Shape grammar is creating and understanding design directly though computation shapes rather than indirectly through computation text or symbols.

As terry knight says “A shape grammar is a set of shape rules that apply in a step-by-step way to generate a set, or language, of designs. Shape grammars are both descriptive and generative. The rules of a shape grammar generate or compute designs, and the rules themselves are descriptions of the forms of the generated designs.”

Shape grammar is a set of rules and an initial shape. The rules apply to the initial shape and shapes produced by previous rule applications to generate designs.

The rules are designed to replace the current shapes. So the user of the rules, either human or computer should decide which rule to use and on which shape of the designed shaped. Since the designed shape may consist of different shapes, and there could be different rules that can apply to each shape.

Stiny has introduced shape grammar and description functions. Shape grammar is the different ways that we can arrange the shape and description function is the procedure for interpreting it.

One of the products that can be used as shape grammar in architecture is color grammar, which is similar to shape grammar but each rule has color. Colors can stand for different things, like material in the construction, or elements like doors, windows or….
Stiny, in computing with Form and meaning in architecture, has started the concept by explaining the procedure which is the way to carry out computation on data. After explaining procedure, we saw how shape grammar is one kind of procedure. So we saw as any kind of grammar, shape grammar also consists of rules and an initial shape. By applying the rules to the initial shape we can replace the shape and create different shapes.

The “Design operator” mostly has discussed the operator on the shapes. It shows how by applying different operators we can replace one shape with another. How we can instantiate from abstract shapes like: rectangle(x, y, x2, y2), line(X, Y, Theta, Length) or abstract from labeled objects like: chair(X,Y, Rotation). And after defining the shapes, different binary, and unary operators which are helpful in computing design are introduced, like rotation, translation, reflection, scaling and Boolean operations like union, intersection and many more operators have been defined for combining and making different designs …

**Conclusion**

I found the idea of shape grammar very useful because it gives the designer the ability to show their design process and design rules by shapes rather than text and mostly understanding shapes is easier than text. Also computers can be very helpful in this area. We just need to define the rules once and then computer can apply then very easily to different complicated shapes. So although shape grammar concept has developed very slowly in the last years, but I believe that it is very helpful in design and mostly in architecture design.

I liked both article.
Stiny’s article was very informative. But since the subject was very new at the time article was written, it was like an introductory to the shape grammar while “design operation” was more detailed about the subject, and completed Stiny’s article in understanding the rules in shape grammar also the way we can apply these rules in computer.