The Evolution of Form:
Exploring Algorithms as Form Determinants in Design

ABIMBOLA O. ASOJO
University of Oklahoma

INTRODUCTION

> *"A person doesn't really understand something until he teaches it to someone else. Actually a person doesn't really understand something until he can teach it to a computer, i.e. express it as an algorithm. The attempt to express it as an algorithm leads to a much deeper understanding than if we try to understand things in the traditional way" (Knuth, 1976, p. 709).*

This paper explores algorithms as form determinants design using the built in programming languages of Computer-aided design systems (CAD). The process involves defining design rules that are programmed into the computer. Several concepts like recursion, random generation and shape grammars are explored in order to produce a wide variety of possibilities. Two models utilizing the AutoCAD's Autolisp are presented in this paper. The first examines a parametric Autolisp routine developed for Le Corbusier, and the second explores genetic programming using elements from Corbusier's design style. The examples involve sets of instructions, which are the rules input into the computer. The user is then prompted for some parameters, and the computer executes the solution. The programs contain variables that incorporate the rules so that the results are not repetitive. The objective is to utilize the inherent properties of computers to generate a wide range of unexpected design alternatives. As a solid modeler, AutoCAD uses closed volumes as its primitives and a series of operations for inserting, deleting, reshaping, and positioning that allow for manipulating form. The programs involve manipulation in the three-dimensional world utilizing the X, Y, and Z coordinates. Such explorations permit the development of certain routines and transformations that are in line with design principles.

GENERATIVE THEORIES IN DESIGN

Since the Roman times, designers have been using generative theories to develop plans and work out the most appropriate plan from several alternatives. In the 19th century "Ecole Polytecnique" and "Ecole des Beaux Art" designed by exploring several ways in which elements of a fixed vocabulary could be assembled in different combinations to generate architectural form. Durand's Precis des Lecon's d Architecture (1803) also suggested ways in which sets of potential plans and elevations combinations can be generated.

Twentieth century architects used three-dimensional volumes in composing design. For example, Le Corbusier used a vocabulary of basic volumetric elements and assembled them into complex architectural composition. Likewise, Frank Lloyd Wright gained inspiration from his early childhood froebel blocks. Many of his designs emerge from a process of taking simple volumes and intersecting them in space to create form. More recently, Coates, Healy, Lamb, and Voon (1997) have employed generative modeling to generate limitless instances of forms by random growth, addition, and decomposition. The process involves computation of algorithms, algebra, and variable combined with design knowledge. The concept involves finite sets of relatively simple rules which result in complex outcomes (i.e. complexity from simplicity). The rules are explicit; their values are assigned, manipulated, and selectively applied. Generally, design has a number of components in relationship to one another and therefore has two aspects: the number of components and the relationship between components (i.e. objects and rules). The rules distinguish a random pattern of objects from a significant design. The rules can be employed in composing and decomposing architectural objects as generating or analytical tools.

The design process disseminated in this paper relies on identifying design rules and their relationships and utilizing concepts such as bottom-up/top down approach, shape grammars, and algorithms. The author also explores the development of design grammars in order to utilize the inherent potential of computer-aided design systems as design tools through programming and not just as drafting tools in design.

BOTTOM UP/TOP DOWN APPROACH TO DESIGN

Mitchell (1990) notes "if we approach architectural composition in bottom-up fashion, we rely on our knowledge of the formal and functional characteristics of given architectural vocabulary elements to suggest feasible and useful ways of putting them together in composition. Conversely, if we approach design in top-down fashion, we rely on our knowledge of formal and functional characteristics to suggest appropriate choices and adaptations of elements to pro-

vide given functions in given contexts. In either case, our knowledge of how to select, shape, and put things together to serve architectural purposes can be expressed in the form of shape rules "(p. 234).

