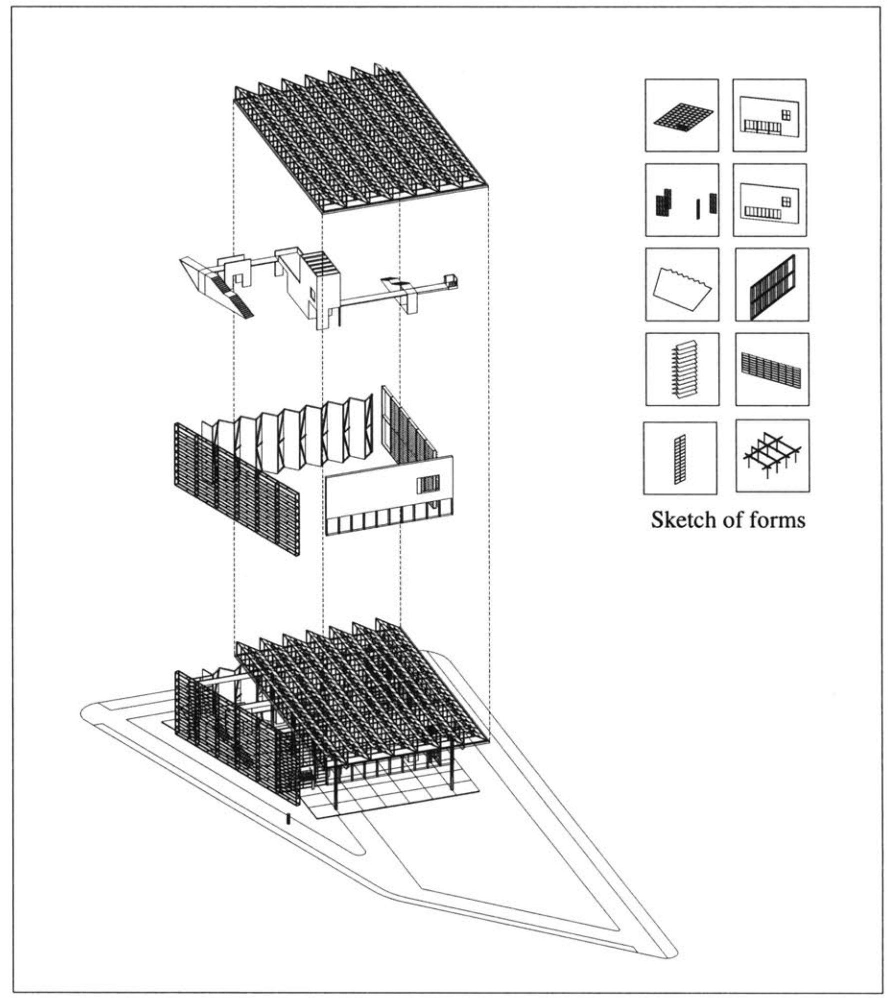
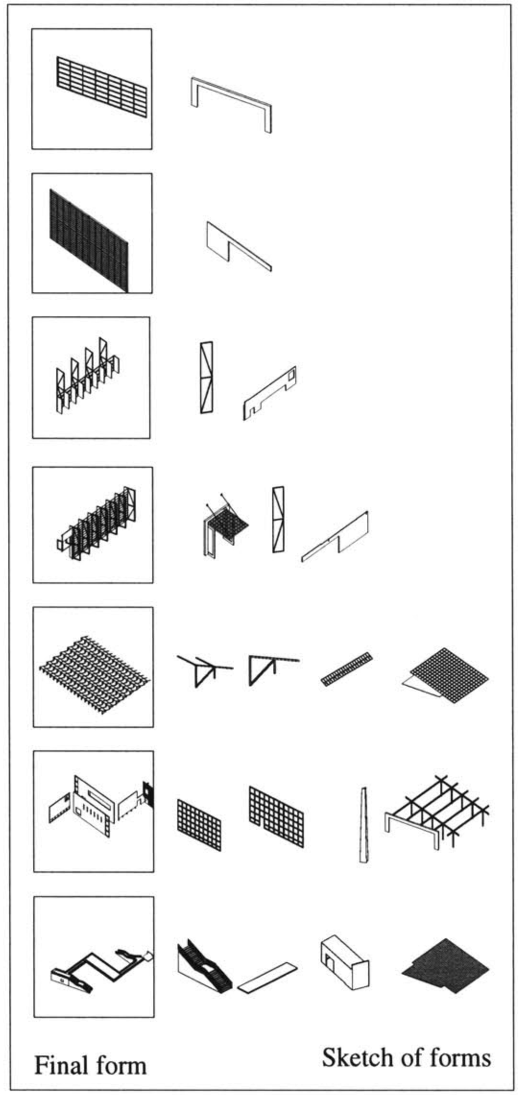


Figure 3.2.2.5 Scheme 3



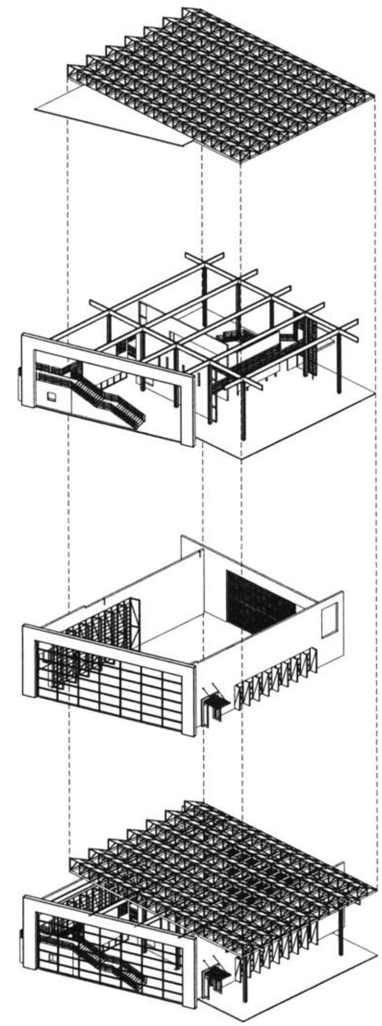
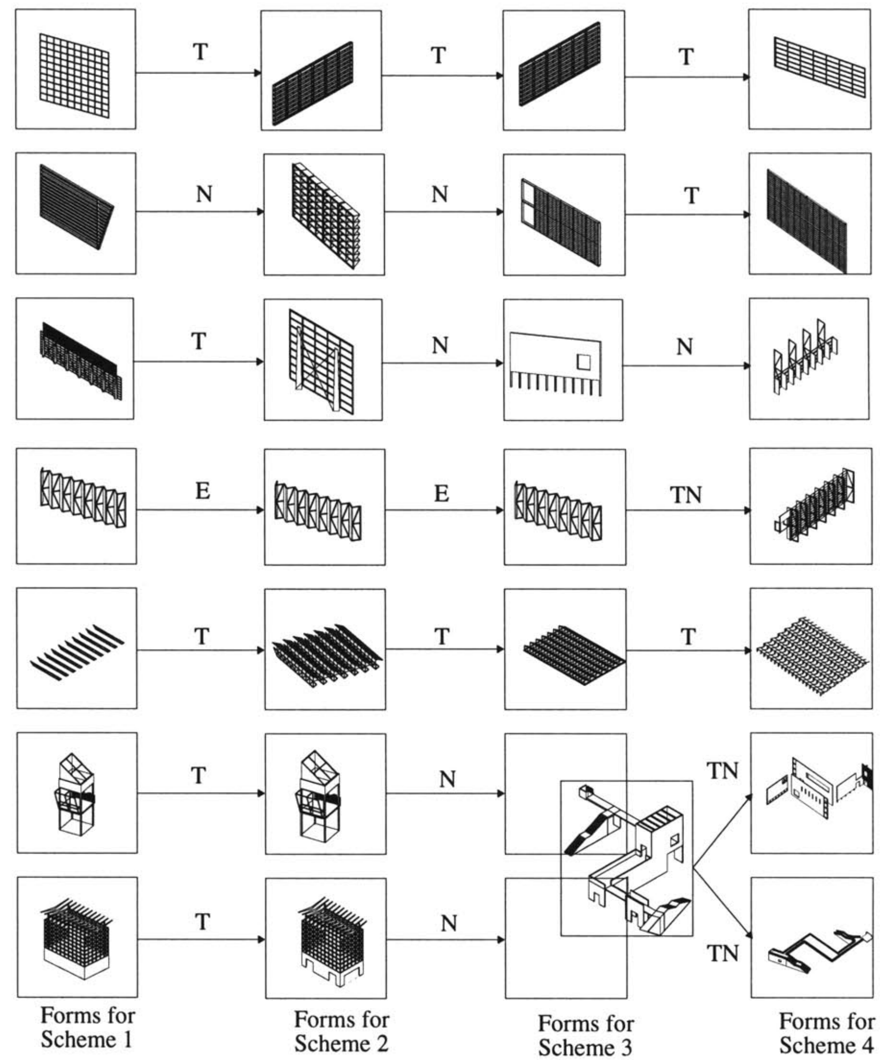


