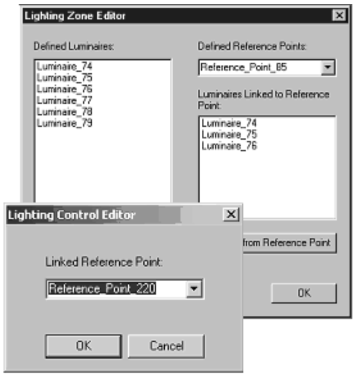

Figure 8: Modeling lighting controls

Current Status and Future Directions

Currently, the BDA software is in its 3.0 release and is available free of charge from http://gaia.lbl.gov/BDA. Main new features since the 2.0 release include a new electric lighting module (ECM), DOE-2 parametric runs for window size optimization and dual unit capabilities (English and Metric). Work is already underway for the next version, which will include links to the RADIANCE day/lighting simulation and rendering software (Ward & Shakespeare 1998; http://radsite.lbl.gov). Plans for future versions include links to additional simulation tools and databases, e.g., cost estimating modules, building rating systems, environmental impact and life-cycle costing modules, as well as links to commercial CAD software and electronic product catalogs from manufacturers of building components and systems.

Acknowledgements

This work was supported by Public Works and Government Services Canada and the Panel on Energy R&D of the Federal Government of Canada, through Enermodal Engineering Ltd. of Kitchener, Ontario, Canada and by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Research and Standards of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

References

Birdsall, B.E., Buhl, W.F., Ellington, K.L., Erdem, A.E. and Winkelmann F.C., (1990). "Overview of the DOE-2 building energy analysis program, version 2.1D." Lawrence Berkeley Laboratory report LBL-19735, Rev. 1, Berkeley, CA.

Hitchcock, R. J., (1995). "Advancing lighting and daylighting simulation: the transition from analysis to design aid tools", Proceedings of Building Simulation '95, International Building Performance Simulation Association.

IES (1993). "Lighting Handbook: Reference and Application". 8th Edition, Illuminating Engineering Society of North America.

Ward, G. and Shakespeare, R. (1998). *Rendering with Radiance: The Art and Science of Lighting Visualization.* Morgan Kaufman, San Francisco, CA.

***Vineeta Pal, Ph.D.** and **Konstantinos Papamichael, Ph.D.** are researchers at the Building Technologies Department, Lawrence Berkeley National Laboratory*

DESIGN ALGORITHMS AFTER LE CORBUSIER

Abimbola Oluwatoni Asojo

Abstract

Some views of design are the act as puzzle making, problem solving, evolutionary, and decision-making. All these focus on form generation as constructive, therefore characterizing design as a path-planning problem through a space of possibilities. Design problems consist sets of information divided into initial, intermediate, and goal states. Design in its simplest state consist of a set of operators, sequences (or paths) between initial and goals states. In this paper, I present design algorithms for Le Corbusier because of his distinct compositional techniques particularly for his "White Villas" in which some elements have been identified to recursively occur.

Introduction

According to Violet le Duc, "the first condition of design is to know what we have to do, to know what we have to do is to have an idea, to express this idea we must have principles and form that is grammar and language. Design literally has to do with the ability to read and write architecture. The skill of reading is more than just appreciating and intuitively recognizing the formal quality of the object. Reading architecture has to do with the process of parsing or resolving the elements and parts of a building and describing them grammatically. Reading architecture is a prerequisite to writing, constructing or composing it."[1] The impact of computers in architecture is parallel to the introduction of printing and engraving in the Renaissance. With Cad feedback through the storage, evaluation, and manipulation of formal constructs from recent and distant past, the instant interaction, and access of many designers to common but continually upgraded database is possible.

The achievement of computer-aided design (CAD) in the field of architecture compared to other fields may seem trivial to date, but current research is transforming the use of computers in design. Current computer-aided design (CAD) practice is rooted in an approach that utilizes computers as mere drafting and rendering machines, offering little or no insight to design. They advance technically to higher speeds and more intuitive displays presenting little theoretical insight to architecture.

