EXPLORING THE INTERACTION OF RULES WITH SHAPES AND THEIR BOUNDARIES

PAUL VARGHESE, ARIF N. MERCHANT
Department of Architecture & Regional Planning, Indian Institute of Technology Kharagpur 721302
Email address: paul@vsnl.com, merchant@arp.iitkgp.ernet.in

Abstract. The interaction of rules and the generation and interaction of shapes and their boundaries is explored. The effects on the design processes especially in the shape grammar formalism are examined.

1. Introduction

Design is perceived as a routine that has similarities with problem-solving, black-box methodologies or associated with creativity, innovation, etc. The design methods themselves are currently in a state of flux. Ground-breaking theories, systems and technologies have ensured that the processes also are innovation-oriented and novel. Not only is the product undergoing change but so is the method, especially with the introduction of the information sciences.

Design procedures, at the level of the architect, involve the visualization of form and the transformations of its various elements. It is an evolutionary process and is often trial-and-error or a search among the candidate options.

Historically too, one sees that designers adapt their working methods to take in new skills and technologies, and that these innovative methods soon become part of their repertoire. Inventive artists and designers get used to and update their skill-sets to move forward to changing methods and know-how. Sometimes they are forced to, or often enough advances force themselves on the profession.

1.1. DESIGNING WITH SHAPES

One of the newer methods of design is the shape grammar formalism, but which has not yet got sufficient attention. Shape grammars have been used to investigate designs (*noun*, as in analysis) as well as to design (*verb*, as in synthesizing) both of which are necessary. In a manner, it captures some of the methods used in design development and puts a logico-mathematical procedure in place that innovates and brings out the essentials of designing. However, not many perceive this as being an easily comprehensible even if fundamental, or perhaps an easily masterable design process.

2. Rules, Shapes and their Boundaries

Rules and shapes are fundamental to the shape grammar formalism. Boundaries are elementary to the shape. One explores the potential of the boundaries of shapes being used in the generation of designs.

Shape boundaries has been investigated in several studies such as Stouffs (1994), Earl (1997), etc. and so has the use of rules (Stiny, 1994), (Stiny, 2006), etc.

One of the results of applying rules to a shape is that the elements (and their boundaries) could behave in irregular manners, if they are not bound by systems to ensure that they work in desired ways.

During rule application, this would point to models of ambiguity and emergence which have been written about as different facets of the design disciplines. Both these features are present in the shape grammar formalism (Knight, 2003). Similarly, besides being perceptual aspects of design, one could associate the facets with aspects of shape recognition and rule-application in shape grammars.

Such behaviour would pose a dilemma for the designer. The question is not really whether this needs to be accepted as such, or whether it can be controlled when required. Literature in this subject area makes passing reference to this issue (Krstic, 2001).

2.1. THE BOUNDARIES OF SHAPES

Under normal design conditions, one usually assumes the various elements of the shape and their boundaries to be 'consistent', and that their behaviour is reasonably predictable. It is demonstrable that this is not necessarily so, especially in the shape grammar formalism.

In using design representations including shape grammars, one can move, rotate, mirror and/or scale the objects which can consist of the elements of points, lines, planes or volumes. In this the subshapes of the elements are 'associated' together, i.e., they are so to say, *tightly coupled* into subshapes that 'go together' in spatial relations.

2.2. THE USE OF RULES

A shape grammar is defined in the algebra U_{ij}. The use of rules is a generic method to create transformations in the design and which utilizes a shape change methodology. A rule $A \rightarrow B$ consists of subshapes A and B in the algebra. The rule applies to an initial shape C in the algebra if there is a transformation t such that $t(A) \leq C$ for application of the rule. The new shape C' would be produced according to the generalized formula

$$C' = [C - t_1(A)] + t_2(B) \tag{1}$$

where A, B, C and C' are shapes/subshapes (including parts of rules), and t_1 and t_2 are the transformations in the (sub)shapes.