Figure 3.2.2.6 Scheme 4



T: transformation
N: new shape
TN: T + N
E: existing shape

Figure 3.2.2.7 Manipulation of vocabulary elements

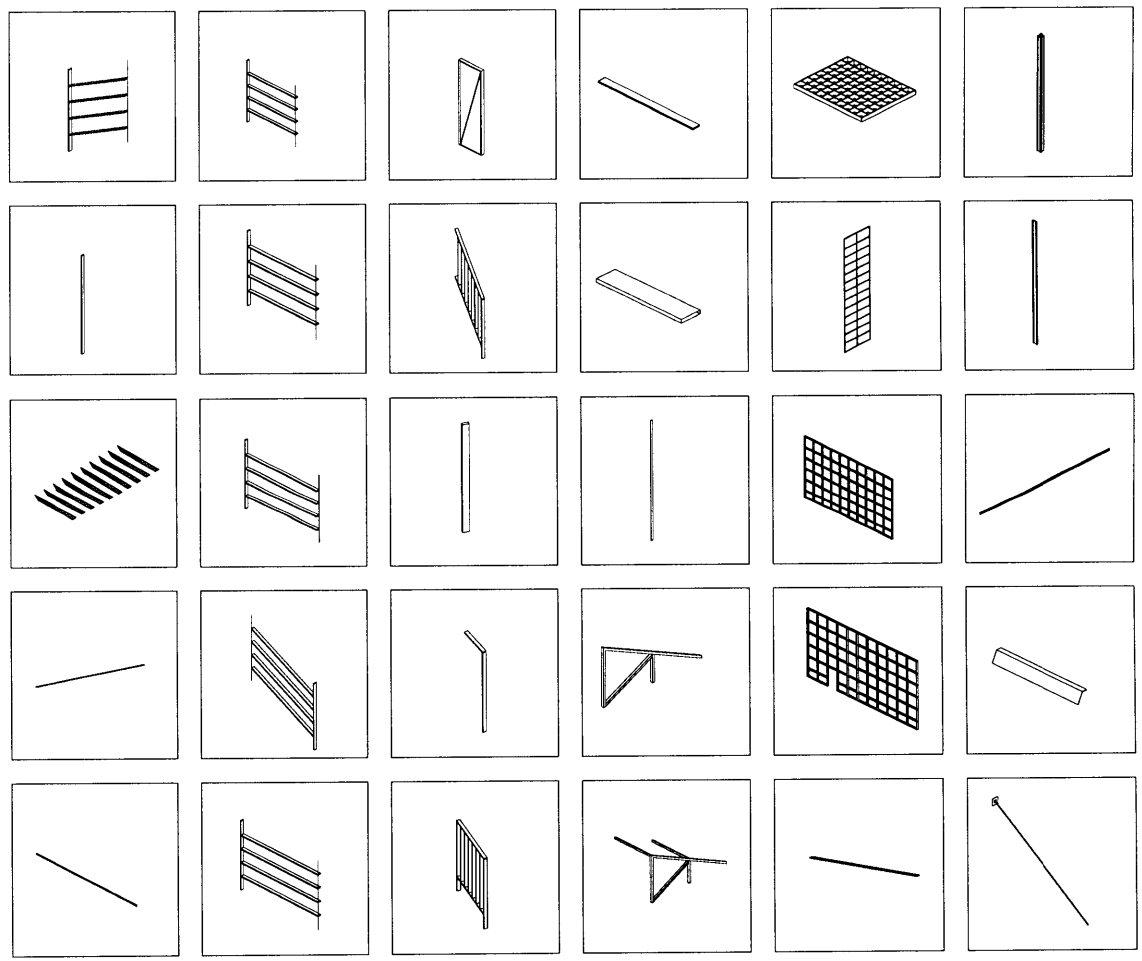