Research currently going on in universities around the world is however establishing the computer as a form generating system. This school of thought defines the design world to be fundamentally characterized by its shapes, vocabulary, and operators, and thus, architectural composition can be segmented into a vocabulary of extensively and finely differentiated relations and functions. Rules for the interrelationship of these parts are defined and the rules are programmed in such a way that the program determines the evolution of the form.

In this paper, I present a grammar for the designs of Le Corbusier (1887-1965), starting from his basic domino system arrangement, the column grid, slabs and stair arrangement to his much acclaimed five points of architecture: the pilotis, roof garden, free plan, ribbon windows and free facade. All these are expressed as rules and algorithms are developed in

Autolisp, Autocad's programming language. The computer is used to investigate how a wide range of possibilities can be generated based on Corbusier's architectural style. The strategies for segmenting his architectural composition into parts are based on his main elements and their interrelationships, which are the rules. These are input as points, lines, volumes, shapes, primitives, and the data is subject to a wide range of operations based on their properties, functions, and relations. The use of randomness is explored through imputing the rules such that the program determines the evolution of the design, therefore, exploring architecture as a self-generating system. This idea of randomness is explored by using random numbers thus producing an element of unpredictability.

Generative theories in Design: An Analysis of Precedents
"A complete design is the end product of a process. What does the process start with, and how does it move from the starting point to the actual design? Suppose it starts with a vocabulary of well-defined physical components. The process of design involves selecting components from the vocabulary, placing the first one, and adding the others successively." [2] Mitchell notes "it is an ancient idea that knowledge base about some topic can usefully be structured as a collection of axioms, facts, and rules, we can think of the axioms as fundamental assumptions or basic definitions, such as those set forth at the beginning of Euclid's geometry."[3] Historically since 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) suggested ways in which sets of potential plans and elevations combinations can be generated.[4]

20th century architects used three-dimensional volumes in design composition. For examples, 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 Wright's designs emerge from a process of taking simple volumes and intersecting them in space to produce design form. More recently, Coates, Healy, Lamb, Voon have employed generative theories in generating limitless instances of forms by random growth, addition, and decomposition.[5] Generative theories involve 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 therefore has two aspects firstly, the number of components and secondly, 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 concept involves the computation of algorithms, algebra and variable combined with design knowledge. Generative theories rely on concepts such as top down/bottom up approaches, linear/non-linear systems, shape grammars, cellular automata, and computing and design.

Top Down versus Bottom up approaches to design[6]
In top down design approach, the designer starts with an abstract geometry, which is elaborated until it is transformed into form. Wojtowicz and Fawcett interpret Le Corbusier's Villa Savoye in a top down fashion.[7] The process begins with a square, which is divided into square grids. The square is extended along one axis to determine the major axis. The two floors are considered separately. On the ground and first floor, walls and circulation elements are added. Columns are located at the intersection of the grid lines, some displaced along grid lines, and some omitted on the first floor. Partitions and openings are added to complete the plan. In the bottom up design approach, the design process begins with fully detailed components. The process of design involves a selection of components from the vocabulary, placing the first one and adding others successively. Wojtowicz and Fawcett demonstrate the process using Louis Kahn's Richards Medical Laboratory.[8] The vocabulary of design comprises the square laboratory tower, a main core, and a secondary core. The building form is established sequentially by placing the main core block first, the laboratory block, and the secondary tower. The approach relies on knowledge of formal and functional characteristics of a given design vocabulary and compositional techniques. Both approaches rely on sets of relatively simple rules, which result in complex assemblies, i.e. complexity from simplicity. Basically rules distinguish a random pattern of objects from a significant design.

Linear versus Non-linear Systems[9]
Linear systems are those for which the behavior of the whole configuration is a sum of behaviors of parts, while in non-linear systems, the behavior of the whole is more than the sum of its parts. Linear systems are those which obey the superposition principle, they can be broken down into parts and analysis of these parts are possible. Once the individual parts are understood in isolation, a full understanding of the system can be achieved by composing the knowledge of the isolated parts, thus, the key feature is that by a study of parts in isolation, it is possible to learn everything about the complete system.