In the application of rules, one sees that the algebra for U_{1j}-U_{3j} are a generalized process. The use of rules for shape transformations is such that substitutions of shapes are done from one side of the rule to the other.

2.2.1. 'Inconsistencies'

As mentioned, it is demonstrable that there could be unexpected behaviours (or *incorrect* as per a designer's perception) when applying rules that correspond to the differing algebras. The result of applying rules in a shape grammatical mode is that shapes and their associated boundaries in differing algebras would behave erratically if not bound by rules that will work in desired ways. This suggests the ambiguity and emergence inherent in shapes, and is oftentimes expressed as desirable.

3. The Interaction of Boundaries and Rules

One can explore the possibilities of the boundaries of shapes being used in the generation in designs.

Methods are proposed here to overcome such inadequacies. This exploration is whether the possibilities allow for behaviours as one wishes them to be. Some possibilities for bringing about the so-called consistency of 'normal' design involve some use of rules, etc. as mentioned above. Possible answers or solutions lie in:

- using labels /markers /symbols in V^*
- using rules in parallel, and
- using algorithms /rules that dynamically generate boundaries of shapes

These paths are explored to see systematic differences as well as the extent of the resultant shapes.

3.1. PROCESS TYPE I: USE OF LABELS / SYMBOLS

In this course of action, the use of labels, markers or symbols in V^* are associated with the elements such that the rules are made to apply when the markers are located consistent with the desired outcome. This is similar to the normal use of labels when using a shape belonging to a single dimension in U_{ij}, and seen in the usual literature on shape grammars.

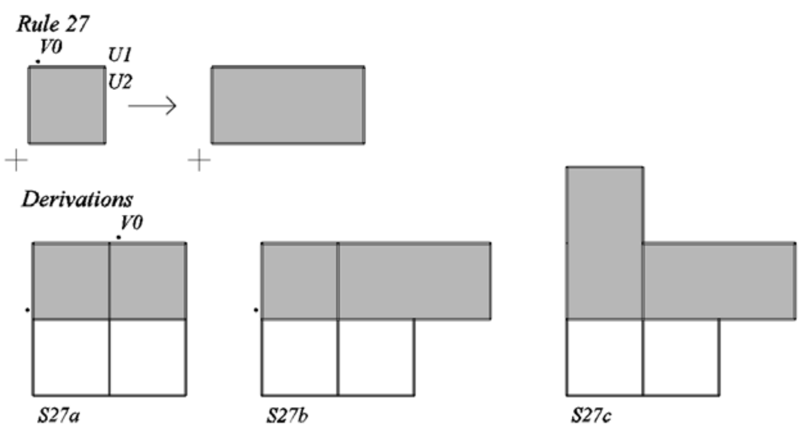


Figure 1. Using labels in V_0

In Figure 1, one sees that labels in V_0 identify the subshape where the rule is made to apply. Labels can be in any algebra of V^*, and identifies shapes and subshapes in any algebra of *U** also. The association is in the Cartesian product of the two algebras of shapes and labels *U*V**.

To demonstrate, in Figure 1, *Rule 27* identifies two instances where it is to be applied in the initial shape *S27a*, and the derivation is listed as *S27a* to *S27c*.

In this instance the rule has been applied to include the shapes in the algebras of U_1 and U_2, but a scrutiny will indicate that the differing derivations are dependent on the rules. This is particularly so because of subtractive processes, especially deriving from (1). Formula (1) indicates that subtraction is inherent to the shape formalism, i.e., that it is never free of such an operation. (Very few rules are probably purely additive, or are there any?). This assumption can be verified against the outcome -- that even if the shapes in the *LHS* and the *RHS* of the rule form are identical, the result of the process might still have a subtractive result.

3.2. PROCESS TYPE II: USING RULES IN PARALLEL

In this scheme, the rules are run in parallel for each algebra. Rules in parallel ensure that there is a consistency in applying them. Thus, rules for points, lines and planes are run in tandem for the differing algebras to ensure consistency (Some algebras might have 'empty' rules?).

