ALGORITHM AIDED ARCHITECTURAL DESIGN (AAAD)

House gen: fallingwater toolbox v 1.0

MAHA H. EL GEWELY
Misr International University (MIU), Cairo, Egypt.
Email address: maha.elgewely@miuegypt.edu.eg

Abstract. Algorithm Aided Architectural Design (AAAD) is considered a second paradigm shift in the Architectural design process after the first one of bridging the conventional design process to the digital realm of design. This paper is divided into two parts, the first part comprehends the Algorithmic Architecture approach from the point of view of tools, techniques, theories and practice in order to find the Algotecture theories on the map of Digital Architecture. Thus, the paper exemplifies an application on Algorithmic Architecture. FALLINGWATER TOOLBOX VERSION 1.0 is a computational design demo tool for architects to aid in the house schematic design phase according to an analytical study of Frank Lloyd Wright's basic design rules and spatial program of his masterpiece; FallingWater House, (Kaufmann House 1939). These rules have been transferred to algorithms and code thereafter. At a preceding stage, the Graphical User Interface was developed using MAXScript 9.0. Using the FALLINGWATER TOOLBOX, infinite number of house prototypes can be generated within few minutes. Although, the FWT is based on a hypothetical design problem of producing prototype alternatives for a new house with the same identity of the Edgar Kaufmann House, the concept of the tool can be applied on a wider range of problems. It may help generating prototype alternative solutions for residential compounds design according to the required constraints.

1. Introduction

The use of computer in different phases of Architectural Design has been the main concern of many researchers in the field of Architecture. The big dream beyond is to reach a complete digital design process. Successful endeavors, that gave acceptable results and have been developed to be complete software, were in the phases of programming, design development,
drafting and visualization. Other phases of design process; such as functional relationships, form finding and manufacturing; still have many trials to reach complete software to aid in the knowledge-based Design, Scripted Architecture and CAM, in respect, towards the big dream. Algorithmic form generation for architectural purposes is not yet familiar to the professionals in this field. On the contrary, it is widely used in other fields of engineering where three dimensional shapes are more strictly expressed as mathematical equations (Gross, 2001). In this research, the aesthetical part of Architecture is tried to be translated to rules through a process of bridging from Reading Architecture to Algorithm Writing.

2. Algorithm Aided Architectural Design (AAAD)

Algorithm is a detailed sequence of procedure that can solve a problem (Terzidis, 2006). Algorithms can be used to solve problems in our daily life, including Architectural Design problems. This means that Algorithm can be considered as a Design Stakeholder which moves from the simple (CAAD) to the new term Algorithm Aided Architectural Design (AAAD). The new concept aims at transforming different constraints that rule the design; including sustainability rules and custom rules; into a digital sequence of digital steps. Using AAAD enhances the design process, gives a wider range of solutions and enriches the final product of the design process.

3. From Digital Architecture to Algorithmic Architecture

In a simple trial to locate Algorithm on the map of Digital Architecture, Computational Generative Design Strategies have been studied from the point of view of different theorists. By now, there is no ultimate classification of the design strategies leading to forms generated by computer. Key names in Digital Architecture theory, each has his own classification. Only classifications which lead to Algorithm are included in this study. There are no conclusive borderlines between the different approaches of the theorists; but they assemble at the focus point of Algorithm.

4. Application

As a practical application on AAAD, FallingWater Toolbox Version 1.0 is a computational design demo tool for architects to aid in the schematic design phase according to Frank Lloyd Wright's basic rules and spatial program of his Falling Water House. The graphical user interface was
developed using MaxScript 9.0. Using the FALLINGWATER TOOLBOX, infinite number of house prototypes can be generated within few minutes.

3.1. WHY FALLING WATER HOUSE?

The FallingWater Toolbox Version 1.0 is based on a hypothetical design problem. Another prototype of the Falling Water house is assumed to be built facing the original building in the same context with the same identity and form characteristics. This hypothetical problem simulates the common daily design issue in several studios which is to produce different prototypes for residence compounds. They have to have the same character, reflect equality and give harmonious visual impression. A masterpiece for a pioneer architect has been chosen to be benchmark reference to evaluate the final product of the experiment which is a step forward towards another software to design prototypes according to the designer own rules and principles. The set of rules of the toolbox have been put according to an analytical study of the original design of the Falling Water according to the following Scenario.

3.2. SCENARIO

3.2.1. Analytical Study

The Falling Water analysis focuses on the transformation of the human
Figure 3. From Analysis to Rules
perception, of the spatial relationships and visual characteristics of form, to a set of rules and thresholds which could be converted to an algorithm and to code that could be feed to the computer afterwards. As an analogy of Francis Ching analysis of form, space and order, this analysis is divided into four categories: Spatial System (Area Analysis and Spatial relationship Analysis), Enclosure System/Form Analysis, Structural System and Circulation System. At a preceding stage, the major design characteristics that form the masterpiece identity are tabulated and transferred into rules which algorithms will represent later.

3.2.2. From Analysis to Rules
This phase is a transitional stage between the architecture reading and the algorithm writing to compose an Algorithmic Architectural form and space according to a set of rules which is concluded from the analysis above.