Fawcett and Woljtowicz (1986) note "a complete design is the end product of a process that begins with a vocabulary of well-defined components"(p.26). In the bottom up approach, the process of design involves a selection of components from the vocabulary, placing the first one and adding others successively assembles the design. In contrast, the top down compositional technique starts with an abstract form, which is elaborated until it is transformed into form. Here we have a comparative research methodology of the black box versus glass box respectively, demonstrated in the creative process that generates form. Both approaches rely on knowledge of a given vocabulary and sets of relatively simple rules which result in complex assemblies (i.e. complexity from simplicity).

SHAPE GRAMMARS

Frank Lloyd Wright (1954) stated in the Natural house that "every house worth considering as a work of art must have its own grammar" (p. 296-297). He also stressed the importance of consistency in grammar and the importance of a design having a language of its own. Stiny (1980) defines "shape grammars as a set of initial conditions, a lexicon of primitive objects, and syntax of transformations on those objects".

Fawcett and Wojtowicz (1986) define "shape grammar as a principle by which vocabulary elements can be put together, and inherent in a grammar is the set of mappings between vocabulary elements such that certain grouping of elements can be transformed to another group"(p.43-67).

The process has been demonstrated in producing line drawings that resemble those of Palladio by Stiny and Mitchell (1980) and Frank Lloyd Wright's villas by Koning and Eizenberg (1981). Shape grammars have been employed by Richard Coyne (1988) to describe algorithms for performing arithmetic operations on geometric entities called shapes.

More recently, Coates and Makris (1999) have incorporated shape grammars and genetic programming in spatial composition by starting with sets of geometrical structures and their relationships. Their approach uses genetic programming with a library of objects in a genetically bred computer program. The concept of design vocabulary is used to enable a simplistic definition of the rules which can be input as points, lines, volumes, shapes, and primitives in a computer algorithm for the exploration of alternative solutions.

RESEARCH MODELS

The following two research models illustrate experiments in form generation in Autolisp, AutoCAD programming language. The first is a parametric Autolisp routine for Le Corbusier and the second explores genetic programming using elements from Le Corbusier's design style.

MODEL I: PARAMETRIC AUTOLISP ROUTINE FOR LE CORBUSIER

In Le Corbusier's own summary of his main architectural elements, he identified in his buildings the following five points of Architecture: Pilotis, Roof Garden, Free plan, Ribbon Windows, and Free Facade. The Pilotis raised the building off the ground into the air and allowed for the space underneath to be used for parking cars, road, or gardens. Space lost was replaced on the top by a roof garden which was a space open to the sky, containing greenery with a view all round. Free planning was possible since the frame carried the weight and partitions were organized independently (e.g. some were curved to express their freedom and function). Ribbon Windows were located from side to side of the facade horizontally lighting the whole interior evenly and giving maximum view. The Free Facade portrayed the exterior walls as non-load bearing, thus reemphasizing the inherent potentials of a frame.

Le Corbusier employed this five points in three main combinations: firstly, as a membrane stretched over reinforced concrete frame where walls enclosed the columns; secondly, setting the walls back from the main structural frame; and third, as mass penetrated.