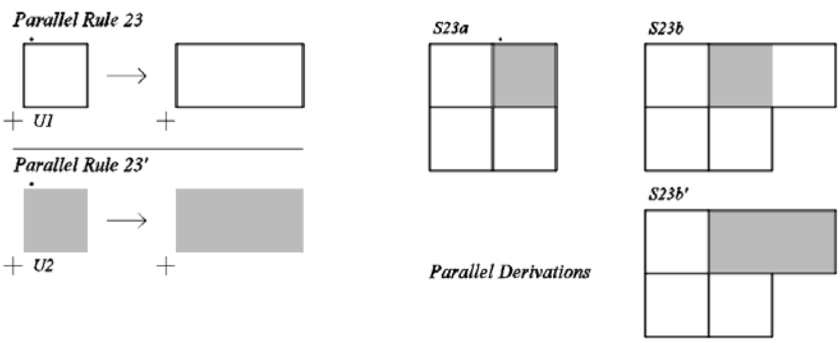
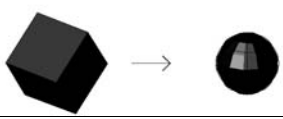


Figure 2. Rules run in parallel and their derivation.

Fig. 2 shows the derivation of shapes run in parallel. Here the resultant shape is the sum of $(S_{23b} + S_{23b'})$ which is $S_{23b'}$

TABLE 1. Generic shape-affecting rules.

Algebra U_{ij}		Rule LHS: Simple shapes	Rule RHS: (for example only) of sample shape	Shape	
U_{12}	S_1			S_4	[R2]
U_{22}	S_2			S_5	[R3]
U_{33}	S_3			S_6	[R4]

3.2.1. Shape Classification

During rule application, shape classification is a probable good method initiated prior to applying the rules. This would be akin to *screening* or a *trial run*. A shape classification could be generated based on the *complexity* of the shape ordered from 'simple' to 'complex'. Computationally, it seems probable that *simple* shapes would take less recognition time but more rule-application time (depending on the numbers), and *vice versa*. With a shape classification scheme, it could probably show that the use of *simple* elements in the left hand side (*LHS*) of rules will make shape change damaging to the desired design intent.

For example, the use of any of the *LHS* of the shape rules shown in Table. 1 would have a detrimental effect on shapes on which it is applied, sometimes beyond repair! Theoretically, if a lattice is made of the shape and its subshapes in the algebra, it could be shown that the *LHS*s of S_1-S_3 are the *simple* elements of the algebra in U_{ij}. For example, *simple* shapes S_1 in U_1 like a 'highest common factor' (*HCF*). So too would be shape S_2 similar in U_2, etc. Being lower in the lattice, it would be the *greatest lower bound* for the respective shapes in the algebras of U_{13} to U_{33}.

3.3. PROCESS TYPE III: BOUNDARIES AND THEIR GENERATION

In this procedure, listed as Type III, specific rules are applied to the higher order subshapes during which process their boundaries are generated.

The shape-rules are made to generate the boundaries, symbolically represented through rules as

$$X \rightarrow X + b(X) \tag{2}$$

or maybe just

$$X \rightarrow b(X) \tag{3}$$

In considering a generic algebraic formulation, the rules (2) or the simpler (3) can be elucidated as

$$X_i \rightarrow X_i + b_{i-1}(X_i) \tag{4}$$

and

$$X_i \rightarrow b_{i-1}(X_i) \tag{5}$$

It means that the given shape generates a bounded shape lower in the algebra.

Rule (2) is explained as [plane region in U_2] becomes [region in U_2 + a boundary element in U_1]. When applied to arbitrary shape S_{11a} in U_2, it generates the shape S_{11b}. Similarly, Rule (3) is explained as [region in U_2] becomes [a boundary element in U_1]

Fig. 3 below shows in shape grammatical terms the application of the rule for an arbitrary shape.

