Form Generator

A CAD tool for conceptual design development

Birgül Çolakoğlu¹, Gökhan Keskin²
¹,² Yıldız Technical University, Design & Computation Unit, Turkey,
http://www.bot.yildiz.edu.tr
¹bigi_c3@yahoo.com, ²gokhan.keskin@yahoo.com

Abstract. This paper presents a new computer-aided design tool named Form Generator (FG). It is developed within a visualization software as a scripted utility. It aims to assist designers and students in creating and exploring abstract form compositions.

Keywords. Generative design; shape grammars; rule based design.

Introduction

Shape grammars invented by Stiny and Gibs have been in use for over three decades. Their utility in design generation and analysis has been well documented. They have well developed mathematical foundation that provides formal mechanism for their computer implementations. Several tools for 3D shape grammar have been developed such as 3DSherher (Wang and Duarte, 2002), QShaper (Yazar and Colakoğlu, 2007). These applications were developed as teaching aids, with the potential of becoming a more powerful design tool.

Computer applications of 3D grammars have been originated from Stiny’s (1980) “Kindergarten grammars: designing with Froebel’s building gifts” study. Frobel developed his method as a series of geometrical gifts and a system of categories of geometrical forms. Stiny transformed it into vocabularies of shapes and a system of shape rules that are specified in terms of spatial relations between shapes in the vocabularies. (Knight, 1991; Stiny, 1980) These shape grammars are the first defined in a three-dimensional space, laying the groundwork for three-dimensional user defined computer implemented grammars.

3D shaper is the first computer application of 3D grammars, that allows exploring the design generated by labeled spatial relations between two orthogonal shapes, such as pillars, oblongs and cubes. It is developed as independent software on java platform. QShaper, utilizes the same orthogonal shapes as design vocabulary. It operates on maximum user interaction. This involves a user choosing which rule to apply and how to apply it in each step of computation. QShaper is developed as a scripted utility using the software’s built-in scripting language. It allows a designer to define shapes and rules, and to manipulate them in the design process synchronously.

This paper describes an ongoing tool development named Frobel Grammar interpreter “Form Generator” (FG) and its utilization as design aid in the conceptual design process. The FG is developed within common visualization software as a scripted utility on the base of a five stage program given by Stiny for creating new form compositions: a vocabulary of shapes, spatial relations, shape rules, initial shape, and grammars (designs). It functions as an experimental tool for rule based design form explorations.
Form Generator

FG is developed as a scripted utility using conventional 3D CAD software's built-in scripting language. It aims to be developed as design aid for conceptual stage of designing. FG operates on predefined number of vocabulary, rule sets and the defined number of transformation points. It consists of four vocabulary (blocks), A1, A2, A3, A4, and three predefined rule set A1A2A1, A1A2A3A1, A1A2A3A4A1 defined through vocabulary relations as shown in Figure 1. The blocks are defined with three: X (width), Y (length), Z (height) parameters. The relations between the blocks are defined based on rotation degree on X, Y and Z axes as shown in figure 1. Once vocabulary, rules, and relations between vocabularies are chosen the number of rule iteration (defined as a loop) is given for design generation.

FG allows user to modify the design in any step of the iteration by assigning transformation point to iteration (loop) number.

Design modification is defined through 6 transformation points (0, 1, 2, 3, 4, and 5) as shown in figure 1. The user defines transformation points through loop (iteration) numbers. Figure 5 illustrates an example of design generation. The rule set A1A2A1 is chosen for design development. The XYZ variables define the parameters of A1, and A2. The relation between A1 and A2 is defined with translation and rotation on XYZ axes and visualized in CAD software as shown in Figure 2. The design shown in figure is generated with five iterations of the rule A1A2A1. Once the design is visualized the user analyzes it based on design requirements, defines transformation points (by assigning iteration number to the transformation point).

In illustrated example in Figures 3 and 4, the user iterates the rule 16 times. He applies one transformation point on 8th iteration. Here the block parameters and block relations of the rule are modified through user interface as shown in Figures 3 and 4.

FG can generate endless design form compositions. It gives control to the user in the generation process of the design allowing him to control step
Figure 2: FG works as plug in on 3D max.

Figure 3: Design generated using the rules defined in FG.
by step generation of the design by modifying the rules, and vocabulary parameters of the rules. The user interacts with the program in reciprocal manner in the process.

Utilization of FG for Housing Design

Each designer can have his or her own way of utilizing FG in the conceptual phase of the design. This paper describes the process of conceptual housing design development using FG through a case study conducted by the developer of the tool. Here the designer extrapolates his design process and describes his responsive design actions with FG tool. Due to length constraints of this paper the process is summarized in 8 steps:

1. Housing design brief;
2. Definition of vocabulary through user interface of FG. Four vocabulary elements are defined by entering their sizes via user interface (UI) of FG (Figure 5);
3. Definition of relations and rules;
4. Derivation of design alternatives by assigning different number of iterations and transforming designs, (Figure 6);
5. Decision of abstract design alternative for possible house design based on site requirements, (Figure 7);
6. Exploration of chosen design by changing the relations of the vocabulary through transformation points, (Figure 8);
7. Interpretation of the design, (Figures 7, 8, 9);
8. Configuration of housing program with the design, (Figures 10 and 11).

From step 2 to 6 the designer explores possible abstract form compositions by modifying rule relations, and parameters of the chosen vocabulary (illustrated in figure 8), in different transformations points. He generates 10 variations of abstract form compositions from the chosen rule set. Then he chooses the most promising form composition (illustrated in figure 10) for required function (house) for further development. Selection of form composition requires experience and the ability to interpret
Figure 5: Definition of vocabulary and the design rules.

Figure 6: Derivation of design alternatives.

Figure 7: Chosen abstract form design for further development.

Figure 8: Exploration of design.
abstract forms. Once the decision is made the designer modifies the form based on existing design criteria such as site, function, program, interior-exterior relations, structure, climate, etc. by using CAD software 3D modification tools.

**Conclusion**

FG is design aid tool specifically developed for form generation and could serve in introductory theory and composition classes, as well as for some design studio exercises. A good tool should allow the user to quickly explore a number of design possibilities as well as generate designs that may not have easily been considered using hand methods and should have easy to use interface.

FG is user friendly in terms of flexibility that it provides to the user however, the graphical user interface of it has not achieved user friendliness in terms of easy-use. An advantage of abstract form compositions generated through rules defined in FG is that they easily demonstrate the mechanics of shape grammars and do not limit the possible interpretations of the designs they generate. For beginning students, the connection between an abstract form and an architectural interpretation may be difficult to make. However, for an advance student they can open wide range architectural interpretation possibilities.

According to the first application tests provided in case study, FG is promising tool to be used in form
explorations in the conceptual stage of the design. FG is constrained with four vocabulary, three rule sets and five transformation points to explore automation of Frobel grammars in given algorithmic sequence.

The development potentials include more flexibility on the rule creation, and implementation of additional futures the visualization software provides.

References