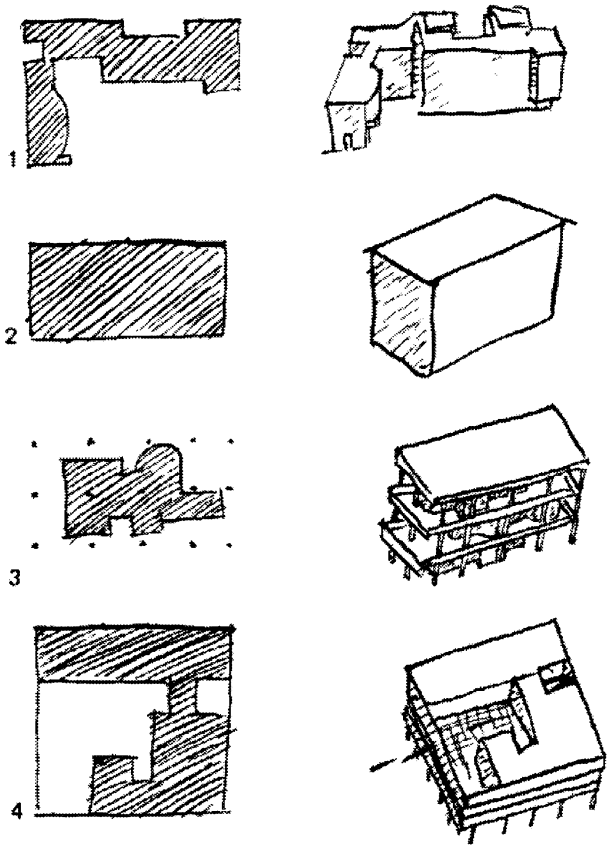

Figure 1 illustrates three main combinations of Le Corbusier's Five points of Architecture.

Table 1 - summarizes major elements from Six of Corbusier's buildings The following buildings were chosen because of their similarities, size, simplicity of language of design and relationship to the five points of architecture, which is the starting point of the design algorithm.

Project	Building Form	Major Elements	Spaces
Citrohan House project 1920	Rectilinear and cubical in configuration. The building is subdivided into two cubical forms.	Double height space in living room External staircase Interior spiral stairs Zoning-Living, cooking, dining and sleeping.	Living, gallery, roof terrace, bedroom and kitchen.
Villa at Vaucresson 1922	Double cubic form, building form based on relationship between two masses. Contrast between smaller vertical and larger horizontal elements	Sloped site Entry is placed between two masses Clearly defining major and minor elements. Detached staircase Main axis parallel with road.	Living, dining and bedrooms.
Houses for Workers, 1924	Cube with triangular Upper floor slab	Double height living room Three activity zones Diagonal reinforced with straight flight of stairs. Minimum accommodation requirements	Living, dining, Bedroom and kitchen.
Weissenhof house 1924	Rectilinear configuration. Building raised on pilotis	Main staircase is enclosed Roof terrace Curved elements control movement Pilotis imply garage is underneath building and living room on the 1st floor. 3 columns in transverse direction 5 columns in longitudinal direction. Double height living spaces	Roof terrace, living, bedroom, dining and kitchen.
Villa at Garches for Michael Stein, 1927	Large scale domino structure. Mass around the reinforced concrete frame.	5 columns in longitudinal and transverse direction. 5,2.5,5,2.5,5 an ABABA relationship longitudinally. 1.25,4,3,4,1.25 an ABCAB relationship transversely. Two apsidal staircase Convex and concave walls define spaces. Free forms. Apsed hammerhead forms.	Living, dining, Bedrooms, gallery and roof Terraces.
Villa Savoye, 1929-31	Contrast between enclosed and open spaces. Building raised on pilotis.	Central Ramp Pilotis. Roof Garden 5 bays in longitudinal and transverse direction. Curved elements encloses and defines circulation and major axis. Entrance on the curved wall	Living, dining, Bedrooms, gallery and roof Terraces.

Table 1 summarizes major elements from six of Corbusier's buildings.

The buildings were chosen because of their similarities, size, simplicity of language of design and relationship to the five points of architecture, which is the starting point of the design algorithm.

Another design configurations used by Corbusier such as the Interlock system had elements locked around the service stack (Baker, 1986). Since the frame of the building carried the load, part of the floor slab was taken out to create double height rooms or semi-open spaces. The facade was also opened at any level or removed entirely or became sunshades with the walls set back from the facade. The reinforced concrete frame and slab formed the basis of Le Corbusier's main design approach, and within this cage various activities of the building were accommodated. The structural elements were organized to easily accommodate circulation elements.

Program Rules

The program rules were based on the main elements of Le Corbusier's architectural style and classified under the following main categories:

1. Orthogonal Cage determination
2. Circulation
3. External walls
4. Main curved elements of interior

These categories were further elaborated into rules specifying points, vectors, polygons, and other graphic tokens that could be interpreted in Autolisp.

Orthogonal Cage Determination

The orthogonal cage represents the structural frame and the rules were based on the following:

> Rule 1. The structural grid system was based on the number of columns in the transverse and longitudinal direction of the building. The relationship between the longitudinal and transverse axis are based on proportions identified in Le Corbusier's design and interpreted in this project as an ABC relationship. This is interpreted in the program in an X and Y-axis. A represents the distance between the columns in the X axis; B varies between being equal to A, 0.5A to 0.75A; and C exists when there are multiple column grid spacing in the X axis.