Figure 3.2.2.8 Sketch 1 of building components

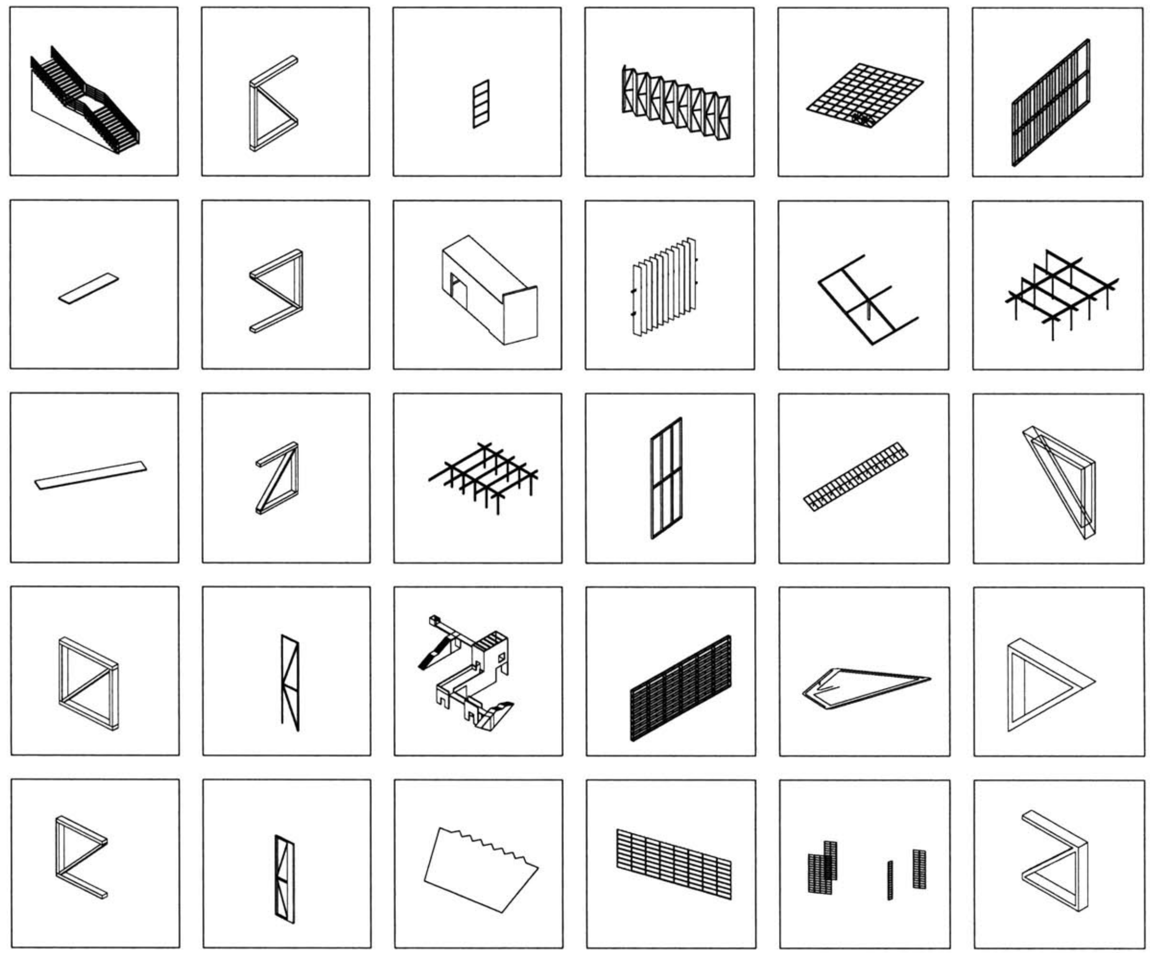


Figure 3.2.2.8 Sketch 2 of building components

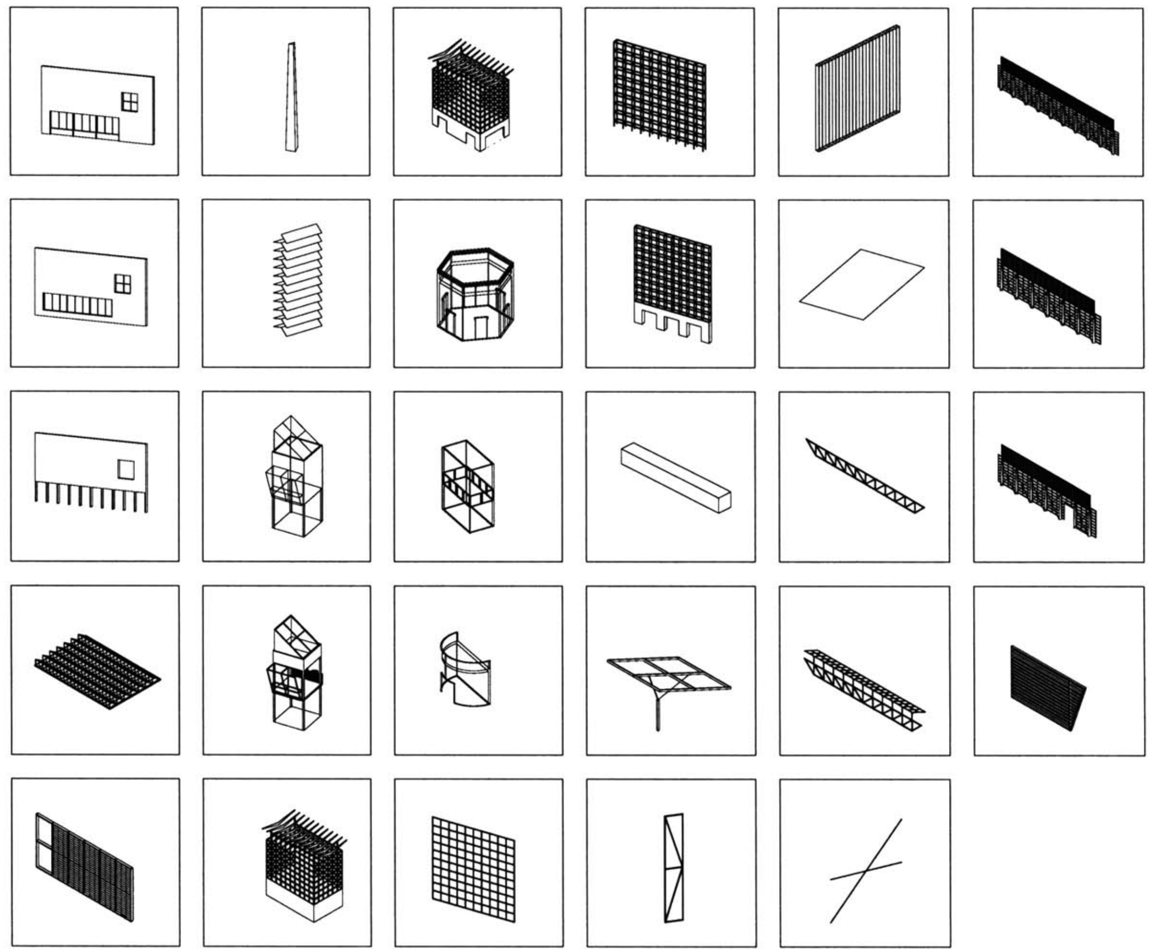
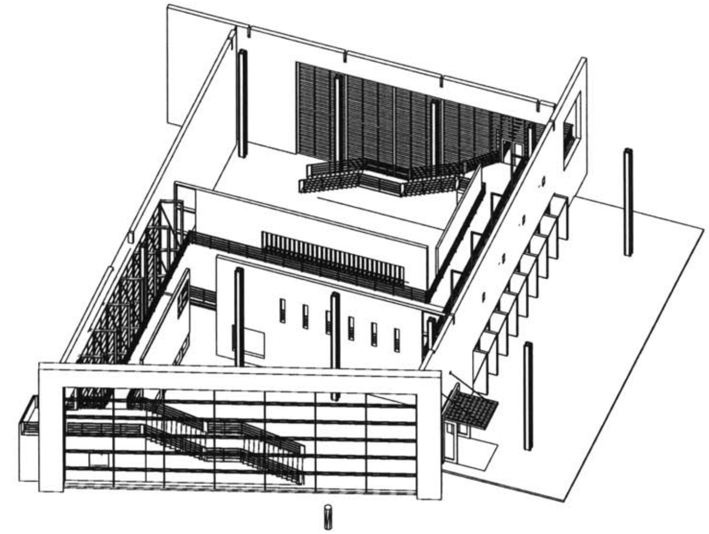