When applied to shape S_{12a} in U_2, the rule generates the shape S_{12b}, where the region is substituted by its boundary in U_1. This should, in general, work for shapes with all straight edges. (Shapes with arbitrary curved edges will need a separate curved shape substitution in the rule.) This probably would be better classified as a *composite algebra* as described in Knight (2003).

When using this rule, it is applied depending on the shape being worked on. This is a generic additive rule. It can be applied to any element of the algebra. In Figure 3, it is applied to $U_2 \times U_1$.

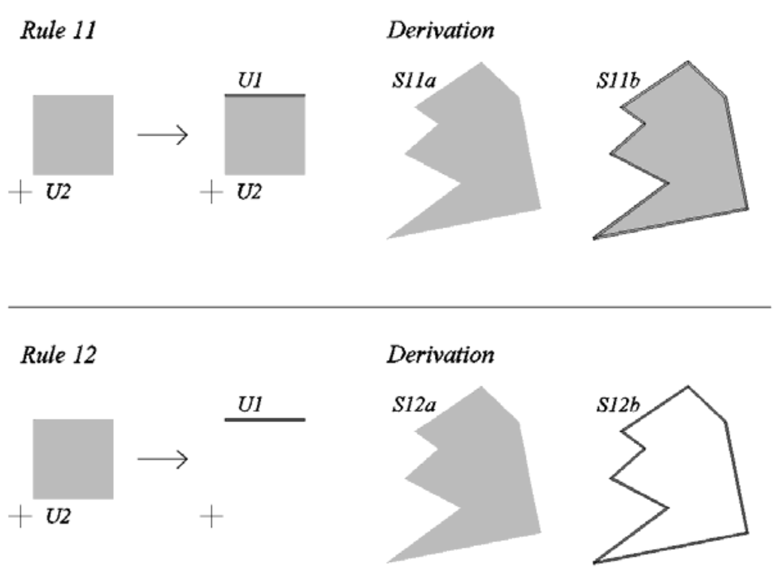


Figure 3. Rule-generated boundaries in U_2xU_1

In fact, the rule should be constructed so generic that it should be applicable to any shape type. Actually, it could also be achieved when combined with a subtractive rule $X \rightarrow \varnothing$. This is a generic representation of a rule which should be applicable for elements in U_0 to U_3.

Thus, a solid element *S* by (2) would generate the solid *S + b(X)* its bounding plane elements, or alternatively by (3), just the boundary elements *b(X)*. Thus, this algorithm/rule could generate the boundary elements for any shape *S* without having to go through difficult procedures. This algorithm could be used repetitively or recursively to generate all the bounding elements in the shape.

If it is applied to a solid or volumetric shape *S*, the derivation would be as indicated in Table 2.

TABLE 2. Derivation of Rule (2) for a Solid/Volume

No.	**Derivation of bounding elements**	**in the algebra of**
0	S	U_3
1	$S + b(S)$	U_3+U_2
2	$S + b(S) + b(b(S))$	$U_3+U_2+U_1$
3	$S + b(S) + b(b(S)) + b(b(b(S)))$	$U_3+U_2+U_1+U_0$
4	$S + b(S) + b(b(S)) + b(b(b(S))) + \varnothing$	$U_3+U_2+U_1+U_0+\varnothing$

The rule would sequentially generate all the bounding elements of the solid or volume *S*, for example. Rule application would not yield any useful result beyond the four listed.

Else by (3), it would sequentially generate the bounding elements in a substitution operation at each stage.

It is seen in Tables 2 & 3 that the application of the rule generates the elements until it is exhausted and cannot generate any more. The rules generate all the boundary elements of the shape in the respective algebras at each stage of rule application according to the formulae (2) and (3).