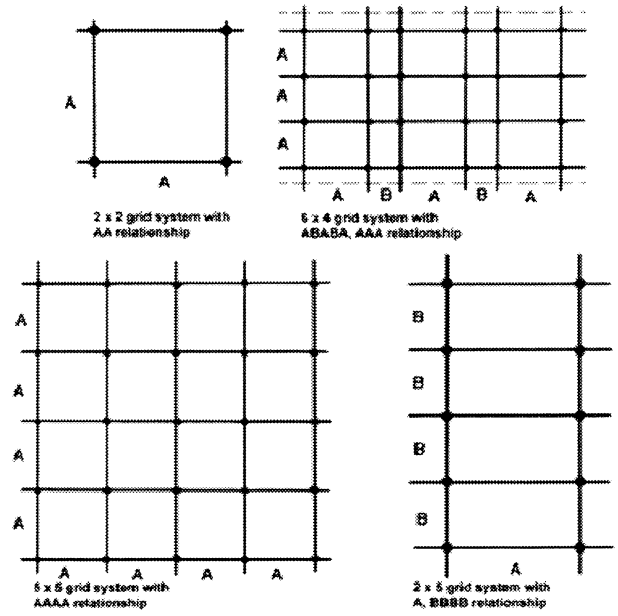

Figure 2 illustrates some grid rules.

> Rule 2. Like Le Corbusier after the determination of the column grid, a major axis is defined in the X or Y-axis. The rule is to place the major axis in the Y direction if the number of columns in X was greater than in Y and vice versa.

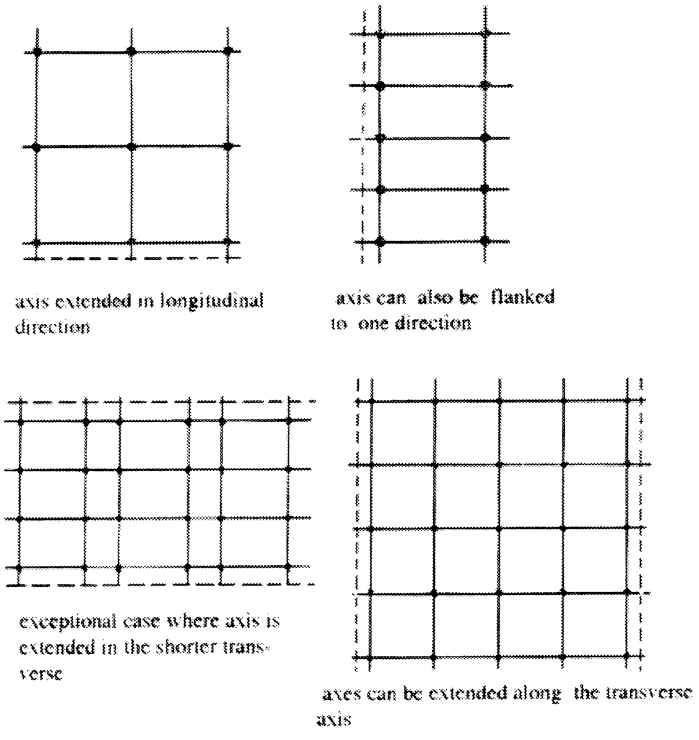

Figure 3 illustrates rules for axis determination.

Rule 3. Randomly determine the number of floors between two, three and four.

Rule 4. Place the living room position either on the first or second floor.

Circulation

The vocabulary of circulation elements were dogleg, spiral, straight flight apsidal staircases, and ramps as identified in Le Corbusier's buildings.