Figure 3.2.2.8 Sketch 3 of building components

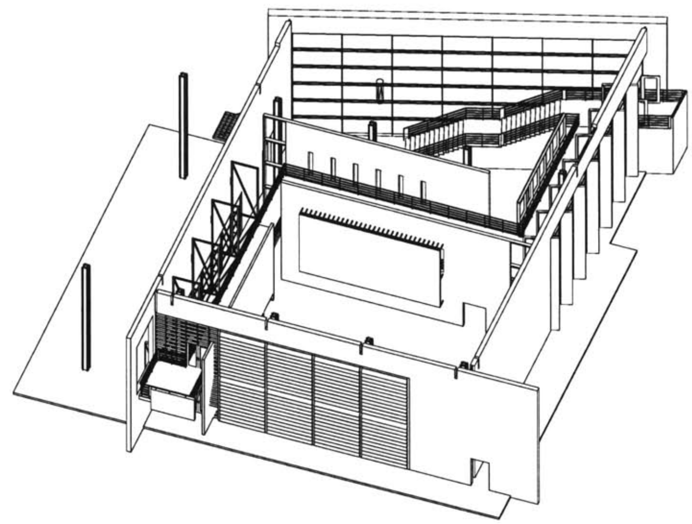
top-down method. Even though a designer has a abstract picture of a final building form in the inital design stage, my belief is that the bottom-up method is open to the changes of design states for shaping a building form and provides quick alternatives for design decision-making in order to determine a final form. While the top-down method tends to have a fixed direction to get to the final building form, the bottom-up method provides flexible ways of manipulating shapes to meet the intention of a designer. In addition, a designer can understand the inside space much better in the process of making a building in the bottom-up fashion than in the top-down fashion. Because the space is constructed with smaller objects from the beginning design stage in the bottom-up method, the designer can have the clear perception of larger space which is made out of the smaller vocabulary elements even though the geometry of the larger space becoms dense and complex.

From my experiments with two design projects, I believe that the bottom-up method is a better design technique than the top-down method to utilize the computer capabilities of manipulating objects, such as inserting, copying, and stretching. Furthermore, the bottom-up method seems to explore inside spaces better inherently in the process of making a building away from the issue of 'computer use' in design. It provides a designer with a better way of exploring complex spaces by providing a wide array of design alternatives for solving architectural problems. But there is a general consensus among CAD researchers and designers that both the top-down and bottom-up methods are complimentary and it may be impossibe to determine one absolute way of desing a building because of the subjective and artistic nature of architectural design. Also my experiment was limited to only two projects. More design experiments need to be excuted to find out the advantages and disadvantages of both bottom-up and top-down methods in the use of computer in the design process.

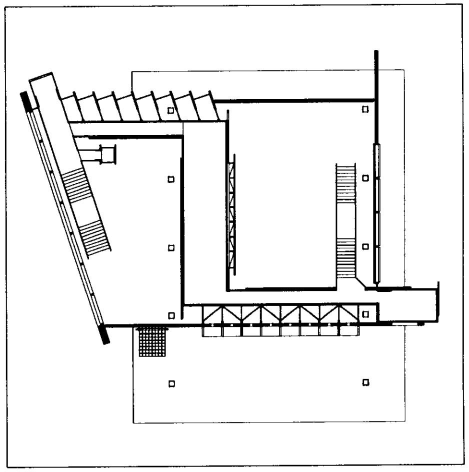


Axonometric View from North-West

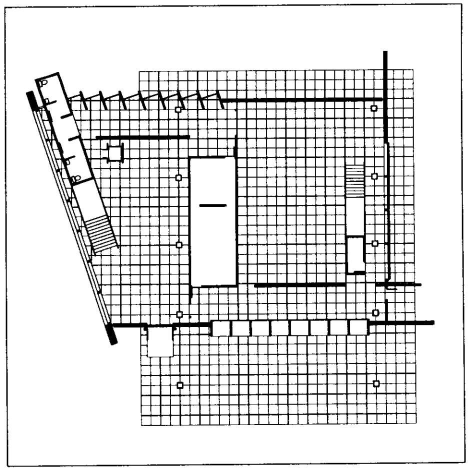
Bird's Eye View
from South-West



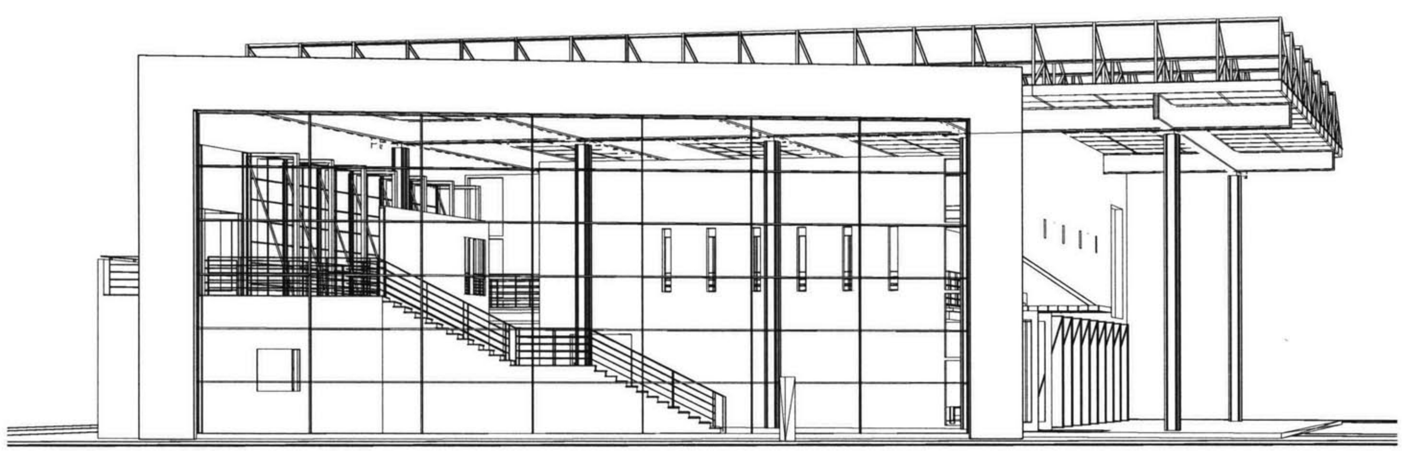
Axonometric View from South-East



Second Floor



First Floor



Perspective view from North

3.3 Knowledge Application

Manipulating and composing vocabulary elements is an integral part of the design process. Complex design issues require the flexibility and adaptability of the vocabulary elements to meet different spatial conditions at various contexts. By encoding a knowledge of vocabulary elements into a program, one can transform an object type into numerous subtypes, thus testing numerous spatial ideas with a wide range of vocabulary elements. This enables a designer to come up with better design alternatives for solving an architectural problem.

3.3.1 Computation of Design Vocabulary

Designing a building is like making a sppeech. Before making a speech, we must know words before sentences are made, and we must know rules of composition before paragraphs are constructed. A good speech can be made only with a good combination of words, sentences, and paragraphs. When we design a building, we must recognize the basic elements of architectural vocabulary which can be manipulated for solving a design problem. The basic architectural forms serve to lay out the organization of space and their transformation into a building. They become the critical means of construction which provides a primary tool for designer.