Non-linear systems do not obey superposition principles. Even if the system is broken down to simpler constituent parts and even if an understanding of the constituent parts can be achieved in isolation, it is impossible to combine this knowledge in understanding the entire system. The major features of Non-linear systems are their primary behaviors of interests are properties of the interactions between parts, rather than being properties of the parts themselves. These interaction properties disappear when the parts are studied independently. Thus, analysis is most fruitfully applied to linear systems. Such systems can be taken apart in meaningful ways, the resulting pieces solved, and the solution obtained from solving the pieces can be put together such that one has a solution for the whole system. In contrast, in non-linear system the constituent parts synthesize the behavior of interest.

Shape Grammars

Frank Lloyd Wright stated in the Natural house that "every house worth considering as a work of art must have its own grammar".[10] He also stressed the importance of consistency in grammar and the importance of a design having a language of their own. Stiny defines shape grammars as "a set of initial conditions, a lexicon of primitive objects, and syntax of transformations on those objects".[11] Wojtowicz and Fawcett 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."[12] The process of grammar expressed as production rules facilitating form generation has been demonstrated by producing line drawings that resemble those of Palladio by Stiny and Mitchell, and Frank Lloyd Wright's villas by Koning and Eizenberg.[13] Shape grammars have also been employed by Richard Coyne to describe algorithms for performing arithmetic operations on geometric entities called shapes.[14] Shape grammar relates with the designer's process of exploring design elements and principles. Wojtowicz and Fawcett note that an "ideal expert system for architectural design should be able to store objects, both primitives and composite, it should be able to use them in different strategies i.e. composition and decomposition." [15]

Coates and Makris have incorporated shape grammars and genetic programming in spatial composition by starting with sets of geometrical structures and their relationships.[16] Their approach uses genetic programming with a library of objects in a genetically breed 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 an algorithm in the computer for exploration of alternative solutions. Shape grammars can employed in generating and composing architectural objects as generating or analytical tools. It can be employed in reading and writing architecture. It can be used to represent the rules governing the setting out of building components.

Cellular Automata[17]

Cellullar automata were developed in the 1940s and this coincided with the development of computers. It was developed to explore mathematical issues. A cellular automaton begins with simple rules and explores how they may be expanded to form complex relationships. It consists of an array of cells, the state of each cell depends on the state of the neighboring cells. Cells can be at any one of a number of states like dead or alive, on or off, red, blue or green etc. A definition of the neighborhood of a cell and a set of transition rules will determine the state of that cell as a function of the state of cells within the neighborhood. All cells are updated simultaneously. Rules are divided into those that lead to inactive state and those which lead to a chaotic state. A finely tuned combination of rules can lead to a state where highly complex transient structures exist. The state between order and chaos is essential to all self-ordering systems capable of evolution. Generative descriptions involve a finite set of relatively simple rules which are repeated over and over, and the order in which they are repeated can have a marked influence on the final form of the entity.

Computing in Design

Over a quarter of a century ago the American composer Hiller used an early Illac computer to compose a piece of music by feeding into it 16th century rules of modal counter point and 20th century ones relating to serialism.[18] Notes that broke the rules were excluded from the program, so the computer chose at random the remaining possibilities. Currently, in music computers are used not only to aid pre-compositional calculation, but also to produce sounds. This is similar to current computing trends in the field of architecture, which explores how a wide range of design solutions can be generated by teaching the underlying principles of design as rules to the computer.

In the sixties, there were numerous efforts to bring architecture a substantial input of theory. The design movements followed the precedent of natural sciences. Its ideal was to use the new computers to discover a compact, universal set of precise laws that would explain scientifically the empirical facts of architecture. Its failure proved that architecture cannot be modeled as a natural science because the products of design are a matter of choice not necessity. This led to the exploration of modeling architectural theory as artificial intelligence. One of the key ideas of artificial intelligence was that people do not rely on super speed calculation to solve problems but on knowledge. Instead of programs consisting of numerous concise equations, the computer became a vast storehouse of knowledge, held in small independent pieces which can be incrementally modified or added to. Two procedures of importance were firstly, the generation and testing, where conclusions are generated and then evaluated for correctness. Secondly, constrained generation, where only a subset of possible solutions are examined.