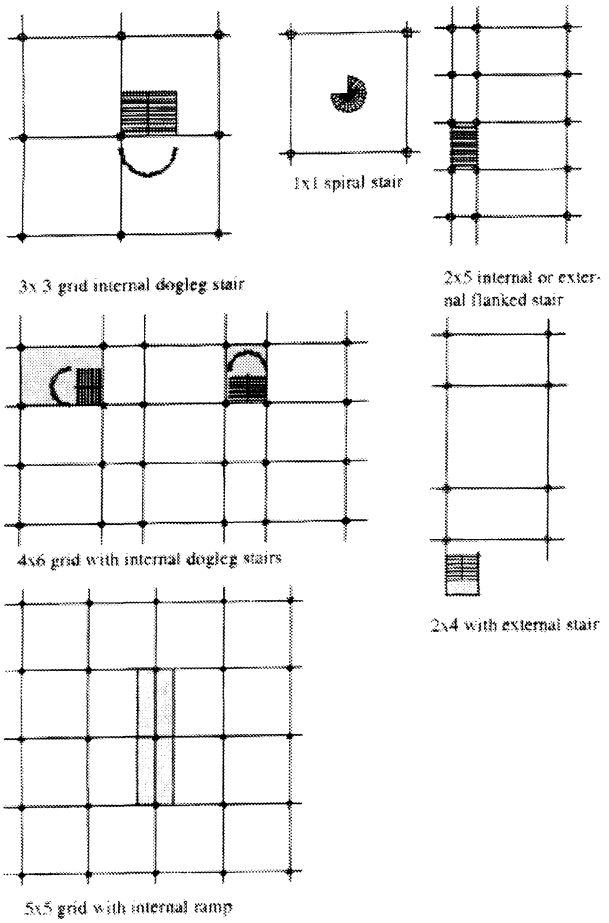

Figure 4 illustrates vocabulary of circulation elements.

Rule 1. Spiral staircases are placed in buildings with two columns in the X and Y-axis.

Rule 2. A straight flight of stairs is flanked to the side when placed in a three by five grid.

Rule 3. A ramp is placed in a five by five grid.

Rule 4. Two apsidal stairs are located in a five by three column grid.

Rule 5. External dogleg stairs are in a three by five column grid.

External Wall Placement

The placement of the structure's external walls were based on three main criteria: firstly, a membrane around the structural frame; secondly, set back from the structural frame; and thirdly, around the structural frame and raised on pilotis.

The following rules were applied:

Rule 1. If the number of floors equals two, then the external wall is a membrane around the structural frame.

Rule 2. If the number of floors equals three or four, then the buildings could be raised on pilotis or set back from the reinforced concrete frame or a membrane around structural frame.

Main Curved Elements of Interior

Corbusier utilized a variety of curved elements in his building interior since partitions were non-load-bearing. Their independence was usually reflected in their free organization, and curved elements were concave and convex in form.

Rule 1. Curved elements defined building entrances.

Rule 2. Curved screens defined terrace floors.

Rule 3. Curved elements defined circulation elements.

Rule 4. Ramps were enclosed by curved elements.

Programming

AutoCAD's built in programming language Autolisp was chosen based on AutoCAD's adaptability and popularity. Autolisp is derived from common lisp and can be customized to specific needs. Lisp presents information in form of lists. Lisp is known to be the most extensive of computer languages; it has about 200 to 300 built-in functions, and programmers can also create their own functions. In lisp, functions are used to express data and programs.

In the program, separate functions were written for main elements like the column grid, circulation elements, terraces, curved screens, external wall placement, etc. The functions were given separate arguments based on the rules, and a main function evaluates all these separate functions. The use of randomness was incorporated so that the program determines the evolution of the design and thus explores architecture as self-generated.