A built environment can be segmented into parts by name as shown in Table 3.3.1.1. Designers describe a building by combining these basic elements. In this process of combining the vocabulary elements, designers must know the mutual relationships beteen the basic elements in order to compose them harmoniously for the making of an architectural space. Table 3.3.1.2 shows the numerous shapes that are composed of smaller elements. The composition rule is based on Georgy Stiny's shape rule of 'addition' and 'subtraction' for the computational application. The shapes can be further combined into architectural spaces [Table 3.3.1.3] which can be eventually assembled to create a complete building identity. [Table 3.3.1.4]

Once the vocabulary elements are defined and constructed with parametric variation capability in a computer, they can be reused and transformed easily by creating various derivatives from them to meet the changing needs of design states. This CAD-generated form can be a powerful tool for design if one can make it as a prototype, and create various derivatives from it with

Taxonomy

Table 3.3.1.1

arch	elevator	panelboard
baluster	equipment room	pantry
balcony	escalator	parapet
bathroom	fence	parlor
beam	fire detector	pattern
bed	fire extinguisher	pavilion
bedroom	fireplace	pergola
bench	flag	photovoltaics
bollard	floor	planter
bookstacks	fountain	pool
bridge	freezer	porch
bumper	gallery	portico
cabinet	gateway	railing
carport	garage	ramp
ceiling	garage door	roof
chair	grid	shade
chimney	grille	shelf
cloakroom	gutters	sign
clock	handrail	sill
closet	hallway	sink
column	jamb	screen
connector	kitchen	skylight
corridor	ladder	sofa
courtyard	lighting	soffit
deck	lamp	stack
desk	laundry	stair
den	lavatory	step
diningroom	livingroom	step-ladder
dishwasher	linen	stool
door	mailbox	storage
downspouts	mirror	switch
dressingroom	monument	table
drinking fountain	nosing	tower
duct	nook	window
eaves	overhead door	walkway

Shape Algebra

Table 3.3.1.2

Balcony =
(+ -) overhang
(+) door
(+) plane
(+ -) support
(+) railing

Chair =
(+) plane
(+) back
(+) side
(+ -) leg

Chimney =
(+ -) cap
(+) flue
(+) fireplace

Column =
(+ -) capital
(+) shaft
(+ -) base

Desk =
(+) plane
(+ -) ornament
(+) leg

Gateway =
(+)facade
(+) door
(+-) portal
(+-) canopy

Pergola =
(+) beam
(+) column

Stairs =
(+) tread
(+) steps
(+ -) stringer
(+ -) body

Tower =
(+ -) flag
(+ -) ornament
(+ -) capital
(+) shaft
(+ -) base

Window =
(+-)lintel
(+) mullion
(+) frame
(+ -) sill
(+-)shading device
(+ -) overhang
(+-)curtain

Ex-wall =
(+)coping
(+-)gutter
(+-)sprout
(+-)window
(+-)gateway
(+-)door
(+-)opening

Ramp =
(+)rail
(+)walk

Truss =
(+)top-cord
(+)bottom-cord
(+)diagonal

Floor =
(+)ceiling
(+)beam
(+)joist
(+)deck
(+-)flooring

Roof =
(+)ceiling
(+)beam
(+)joist
(+)deck
(+)roofing

Skylight =
(+)ver-beam
(+-)slope-beam
(+)hor-beam

Door =
(+)plane
(+-)frame

Frame =
(+)ver-beam
(+)hor-beam

Ceiling =
(+)lighting
(+)grille
(+-)duct

Mailbox =
(+-)box
(+-)shaft
(+-)base

In-wall =
(+-)opening
(+-)grille
(+-)lighting
(+-)drinking-fountain
(+-)bench
(+-)shelf
(+-)panelboard

Rail =
(+)horizontal bar
(+)vertical bar

Space Algebra
Table 3.3.1.3

Entry(public) =
(+)gateway
(+-)vestibule
(+)lobby

Lobby =
(+)furniture
(+)colonnade
(+-)elevator
(+-)stair
(+-)skylight
(+)wall
(+-)glass
(+-)monument

Entry(private)=
(+)gateway
(+-)vestibule
(+)livingroom

Livingroom =
(+)furniture
(+)colonnade
(+-)chimney
(+-)stair
(+-)skylight

Closet =
(+)shelf
(+-)drawer

Kitchen =
(+)refrigerator
(+)dishwasher
(+)sink
(+)cabinet
(+)oven
(+-)wall-oven
(+)houseware

Bedroom =
(+-)bathroom
(+)bed
(+)desk
(+-)dresser
(+)closet
(+)chair
(+-)balcony

Dining =
(+)kitchen
(+)table
(+)chair

Walkway =
(+-)pattern
(+-)street-lighting
(+-)bench
(+-)kiosk
(+-)bollard
(+-)trash-container

Courtyard =
(+-)wall
(+-)glass
(+-)fountain
(+-)monument

Hallway =
(+-)in-wall
(+-)glass
(+-)door
(+-)ceiling
(+-)skylight

Plaza =
(+-)monument
(+-)tree
(+)walkway
(+)pattern

Built Worlds
Table 3.3.1.4

Museum

School

House

High-rise

Shopping Mall

Supermarket

Library

Theater

Bank

Prison

Furniture Store

Airport

Restaurant

Parking Garage

Car Dealer

Aquarium

Church

Nursing Home

Gymnasium

parametric variation capability, so that one can superimpose and transform them for spatial explorations in alternative compositions. This enables a designer to explore form and space and to produce a great number of plausible schemes efficiently for solving design problems.

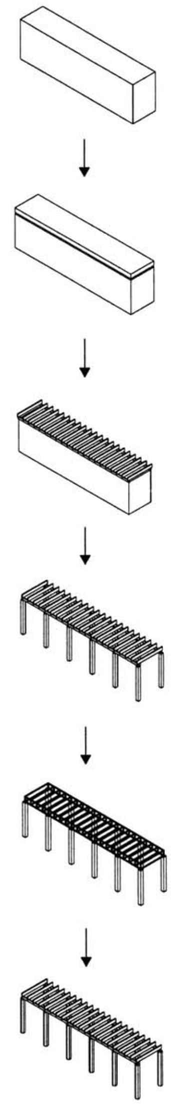
3.3.2 Top-down and Bottom-up Methods

With the design knowledge acquired from previous design experiments, two computer programs were written using Autolisp with two knowledge-base design approaches: one is bottom-up design method and the other top-down method. NITROS [1], a software shell, was used to implement the programs in Autocad environment. Through the interface with HTML, the NITROS can be dispatched to an Autocad environment; defined formal types (that is, pergola and gateway) can be loaded.