Generative approach to computer aided design that has been quietly developing in universities around the world considers design as an act of search in the realm of spatial compositions and sets as its task the discovery of formal machinery approach to express design. In contrast, current computer-aided design (CAD) practice and search share little in common. In computer-aided design (CAD) systems, the data structure and its associated operations establish the design world. There are records to store information specifying points, vectors, arcs, polygons and other graphic tokens. The computational devices relevant to this project involve the use of the computer to generate alternatives by mechanically applying shape rules.

Research Models

The following two experiments illustrate form generation in Autolisp, AutoCAD programming language.

Model I: Parametric Autolisp Routine for Le Corbusier

In Le Corbusier's own summary of his main architectural elements he identified his buildings to be made up of the five points of Architecture; Pilotis, Roof Garden, Free plan, Ribbon Windows and Free facade.[19] The Pilotis raised the building off the ground into the air and the space underneath used for parking cars, road or gardens. Space lost was replaced on the top by a roof garden. The Roof Garden was a space open to the sky, containing greenery with view all round. Free planning was possible since the frame carried the weight partitions were organized independently e.g. some were curved to express their freedom and function. Ribbon Windows were 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 re-emphasizing the inherent potentials of a frame. These five points were often used 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 thirdly as mass penetrated Figure 1.

The buildings in Table 1 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.

Some of the other design configurations used by Corbusier as the Interlock system, which had elements locked around the service stack.[20] 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 identified above and separated into the following main categories:

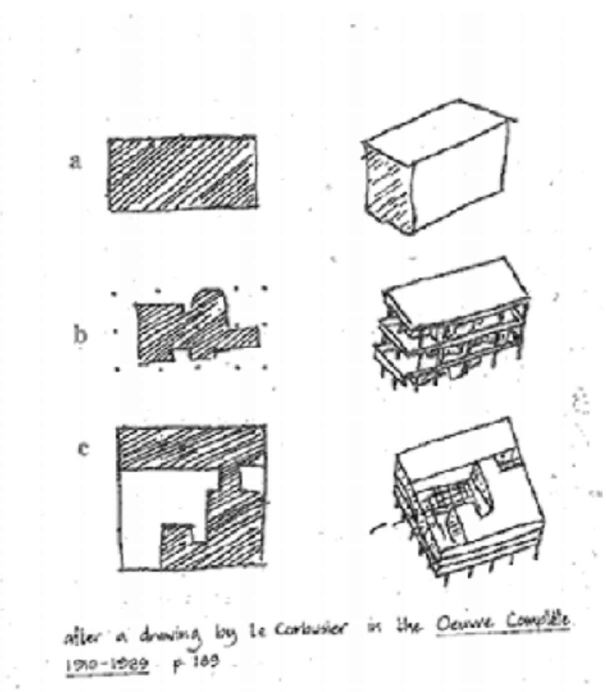

Figure 1. Le Corbusier's Design Configuration

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

These categories were further elaborated into rules specifying for example points, vectors, polygons and other graphic token 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 the X and Y-axis. A represents the distance between the columns in the X axis and 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).

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 Y direction if the number of columns in X was greater than in Y and vice versa.

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.

Table 1 – summarizes major elements from Six of Corbusier's buildings

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.

Circulation

The vocabularies of circulation elements were dogleg, spiral, straight flight, apsidal staircases, and ramps as identified in Le Corbusier's buildings. Figure 3 illustrates rules vocabulary of circulation elements.

Rule 1. Spiral staircases are placed in buildings with two columns in 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 in a three by five column grid.