The Autolisp routine is loaded from the command prompt in an AutoCAD drawing file, which then prompts the user for a random number function. The user can input a random number between 0 and 32567, then the program by itself selects a column grid system based on the program rules. Upon determining the column grid system, it determines the number of slabs, types of external faces, elements of interior, and then the curved screen. Figure 5 to Figure 7 illustrates some of the "Corbu" like prototypes generated

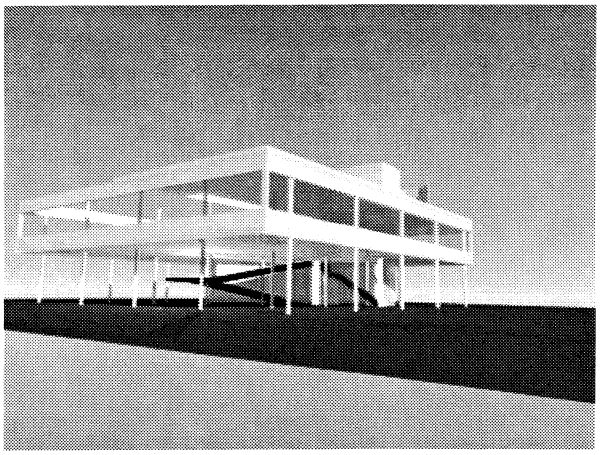

Figure 5 illustrates generation from random no. 13.

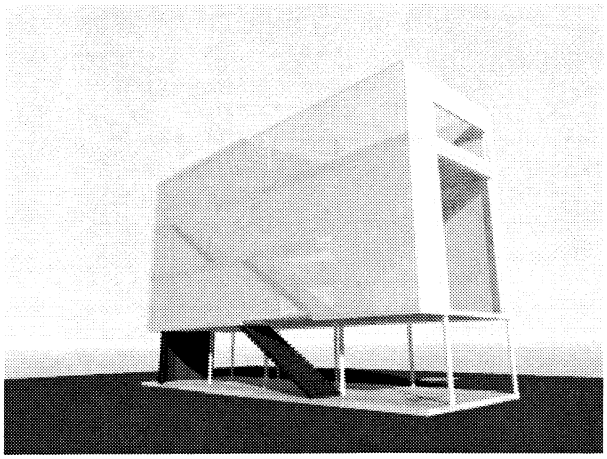

Figure 6 illustrates generation from random no. 35.

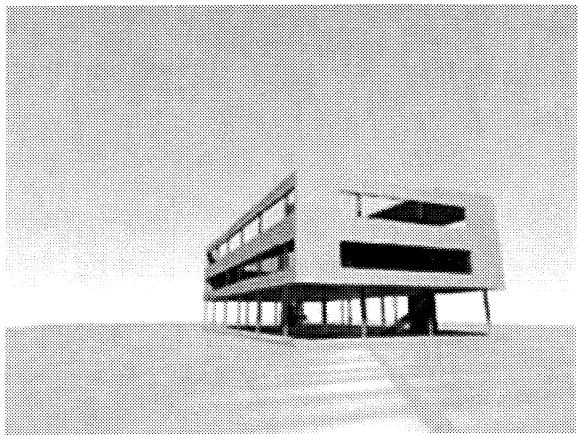

Figure 7 illustrates generation from random no. 119.

Main part of 16 page Autolisp Routine

```
(defun c:bay ( / )
    (setvar "cmdecho" 0)
    (solservmsg 0)
    (command "ucs" "w")
(command "layer" "make" "0" "")
    (command "erase" "all" "")
(setq p1 (list 0 0 0)
        rad  0.1 (getreal "enter radius of column")
h  3.0
seed(getint "seed value for random number function"))
(setq a(getreal "max dist between cols"))
(setq choice (fix (rand 1 4)))
(cond ((= 1 choice)(setq b (/ a 2)))
((= 2 choice)(setq b (* a 0.75)))
        ((= 3 choice)(setq b a)))
    (setq colx (fix (rand 2 6)))
    (setq coly (fix(rand 2 6)))
    (setq ch(fix (rand 1 4))
        makeit T copyouter T)
(cond  ((and (= colx 2)(= ch 1))(setq coly 2 makeit nil))
        ((and (= colx 2)(= ch 2))(setq coly 6))
        ((and (= colx 2)(= ch 3))(setq coly 5))
        ((and (= colx 3)(= ch 1))(setq coly 3))
        ((and (= colx 3)(= ch 2))(setq coly 6))
        ((and (= colx 4)(= ch 1))(setq coly 2))
    ((and (= colx 4)(= ch 2))(setq coly 6))
    ((= colx 5)(setq coly 5 copyouter nil))
    ((and (= colx 6)(= ch 1))(setq coly 2))
    ((and (= colx 6)(= ch 2))(setq coly 4 copyouter nil)))
(command "layer" "make"  "cols" "")
(setq tp(colgrid p1 (list a b) b coly colx makeit copyouter))
```

```
(command "layer" "make"  "slabs" "")
(setq h1  0.2)
(setq slabnum (slab p1 h1 tp ))
(command "layer" "make"  "external" "")
    (setq face (external))
    (terrace)
    (command "layer" "make"  "steps" "")
    (stairsNstuff)
    (openings)
    (command "layer" "make"  "duct" "")
    (stack)
    (curvescreens)
    (command "zoom" "e"))
```