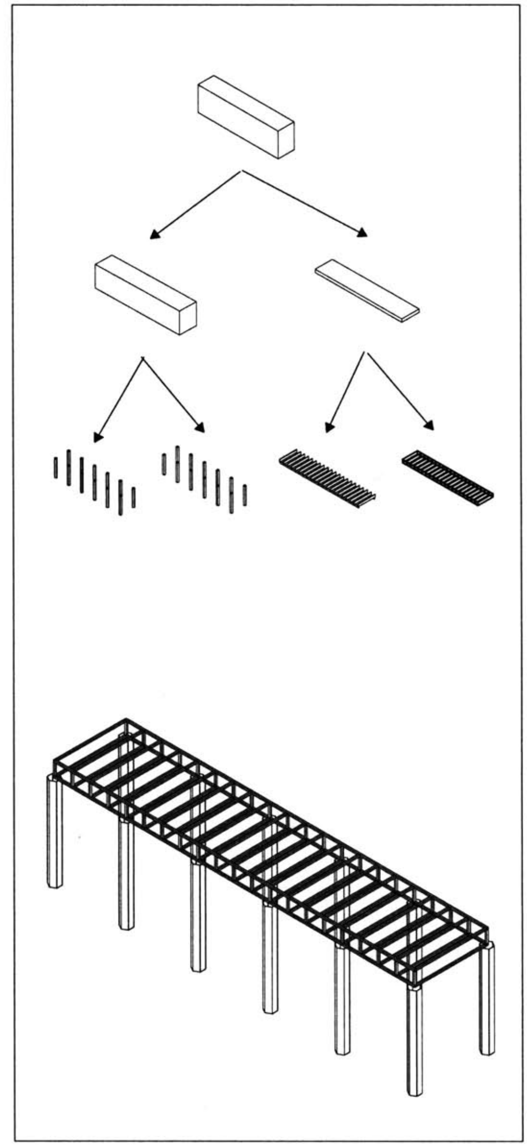
[1] NITROS is short for Nagakura's Network-interfaced Inter-Type Relation Operating Shell. Developed by Takehiko Nagakura, NITROS interface enables a user to assemble a complex geometry by instantiating a formal type, transforming it into another, parametrically revising the resulting assembly, and editing its constraint progagation paths.

In top-down method, one starts with a simple volume, and then articulating it into smaller volumes. Smaller subshapes are constrained parametrically to fit within a large shape so that they can affect each other when any of the shapes changes. Figure 3.3.2.1 shows how a simple rectangular box is articulated into a detailed pergola. The pergola is composed of a row of columns and capitals. If a parameter of columns is changed, not only the shape of columns is changed, but also the form of capitals is affected. By parametric variation, the pergola can be transformed into multi-shpaes in response to the progrmam and site. Figure 3.3.2.2 shows the interface between NITROS and HTML for dispatching, loading, and transformation of the pergola.

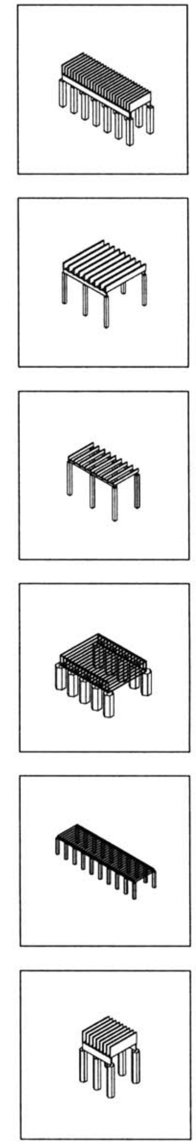
In bottom-up method, one starts with a detailed element of form, and then adds new elements to generate a desired shape. Individual vocabulay elements are parameterized independently in the beginning, but they are combined differently to create different types of shapes. In Figure 3.3.2.3, a door is the beginning element of the composition. There are three optional elements (that is, a portal, a wall-face, and a canopy). One can choose different combinations of elements to obtain different results. The door and a wall-face can be combined to create an ordinary facade and then a portal or a canopy or a combinatin of a portal and a canopy can be added to the door for the creation of different types of facades. It seems that the bottom-up design technique provides a effiecint method of manipulating shapes for spatial exploration. There are limitless possibilities of combining one vocabulary element with others. For example, the door can be combined with other elements to create not only a gateway, but



Top-down
Process



Pergola = Capital + Shaft



Parametric
Variations

Figure 3.3.2.1 Top-down Tree

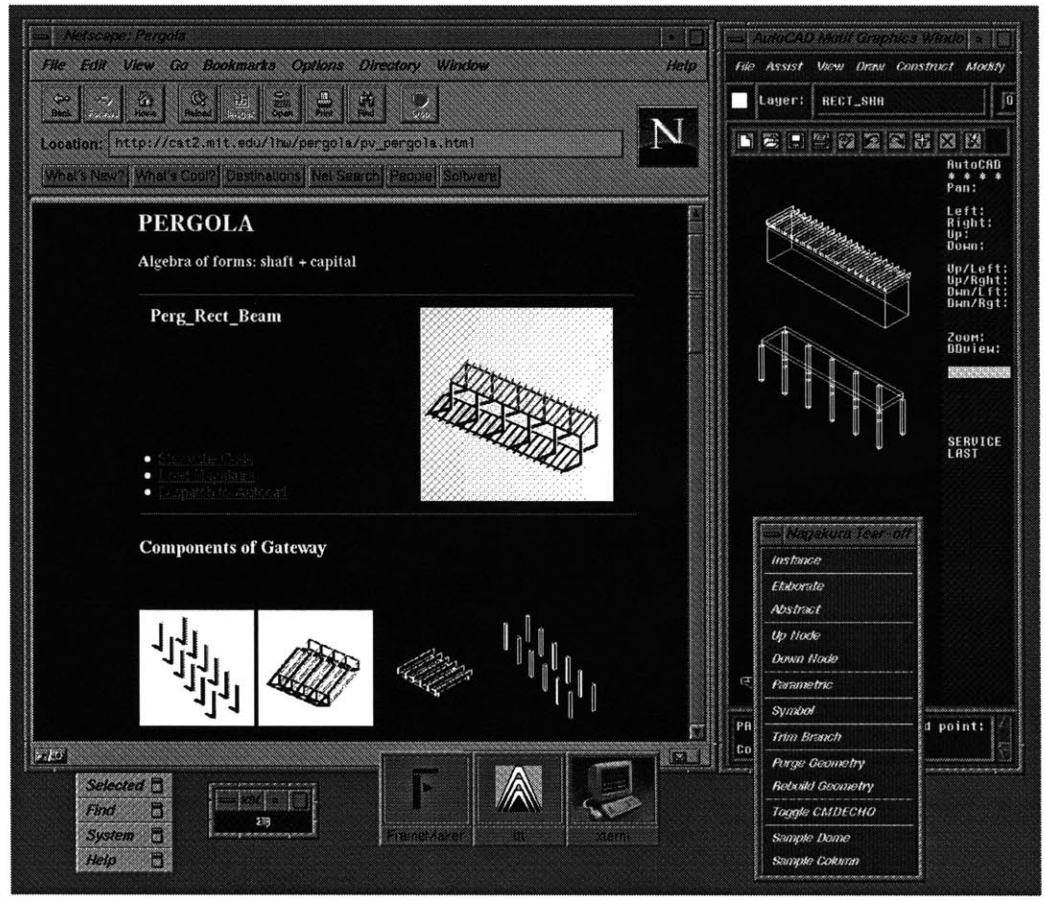
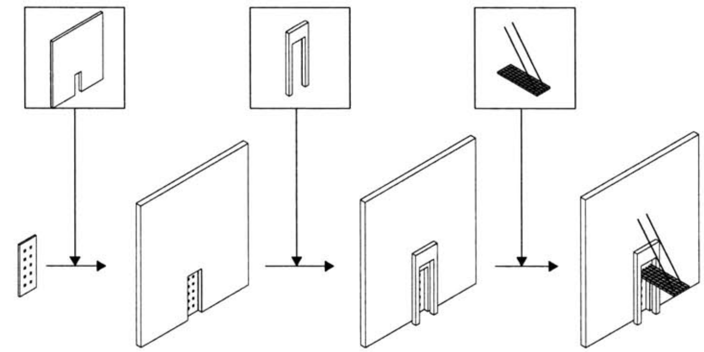
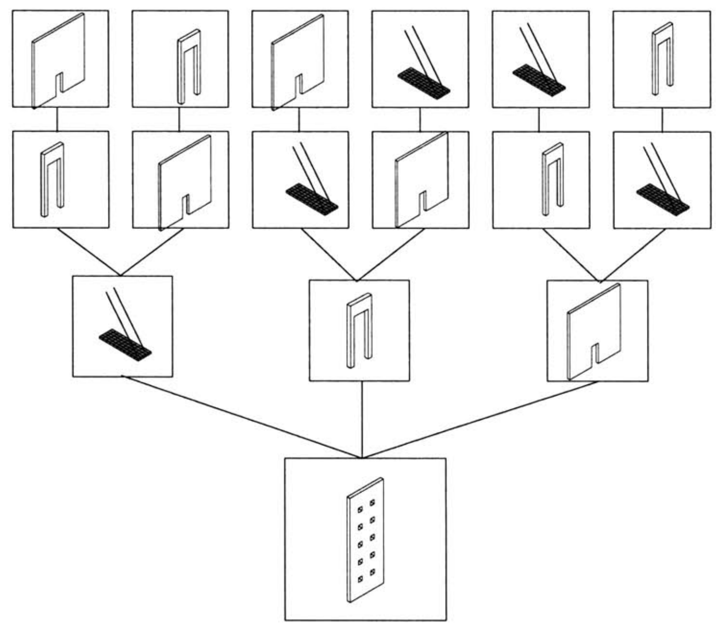