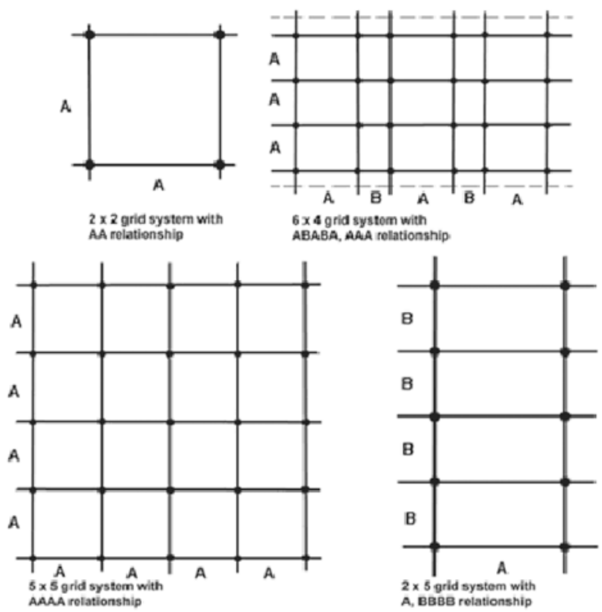

Figure 2: Grid Rules (Source: Author, 1995).

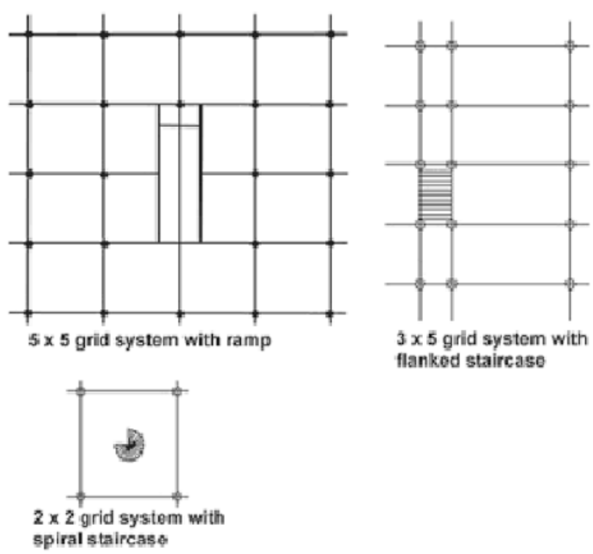

Figure 3: illustrates rules vocabulary of circulation elements (Source: Author, 1995).

External Wall Placement

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

The following rules applied:

Rule 1. If the number of floors equals two, then 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 is open for customization 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 the programmer 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 such that the program determines the evolution of the design thus exploring 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. The program then 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 4 illustrates some of the "Corbu" like prototypes generated (Insert Figure 4 here).

Main part of 16 page Autolisp Routine – This routine is based on the arguments and rules which were derived from the following categories –

Orthogonal Cage determination, Circulation, External walls and main curved elements of interior