MODEL II: GENETIC PROGRAMMING

This model explores genetic programming using elements from Le Corbusier's vocabulary. Genetic programming allows the parallel exploration of design worlds defined by initial axioms and production. The aesthetics of the end product depend entirely on the initial grammar. A good set of axioms and production may lead to success, while badly chosen axioms may lead to small design worlds. Coates P. and Makris D. 1999 note "a well chosen grammar leading to a large number of non-trivial design worlds increases the likelihood of finding a suitable candidate as the solution to a properly posed problem"(p. 4).

In genetic programming the basic idea is that architecture results from the multiplication of simple relationships. The range of moves available when exploring by hand are limited. Coates and Makris, 1999 note "the use of a recursively defined generative grammar using genetic programming allows for recombination and embedding of morphological moves to any level of complexity required"(p. 2).

Program Rules

This model starts with some basic configurations from Le Corbusier's vocabulary; the columns, slabs, and column grid relationships already identified in the previous algorithm represent the initial conditions. In the Autolisp program, the geometry of the initial conditions are defined and a set of transformation defined in the x, y, and z axis. These transformations like the previous model are based on an ABC relationship, where A represents the distance between the columns in the X axis and B represents the Y axis and varies between being equal to A, 0.5A to 0.75A and C exists where there are multiple spacing in the x axis. In the Autolisp routine the first generation of eight objects are generated from this initial conditions, and the user has control over future generations by selecting two parents from this generation to be mutated

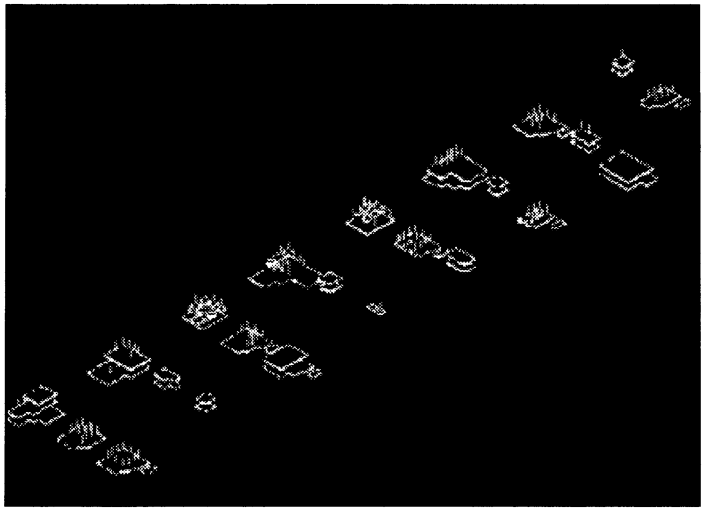

Figure 8 illustrates generation from GP for Le Corbusier – Random seed 1. Probability of mutating 2. No of Generations 2. Parent of Row 2 = 1&2. Parent of Row 3&4.

The Autolisp routine presents configurations that are similar to Le Corbusier's and the mutations are driven by visual judgement, which encourages cooperation between the computational power and human creativity. The resulting forms at this stage illustrate that an evolutionary approach is related to the process of generating composition. Coates 1999 suggests "that a complete examination of the implication of genetic programming in architectural design would necessarily reflect the inevitably complex and dynamic character of architecture, and draw some lines towards methodologies to model brief and space"(p.3).