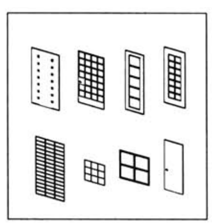
Figure 3.2.2.2 Pergola loaded and dispatched from Netscape to Autocad through NITROS



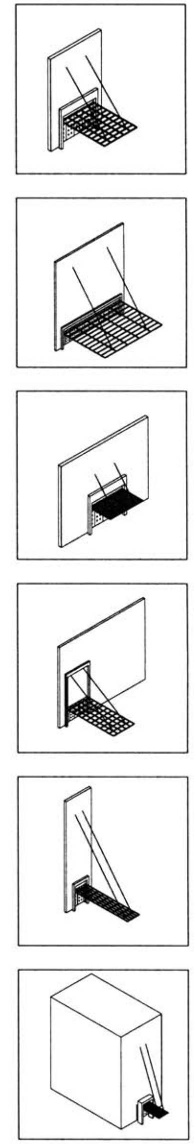
Bottom-up process of making a gateway



Gateway = Door + (Wallface/Portal/Canopy)



Door Variations



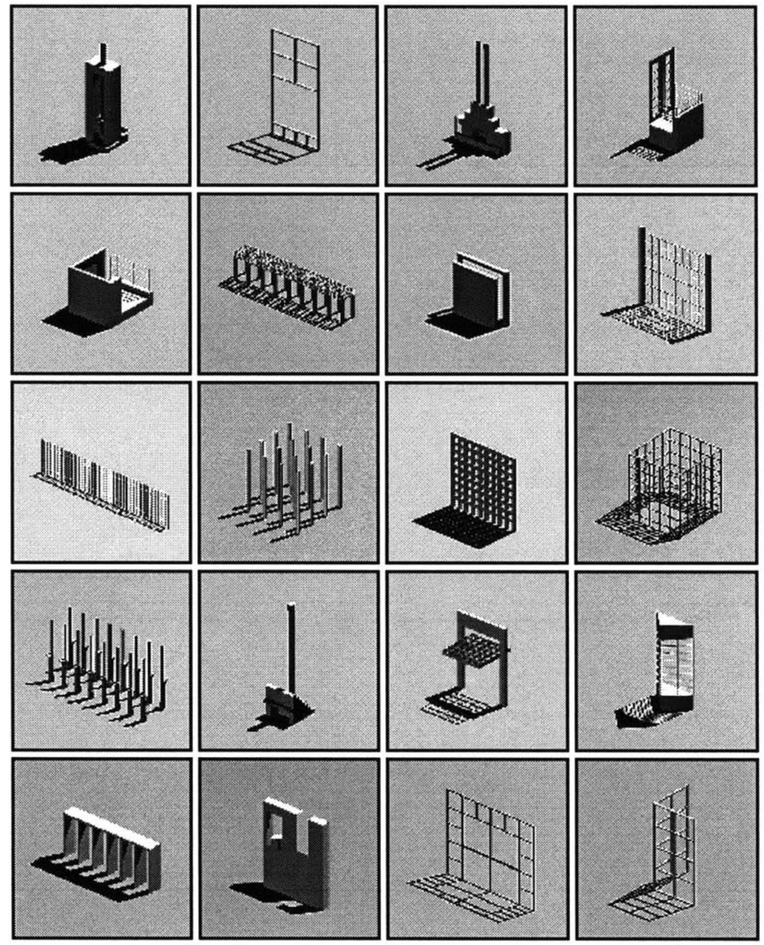
Parametric Variations

Figure 3.3.2.3 Bottom-Up Tree

Figure 3.2.2.4 Gateway loaded and dispatched from Netscape to Autocad through NITROS

also a hallway, a lobby, and any type of occupied space. Figure 3.3.2.4 shows the interface between NITROS and HTML for dispatching, loading, and transformation of the gateway.

It seems that top-down and bottom-up design methods are complimentary. When some shapes are inter-dependent, they can be parameterized to fit within a geometry resulting from their combination in top-down fashion. Otherwise sub-shapes are designed using the bottom-up method to be compatible with any other shapes parametrically so that their combination can produce a desired shape for a particular situation. However, from my own experience, the bottom-up method seems to be a more efficient method of manipulating shapes for creative design. The method is more flexible for adaptation to different conditions at various contexts. It is not limited to a pre-determined form, but provides limitless possibilities for combining vocabulary elements for various design alternatives.



Cubic Vocabulary

Reflection

Design is the art of assembling forms harmoniously to create space for human needs. From design concept to final production, designers manipulate forms until the forms are transformed into a building entity. The broad knowledge of architectural vocabulary navigates a designer to understand design activity, and enables him or her to lay out forms to create space.

This thesis developed a set of architectural vocabulary for experimenting with design projects. Based on the Aldo Rossi's methodology, my own vocabulary was developed to investigate how forms could be manipulated, transformed, and composed with top-down and bottom-up methods. By actual design experimentation, the advantages and disadvantages of the two methods were discovered: the current computational theory was evaluated: and a new direction for developing software tools for creative design was explored.

The project to design a school was approached in the top-down fashion. From the beginning, simple volumes were created in response to program requirements, and then gradually detailed into smaller, concrete volumes. Because it was my first attempt to use a computer in the design process, several obstacles were met right from the design development. Communicating ideas was difficult due to my inexperience in the use of a computer in the design process. But the most difficult was the top-down-design itself. Each time the design state was changed, most of the volumes had to be redesigned from the beginning to meet a new design state. The top-down design was inefficient for keeping up with the speed of changing ideas, and difficult to explore spatial relationships between objects - especially for understanding the inside space as complex geometries are overlapped.