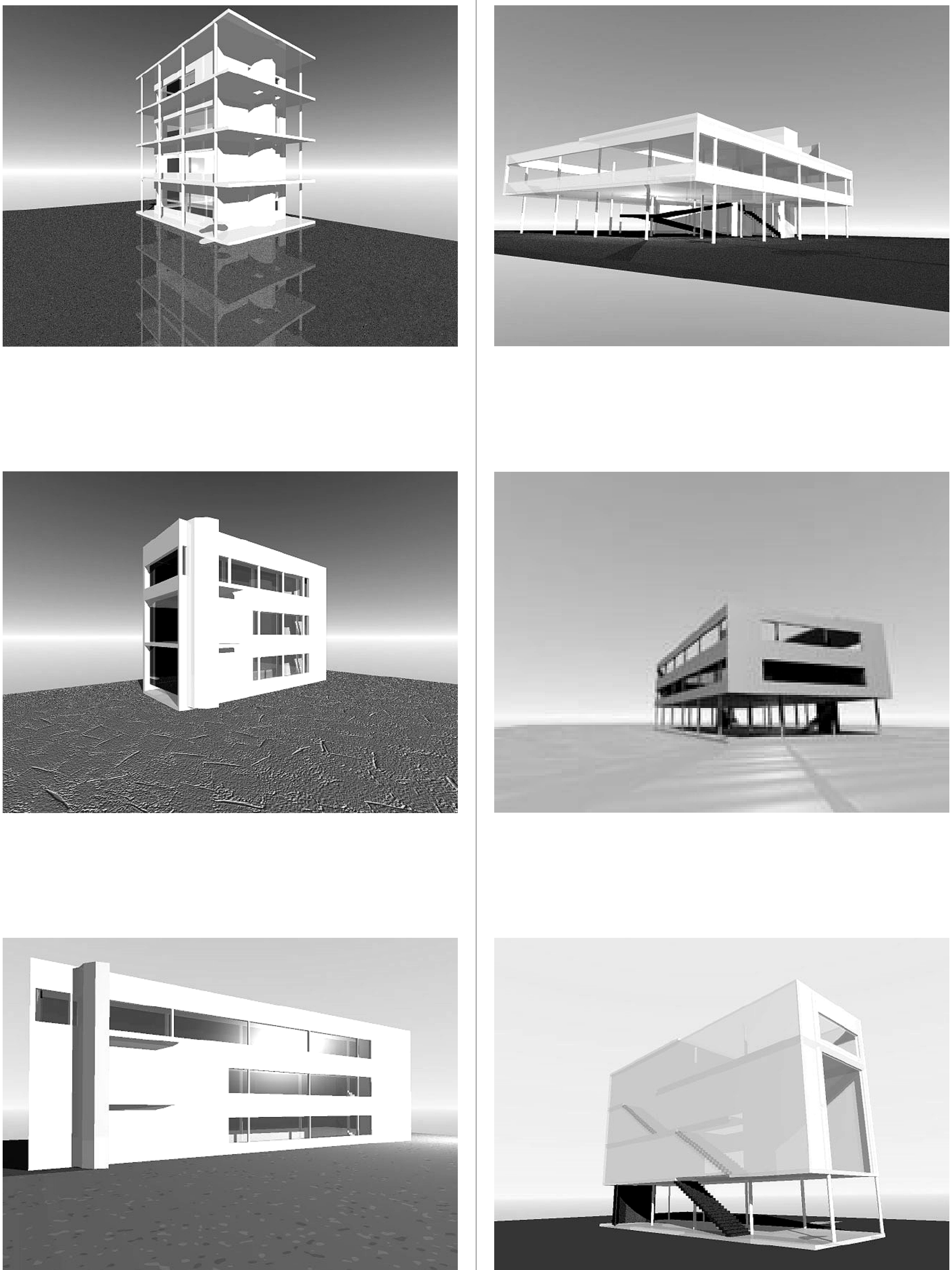

Figure 4: illustrates some of the "Corbu" like prototypes generated (Source Author, 1995).

```
 (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"))
```

The Autolisp routine generates a wide variety of similar configurations to Corbusier building types and supports the idea of generating complex outcomes from simple rules. The designer also has available for them some unexpected design options.

Model II: Genetic Programming

This section 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 depends 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 and Makris notes "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." [21] 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. 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.

Program Rules.

This model starts with some basic configuration 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 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 were 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 5 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 generates column grid arrangements similar to Le Corbusier's designs. The resulting forms at this stage illustrate that an evolutionary approach is related to the process of generating composition. The Autolisp routine presents configurations that are similar to Le Corbusier's and the mutations are driven by visual judgment, which encourages co-operation between the computational power and human creativity. This model raises some critical issues unlike the first model where the process is linear. This model encourages the human-technology interface, since the designer is responsible for selecting the parents in the genetic algorithm. Coates and Makris notes "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." [22]

Conclusion

Coates and Makris notes "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." [23] Exploring