CONCLUSION

Coates and Makris (1999) note "the basic design problem consist of the prediction composition of the solution from primary determinants. Different design strategies contain different theories to approach the compositional problem. The problem is that though it is relatively easy to determine the putative structure of the problem, the determination of its possible formal structure is extremely difficult. Furthermore, the problem definition (brief, program, criteria matrix, bubble-diagramming etc.) does not imply a solution, but rather should form a basis for testing possible configurations"(p. 3).

Exploring design algorithms present an opportunity to examine a wide variety of possibilities and unexpected alternatives. Exploring algorithms in form generation highlight the importance of rule based systems as an integral part of the design process and rules can be modified to systematically define a new language of design

that reflect changing circumstances and incorporates new ideas. The process offers an opportunity to develop a deeper understanding of the design process through defining simple rules in form of grammatical relationships between design. The process contributes to the human technology interface debate and serves as a starting point for utilizing CAD systems in generating design rather than utilizing them merely as rendering and drafting tools. Perhaps, as expertise in visualization skill increases within the profession, the development of design algorithms through automated systems becomes an area that needs to be explored by designers, to utilize computers to their fullest capabilities in creative thinking and problem solving.

Since the range of solutions available when exploring by hand are limited by the increasing complexity of the design, the next step of this research is to introduce this methodology in studio and digital media classes. The author does not propose to diminish the designer's capability in anyway; it should be noted that the end product strictly relies on the chosen axioms defined by the designer. A well-chosen grammar results in the likelihood of finding more possible design solutions. The methodology proposes alternative creative techniques and offers the designer an opportunity for exploring a wide range of design possibilities.

REFERENCES

Asojo, Abimbola O. "A Design Algorithm After Le Corbusier," *Proceedings of the Association of Collegiate Schools of Architecture Technology Conference.* Montreal, Canada,(June 25-27, 1999): 110-115.

Baker Geoffrey H. *Le Corbusier: an Analysis of Form.* New York: Van Nostrand Reinhold Co. Ltd, 1996.

Boesiger Willy. *Le Corbusier.* Barcelona: Ingoprint, 1992.

Coates, Paul; Healy, Neal; Lamb, Chris; and Voon, Wi. "The use of Cellular Automata to explore bottom up architectonic rules", Proceedings of the Eurographics UK Chapter 14 Annual Conference, Imperial College, London, UK, (1997): http://www.ic.ac.uk/rer01/cfp.html

Coates Paul. and Makris Dimitris. *Genetic Programming and Spatial Morphogenesis.* London: CECA, 1999.

Coates Paul, Thum Robert and Walker Miles. *Generative Modeling Workbook.* London: CECA, 1998.

Corbusier Le. *Towards a New Architecture.* New York: Dover Publication, 1986.

Coyne, Richard. *Logic Models of Design.* London: Pitman Publishing, 1988.

Hillier, Bill and Hanson, Julienne. *The Social Logic of Space.* Cambridge: University Press, Cambridge, 1988.

Jean-Nicholas-Louis Durand, *Partie graphique des cours d'architecture faits a l'Ecole Royale Polytechnique* . Paris, 1821.

Knuth, Donald E. "Computer Science and Mathematics", *American Scientist* 61(1975):709

Koning H. and Eizenberg J. "The Language of the Prairie: Frank Lloyd Wright's Prairie Houses" *The Environment and Planning B8* (1981): 295-323.

Mitchell, William. *The Logic of Architecture.* Massachusetts: The MIT Press, 1990.

Stiny George "Introduction to Shape and Shape Grammars" *The Environment and Planning B7* (1980): 343-51.

Stiny George and William J. Mitchell "Counting Palladian Plans" *The Environment and Planning B5* (1978b): 189-198

Wojtowicz Jerzy and Fawcett William, *Architecture: Formal Approach.* London: Academy Editions/St. Martin's Press, 1986.

Wright Frank Lloyd. *The Natural House.* Ohio: Meridian Books, 1960.