3.2.3. Set of rules
All rules from the previous phase are re-arranged and classified to three categories of values: Constant Values, User Entry Variables and Random Values.

<table>
<thead>
<tr>
<th>Constants</th>
<th>User Entry Variables</th>
<th>Random Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Number of levels (No):</td>
<td>1. Land size dimensions (X and Y)</td>
<td>1. Terraces’ area (range between 15 to 65 m2)</td>
</tr>
<tr>
<td>2. Space distribution among different levels with respect to the functional relationships.</td>
<td>2. Areas of first level spaces within the predefined range (5 Entries)</td>
<td>2. All X and Y shifts of all rooms</td>
</tr>
<tr>
<td>3. Three types of staircases: (suspended, external and indoor)</td>
<td>3. Areas of second level spaces within the predefined range (4 Entries)</td>
<td>3. Stochastic search in positioning objects</td>
</tr>
<tr>
<td>4. Staircase dimensions</td>
<td>4. Areas of third level spaces within the predefined range (1 Entry)</td>
<td></td>
</tr>
<tr>
<td>5. Number of terraces along each level (Three Three, One)</td>
<td>5. Adding bedroom to the second level</td>
<td></td>
</tr>
<tr>
<td>6. Four inclined supports at the ground level to lift the building</td>
<td>6. Adding a bedroom to the third level</td>
<td></td>
</tr>
<tr>
<td>7. Vertical chimney and vertical glass shaft</td>
<td>7. Number of prototypes to generate</td>
<td></td>
</tr>
<tr>
<td>8. Wide internal walls (50 to 60 cm width)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. The Set of Rules

3.2.4. Algorithm
The set of rules is to be transferred to an extended flow chart that represents the function algorithm of the tool. A small part of this flow chart is illustrated below.

3.3. IMPLEMENTATION
FWT is to be called by pressing a button on MyScripts toolbar in 3DMax Interface. The FWT combines two tabs; the first tab is titled Form Generation. It has 4 rollouts, as illustrated in Fig 6:
1. First Level Spaces  
2. Second Level Spaces  
3. Third Level Spaces  
4. Prototypes Generation  

A range is used to define how many prototypes to be generated in the entire experiment because one or more prototypes may be aborted or invalid. Booleans as a stochastic search technique is based on finding a possible position without overlap for each object which may be impossible after a predetermined number of trials that lead to exit the loop. The Second Tab contains information about the original Falling Water House design and the FWT.

![Algorithm Flow Chart](image)

Figure 5  A small part of the Algorithm Flow Chart

3.4. OUTCOMES  
The experiment has been tried several times. Ten successful prototypes are generated in 6.7 seconds, twenty successful prototypes are generated in 11.7 seconds, 30 prototypes are generated in 18.7 seconds with two aborted trial and 40 prototypes are generated in 24.9 seconds with two aborted trials.

In one experiment, forty prototypes have been generated within 24.9 seconds. The final product is different patterns / tectonics that reflect diversity and asymmetry. None of the resulted tectonics coincides with the original design of the Falling Water House. Some designs are compacted while others are extended. They may suit different site conditions.
3.5. FINDINGS AND RESULTS
The few parameters used in this application as user entry variables are not enough to support interaction between the user and his computer. Using Stochastic Search reveals disadvantages of this Algotecture Technique. As it is based on random search and calculating overlap till reach a certain condition, invalid trials have a higher opportunity to happen. Evaluation and
development processes are not supported by the tool. The user manually evaluates then develops the alternative which is appropriate to the design problem.

5. Conclusion

AAAD approach of design gives a birth to a second paradigm shift in the Architectural Design Process after the first one of bridging the conventional process to the digital realm of design. The new trend causes a dramatic change in the Architectural tools, techniques, language and subsequently the form generated by applying this process. Hence, architects should extend their human intellect by updating their traditional use of the computer to a new one that utilize this machine to the maximum power and involving it in the creative part of the design process.

Till now, Algotecture approach is still limited to the academic community. Although, many trials have been implemented in the realm of Algotectural practice, none of them has bridged to the Architectural market as commercial software. Therefore, Algotecture theories are still boiling and producing more and more notions of Architecture on either the typological or the topological level.

As the main obstacle of Algotecture is the transformation of what is aesthetical to an encoded procedure, one can realize that starting with a physical building that has a dominant identity makes the mission more relevant. The process is parcelled into a three steps; reading architecture, from analysis to rules and algorithm writing. That’s why a masterpiece is selected as a kernel of the application tool that can thereafter be extended to enclose more general methodologies which can be applied on a wider range of problems.

6. Future Work

FallingWater Toolbox version 1.0 is to be updated and developed following few stages;

- More interactive parameters that related to the context, site restrictions and building code to be supported by the toolbox that can mature the applied constraints and thereby enhance the final product.
- User entry variables should have more control on the identity of the generated alternatives.
- Empower the user to access the design rules and constraints to widen the range of design problems that can use this application and support the mass customization concept.
- Extend the toolbox application to include more design methodologies for other star architects and their masterpieces.
- Extend the toolbox concept to be applied for educational purposes.

References