In contrast to the school project, the pavilion project was developed with a bottom-up design method. In response to the program requirement, initial detailed forms were created; some of these forms were recycled from my existing vocabulary while others were newly introduced, and then built up to organize a space. A set of rules were created to compose forms in the early stage. These rules enabled me to simplify ideas from complex design issues and articulate form elements with mutual relationships.

With experience gained from two design experiments, two computer programs were written using Autolisp to investigate the role of design knowledge encoded by programs. It was not an easy task to learn the program, clarify the relationship between building components, and implement the parametric variation with top-down and bottom-up methods. But it seems that the encoded knowledge is extremely powerful once all the building components are defined parametrically with mutual relationships. The parametric variation seems to provide a designer with a wide range of design alternatives in response to the changing needs of design states.

In architecture, there is no absolute way of designing a building. Due to the artistic nature of design, none can suggest or teach a design doctrine which seems to solve all architectural problems. However, my belief is that there is a more efficient design method in the use of a computer and the bottom-up design method can be a powerful technique to explore form and space with the idea of typing vocabulary. This conclusion about the design methodology is not meant to be universally accepted, but to set out my journey and search for an principle to which I can look up for simplicity in architectural life.

Illustration

Bibliography

Adjmi, Morris. *Aldo Rossi: Architecture 1981-1991*. New York: Princeton Architectural Press, 1991.

Anderson, Stanford. *Types and Conventions in Time: Toward a History for the Duration and Change of Artifacts*. Perspecta 18, 1982.

Antoniades, Constantine G. *The Function of Repetition in Typology.* Thesis Arch. 1988. M.Arch.

Argan, Giulio Carlo. "*On the Typology of Architecture",* English translation by Joseph Rykwert, Architectural Design, Dec. 1963.

Arnell, Peter and Bickford, Ted, ed. *Aldo Rossi: Buildings and Projects*. New York: Rizzoli, 1985.

Baker, Geoffrey H. *Design Strategies in Architecture: an Approach to the Analysis of Form*. London: Van Nostrand Reinhold, 1989.

Baker, Geoffrey H. Le Corbusier: *An Analysis of Form*. New York: Van Nostrand Reinhold, 1984.

Baudini, Micha. *Typology as Form of Convention.* AA Files, no 6, May 1984: 73-82.

Ching, Francis D. K. *Architecture, Form, Space, and Order*. New York: Van Nostrand Reinhold, 1979.

Colquhoun, Alan. *"Typology and Design Method,"* in Essays in Architectural Criticism. Cambridge: MIT Press.

Frank, Karen A. et al. *Ordering Space: Types in Architecture and Design.* New York: Van Nostrand Reinhold, 1994.

Gero, John J. et al. *"Structural Design Knowledge as Prototypes."* Sydney: The Architectural Computing Unit. Department of Architecutal Science, University of Sydney, 1988.

Frampton, Kenneth. et al. *Oswald Mathias Ungers: Works in Progress.* New York: Rizzoli, 1981.

Jacobs, Stephen Paul. *The CAD Design Studio: 3D Modeling as a Fundamental Design Skill.* New York: McGraw-Hill, INC., 1991.

Jules, Frederick. *Form/Space and the Language of Architecture. Milwaukee: Center for Architecture and Urban Planning Research*, University of Wisconsin-Milwaukee, 1979.

Jung, C.G. *"Abstracts of The Collected Works of C.G. Jung,"* A Guide To Collected Works, Volumes I - XVII: Bollinger Series XX . Princeton University Press, 1968.

Hermander, Antonio. *J.H.L. Durand's Architectural Theory.* A study in the History of Rational Building Design.

Lansdown, J and Roast, C. *"The possibilities and problems of knowledge-based systems for design"* Environment and Planning B: Planning and Design, Volumne 14, 1987. Pp. 255-266

Lavin, Sylvia. *Quatremere de Quincy.* Pp. 86-100.

Madrazo, Leandro. *"Durand and the Science of Architecture"* Journal of Architectural Education, September, 1994, p. 12-24.

March, Lionel, ed. *The Architecture of Form. Cambridge*: Cambridge University Press, 1976.

McCullough, Malcolm. et al. *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era*. Cambridge: MIT Press, 1990.

Mitchell, William J. *The Logic of Architecture: Design, Computation, and Cognition.* Cambridge: MIT Press, 1990.

Mitchell, William J. et al. *Digital Design Media: A Handbook for Architects and Design Professionals*. 2nd ed. New York: Van Nostrand Reinhold, 1995.

Mitchell, William J. et al. *The Art of Computer Graphics Programming: A Structured Introduction for Architects and Designers.* New York: Van Nostrand Rinhold, 1987.

Moneo, Rafael. *"On Typology",* Oppositions 13, 1978.

Moos, Stanislaus von. *Le Corbusier: Elements of a Synthesis.* Cambridge: MIT Press, 1979.

Nagakura, Takehiko. *Form-processing: A System for Architectural Design.* PHD dissertation at Harvard University, 1995.

Oechslin, Werner. *Premises for the Resumption of the Discussion of Typology.* Assemblage 1, 1986:37-52.

O'Regan, John, ed. *Aldo Rossi: Selected Writing and Projects.* London: Architectural Design, 1983.

Quatremere de Quincy, *Type.* Introduction by Anthony Vidler, Oppositions 8. Cambridge:MIT Press, 1977.

Rossi, Aldo. *A Scientific Autobiography.* Trans. Lawrence Venuti. Cambridge: MIT Press, 1981

Smith, Max J. *Space-Form: Computer-Aided-Design for Architecture.* Salt Lake City: Architectural Research, Information Processing Systems, University of Utah, 1968.

Stiny, George. "*Computing with Form and Meaning in Architecture*," JAE, Fall, 1985, p. 7-19.

Stiny, G. and Mitchell, W.J. *"The Palladian grammar,"* Environment and Planning B 5:5-18, 1978.

Schmidt, Gerhard. *Microcomputer Aided Design for Architects and Designers.* New York: John Wiley & Sons, 1988.

Ungers, Oswald Mathias. *Architecture as theme.* New York: Rizzoli

Schirmbeck, Egon. *Idea, Form, and Architecture: Design Principles in Contemporary Architecture*. Trans. Henry J. Cowan. New York: Van Nostrand Reinhold, 1987.

Ungers, Oswald Mathias. *Oswald Mathias Ungers* Trans. Pamela Johnston. London: Studiopaperback, 1994.

Ungers, Oswald Mathias. *City Metaphors.* Köln: Verlag der Buchhandlung Walther König, 1982.

Thiis-Evensen, Thomas. *Archetypes in Architecture.* Oslo: Norwegian University Press, 1988.

Vidler, Anthony. *The Idea of Type: The Transformation of the Academic Ideal, 1750-1830.* Oppositions 8. 1977.

Vidler, Anthony. *The Production of Types.* Oppositions 8. 1977:95-115.

Villari, Sergio. *Sergio Villari:J.N.L. Durand(1760-1834):Art and Science of Architecture.* Trans. Eli Gottlieb New York:Rizzoli, 1990.

Yessios, Chris I. ed. *New Ideas and Directions for the 1990's*: ACADIA 89 Proceedings. Gainesville, University of Florida: ACADIA, 1989.