TABLE 3. Derivation of Rule (3) for a Solid/Volume

No.	**Derivation of the bounding elements**	**in the algebra of**
0	S	U_3
1	$b(S)$	U_2
2	$b(b(S))$	U_1
3	$b(b(b(S)))$	U_0
4	$\varnothing$	$\varnothing$

This could make the resulting shapes some more consistent to expectations. It is probably necessary that manipulation of higher order elements would be higher in the priority. Once the higher order elements have been transformed to the required design, one can use the locations in *3-space* to generate boundaries. These boundaries could become the elements for the next level of rules and rule application. As a simplistic comparison, one observes that in ‘normal’ design, one might do a preliminary massing/blocking study in *3-space* and then gradually fill in the details of the interiors, surfaces, etc.

Once the higher dimension elements have been changed to the designer's satisfaction, then the boundaries are generated, one notices that there could be more consistent design solutions.

3.4. AN EVALUATION

Three paths are explored to observe the systematic differences as well as the scope of the resultants.

It is felt that there are no sure answers to the question of consistency in the forming of boundaries. So too, it is unsure that these methods are the only possibilities. One merely explores the potential to observe in what manner the interactions can be mapped under normal design conditions.

Hybrids of all the above processes, Types I – III, are also part of the repertoire that can be used and built up by the designer to prevent inconsistencies in the development of the design, as an examination of Fig. 2 will indicate. This actually goes without saying, since the processes mentioned here are unaffected. They are merely being enumerated to demonstrate methods by which the designer could overcome the predicament.

4. Applications

The indicated processes triggers the possibility of application in the design of shape design interfaces, especially in *0D-3D* shape editors.

The use of labels, parallel rules and boundary generation are merely possible means of interfacing with familiar design processes. Design interfaces are one of the emerging areas in the human-computer interaction space. This probably points to possibilities for development. One can expect similarly that other promising avenues could be examined.

The generation of boundaries in form/shape production points to the fact that there would be heuristics and algorithms which could aid the novice designer if or when they use rule based systems or shape grammatical systems in particular.

5. Conclusion

The short exploration demonstrates that there could be snags in perceiving *garden variety* rule application as one might in the conventional sense of designing.

The use of *simplistic* rule applications could need a more detailed exploration, more so in the algebraic form when multiple elements are involved.

Although, what are demonstrated here are only possibilities, there is no assertion that the solution process is unique. There could be other methods of solving this predicament. There is no declaration that the processes listed will work every time a similar rule is applied.

Perhaps there is much more to learn about the design process even in the simple generation of form.

In architectural terms, we are looking at new ways to design which are still in the developmental stages. Theory-wise, these are quite advanced, but needs consistent applications. The process captures some of the deep-seated and fundamental practices that humans subconsciously use in design, and has been converted into a formalism that captures this for manipulation by hand or using reasonably advanced computational methods. These are the next generation technologies for the designer. This paper points out methods to overcome seeming shortcomings which from a designer's viewpoint might look as if as undesirable or 'inconsistent'.

References

Earl, C. F.: 1997, Shape boundaries, *Environment and Planning B: Planning and Design*, **24**(5), 669–687.

Knight, T.: 2003, Computing with ambiguity, *Environment and Planning B: Planning and Design*, **30**(2), 165–180.

Krstic D.: 2001, Algebras and grammars for shapes and their boundaries, *Environment and Planning B: Planning and Design* **28**(1), 151–162.

Stiny, G.: 1991, The algebras of design, *Research in Engineering Design*, **2**(3), 171–181.

Stiny, G.: 1994, Shape rules: closure, continuity, and emergence, *Environment and Planning B: Planning and Design*, **21**(7) Supplement, S49–S78.

Stiny, G.: 2006, Shape: talking about seeing and doing, MIT Press, Cambridge MA.

Stouffs, R.: 1994, The Algebra of Shapes, *PhD dissertation,* Department of Architecture, Carnegie Mellon University, Pittsburgh, PA.