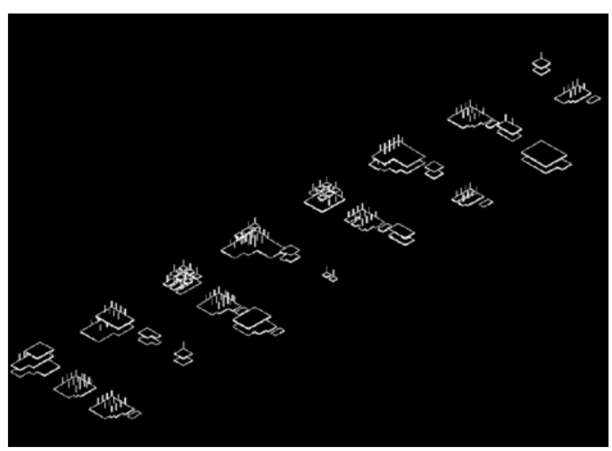

Figure 5: illustrates generation from GP for Le Corbusier -Random seed 1, Probability of mutating 2, No of Generations 2, Parent of Row 2 = 1&2, Parent of Row 3&4 (Source Author, 2000).

design algorithms present an opportunity to examine a wide variety of possibilities and unexpected alternatives. The process highlights the importance of rule based systems as an integral part of the design process, and rules can be modified to systematically to define new language of design that reflect changing circumstances and incorporate 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 elements. The process contributes to the human technology interface debate and contributes to the process of utilizing CAD systems in generating design form 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 is 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, this methodology can be explored in studio and digital media classes to assist form generation. I am not proposing to diminish the designers capabilities 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.

Notes

1 Violet Le Duc, Eugène-Emmanuel, *Dictionnaire Raisonne de l'Architecture Francaise* (Paris: 1866)

2 Wojtowicz, Jerzy and Fawcett William, *Architecture: Formal Approach* (London: Academy Editions/St. Martin's Press, 1986), p. 26.

3 Mitchell, William, *The Logic of Architecture* (Massachusetts: The MIT Press, 1990), pp. 74-75.

4 See Jean-Nicholas-Louis Durand, *Partie graphique des cours d'architecture faits a l'Ecole Royale Polytechnique* (Paris, 1821) for more detailed discussion.

5 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://ceca.uel.ac.uk/cad/ paper_bits/paper.html

6 Wojtowicz, Jerzy and Fawcett William, *Architecture: Formal Approach* (London: Academy Editions/St. Martin's Press, 1986), pp. 26-36.

7 ibid

8 ibid

9 Coates, Paul, Thum Robert and Walker Miles, *Generative Modeling Workbook* (London: CECA, 1998), p. 10.

10 Wright Frank Lloyd, *The Natural House, Meridian Books* pp. 296-297.

11 Stiny George "Introduction to Shape and Shape Grammars" *The Environment and Planning B*7 (1980): 343-51.

12 Wojtowicz, Jerzy and Fawcett William, *Architecture: Formal Approach* (London: Academy Editions/St. Martin's Press, 1986), pp. 43-67.

13 For a more detailed discussion on Palladian plans see Stiny George and William J. Mitchell "Counting Palladian Plans" *The Environment and Planning B5* (1978b): 189-198. See also Stiny George and William J. Mitchell "The Palladian Grammar" *The Environment and Planning B5* (1978b): 5 - 18. For details on Frank Lloyd Wright's villas see Koning H. and Eizenberg J. "The Language of the Prairie: Frank Lloyd Wright's Prairie Houses" *The Environment and Planning B8* (1981): 295-323.

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

15 Wojtowicz, Jerzy and Fawcett William, *Architecture: Formal Approach* (London: Academy Editions/St. Martin's Press, 1986), p.46.

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

17 Coates, Paul, Thum Robert and Walker Miles, *Generative Modeling Workbook* (London: CECA, 1998), p. 10.

18 *The Concise Oxford Dictionary of Music Oxford: University Press* 1980 p.142.

19 For a more detailed discussion on the five points of architecture see Le Corbusier, *Towards a new architecture* (London: Dover Publications, 1986).

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

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

22 ibid

23 ibid

***Abimbola Oluwatoni Asojo** is an Assistant Professor in the College of Architecture, University of Oklahoma*