Generative Systems

Shape Grammar is a well-known formalism in CAD research. Briefly explain shape grammar, i.e.: what is a shape grammar? what does one do with one? What is attractive about the idea? Why is 'emergent form' potentially problematic?

Describe (using simple examples) the operation of a shape grammar interpreter - what are its inputs and outputs; what does it do? Compare a shape grammar interpreter and the inference engine in a rule based expert system.

What is a shape grammar?

Shape, as Stiny defines it is "a limited arrangement of straight lines (Stiny 1980)." A shape grammar is a generative system of shapes to represent form. It is derived from linguistics and computer science. In linguistics, a formal grammar (or simply a grammar) has rules for structure with starting symbols and several rewriting rules (Chomsky 1985; Partee and others 1990). A grammar is essentially a deductive system of objects and rules of inference to generate sentences of a language. A grammar, therefore contains a set of rewriting rules. These rewriting rules have the form A -> B. Both A and B are in the form of strings. The execution of a rule is that whenever A occurs in a string as a sub-string then this sub-string can be substituted by B and therefore the original string is changed. For example, if a rule A -> B, then a string of XAY can be replaced by a string XBY.

Here are some basic rules for constructing a sentence:

1. Sentence -> Noun Phrase  Verb Phrase  S -> NP + VP
2. Noun Phrase -> Article  Noun  NP -> Art + N
3. Verb Phrase -> Verb  Noun Phrase  VP -> V + NP
4. Article -> a, the, my, yours  Art -> a, the, etc.
5. Noun -> girl, boy, picture, house  N -> girl, boy, etc.
6. Verb -> buy, run, draw, make  V -> buy, draw, etc.
As shown in Figure 1, a sentence of "The girl draws a picture." can be derived from applying the above five re-writing rules. A Sentence can be substituted into a Noun Phrase and a Verb Phrase (rule 1). A Noun Phrase can become an Article with a Noun (rule 2). A Verb Phrase can become a Verb and a Noun Phrase (rule 3). A sequence of substituting Article, Noun and Verb into appropriate elements then form a complete sentence (rules 4-6).

Similar to grammar, a shape grammar, instead of dealing with one-dimensional word strings, it deals with two-dimensional or three-dimensional shapes. For example, the starting point can be a concept of "house," and all the substituted elements can be architecture vocabularies such as columns, beams and roofs, or rooms and space. A rule applies when a sub-shape from the left hand side matches a rule, then that rule can be applied to evolve a building design. For example, Figure 2 illustrates some simple rules for generating a house with its architectural elements.
Here are the rules of shape grammar for Figure 2:

1. House -> Roof + Body
2. Roof -> Roof Tiles + joists
3. Body -> beams + columns + base

The bottom shaded part of "supported" by are the implicit assumption of functional concerns that can be expressed as "on top of."

**What can we do with a shape grammar?**

We can use shape grammar for several things because it deals with form description, rule representation and type recognition. First, we can use shape grammar as a generative system to generate forms, to explore possible alternatives derived from the same set of rules. Second, we can use shape grammar to manipulate and adjust parameters of different variables to select fittest or preferable results of design. Third is to use computer to simulate constructing and deconstructing an artifact, a building through the use of rules and reversed rules (such as reverse right hand and left hand side). Fourth, we can explore reasoning in an abstract level about form and function.

**What is attractive about shape grammar?**

Many shape grammar have been written for architectural design such as Palladian villas, Japanese tea rooms, Frank Lloyd Wirght's prairie houses, Buffalo bungalows, etc. (Downing and Flemming 1981; Knight 1981; Koning and Eizenberg 1981; Mitchell 1990; Stiny and Mitchell 1078) The popularity of shape grammars can be attributed to the following reasons:

1. Shape grammar can represent a particular style of how a design has been evolved. After the grammar has been generated, then it is theoretically possible to produce many variations of un-built design and new possibilities of this style. For example, we can examine how a famous architect, such as Palladio, uses particular elements at a particular configuration. We can also use shape grammar to derive possible architectural intentions, such as elevated base for ceremonial appearance, and we can then produce designs that
use the same vocabulary in the same way to produce additional designs that respond to the same architectural intentions.

2. Shape grammar is a short hand description of form arrangement. Shape grammar consists of a small set of rules that can concisely specify a very large set of designs. For example, a set of finite rules (such as 41 steps) of a grammar can generate many (almost infinite) possible Japanese Tea room design (Knight 1981).

3. Once a form be depicted as a shape grammar then it is computable. The belief is that a machine can generate forms or a language specified by a shape grammar. Furthermore, if a grammar is suitable for generating, it may as well suitable to search for designs with some specified properties specified by the graphic re-write rules.

4. Shape grammar attempts to capture knowledge of design. (form, of course, function also attempted) and make elements substitutable. For example, round columns and square piers can all belong to the same class and are interchangeable because they perform the same type of function as supporting elements (though how much functional knowledge embedded in a shape grammar is arguable, there are claims about reasoning with form and function, e.g. (Mitchell 1990; Stiny 1985)). A side effect, however, in the processing of writing or applying shape grammar, we may discover new uses of familiar architectural vocabularies, or introduce new types of elements to extend the possibilities of that grammar.

However, design researchers have attacked shape grammars as pure theory for not taking functional, budget and site specific information into accounts. Some pointed out many claims of the usage of shape grammar are just illusion (Wang 1987). One obvious problem is the multiple interpretations of a shape, and emergence of sub-shapes within a shape.

Why is emergent form potentially problematic?
Emergence of shapes can be problematic for shape grammar generation because a figure can be interpreted in many different ways. For example, a cross shape can be ambiguous to read its sub-shapes as shown Figure 3. Figure 3a is the shape for pattern recognition which can be read as two overlapping rectangles, one vertical, one horizontal. Figure 3b interprets the cross figure as long and short horizontal and vertical lines. Figure 3c depicts the figure as sixteen equal length lines either is horizontal or vertical. Figure 3d represents the cross figure as five same size small rectangles. Therefore, each different representation will result applications of different graphic re-write rules, and generate different grammar. That is, the assumption of a style can be constructed by a grammar and deconstructed back to its original form will no longer hold true. The computation will become complex and need more rules to eliminate unwanted or inappropriate rules and transformation.

The operation of a shape grammar interpreter -- what are its inputs and outputs, what does it do? Compare a shape grammar interpreter and the inference engine in a rule based expert system.

In "Computing with Form and Meaning in Architecture " [Stiny, 1985 #205] Stiny presented a simple shape grammar that has only three rules (see Figure 4). The input, the initial shape is a small dot, and the final form is a pattern composed of several rotated and shrunk squares.
Knowledge-based expert systems have two parts, a general problem solving technique (symbolic reasoning) and domain specific knowledge representation. The general problem solving is the development of inference mechanism (Fenves and others 1994; Radford and Gero 1988) that has the form of IF -> THEN. The IF-> THEN statement is "IF a condition THEN a action." This inference engine controls not only the application but also selection (the order in which rules are selected to apply) of a rule in the problem solving process. Therefore, the inference engine consists of accumulated rules and related procedures. Each time a rule is applied, the action then will generate a new statement and feed back into the knowledge base and therefore change the situation and context. A rule can also requires a number of conditions be satisfied before an action is taken.

In short, the shape grammar interpreter applies a set of rules that have the A -> B form to substitute an original shape A into new shape B while rule-based inference engine takes a set of rules of IF -> THEN to update statement of current situation state. They both have the same form of left hand and right hand sides. They both apply the logic in the process. However, they do not have same representation of their symbols, one use geometric shapes, one is language statement.

References


