Theoretical Refinements

by

Lionel March
University of California
Los Angeles, California 90095
lmarch@ucla.edu

These notes follow the six part division of Stiny, 1999 given in the Appendix.

1. Shape

Comparison of standard symbolic devises for shape computation with the shape grammatical approach. The use of units (atoms) and primitives in the former, and their absence in the latter. The importance of the embedding relation. The need for ambiguity.

2. Algebra

The table of *ij* algebras where $i \leq j$ is the degree of a spatial element and j is the dimension of the space: point $i = 0$, line $i = 1$, plane $i = 2$, solid $i = 3$. The question of boundaries (Earl 1997). The algebras in use (example, Krishnamurti, Earl 1998). Distinct *ij* algebras for shapes U, for labels V, and for weights W. Combining algebras (Krstic 1999). Relation of this approach to the spatial configurations of Grassman and the combinatorial simplices of Poincaré. Sets versus shapes. Equivalence relation versus embedding relation. Boolean rings and transformation groups. Shape grammars have been conceived from the beginning in static space, what about kinematic design- introducing t?

3. Rules and computations

Recognition of subshapes for rule application (Krishnamurti, Earl 1992). Representation using homogeneous coordinates and linear transformations. Plücker line coordinates. Real versus rational coordinates. Boolean versus Heyting algebras. Special rule systems (symmetry patterns, fractals) and generalized shape rules. Subshape recognition under similarity - determinate/indeterminate cases. Avoidance of special cases? The notion of self-referential shape description, that is, no extraneous reference frames. Are spatial transformations necessary?

4. Structure

Shapes have no intrinsic structure. Structure is an artifact of computation. Reading structure backwards (Stiny 1994). Topological closure, computational continuity, and emergence. Complexity.

5. Design
"A design is an element in an *n*-ary relation among drawings, other kinds of descriptions, and correlative devices as needed". What are the theoretical issues related to augmenting shape descriptions of designs with non-geometrical information. For example, putting the physics and the materials into shape. Shape computation in the design of geographical information systems (Chase 1999) - use of logical operators.

6.Ambiguity and process
The standing of the shape computation approach in philosophical discussions today. How do shapes compare with symbolic and numerical systems in particular?

References

Chase S C 1999 'Supporting emergence in geographic information systems 'Shape recognition in three dimensions' Environment and Planning B: Planning and Design 26 pp 33-44

Earl C F 1997 'Shape boundaries' Environment and Planning B: Planning and Design 24 pp 669-687

Krishnamurti R, Earl C F 1992 'Shape recognition in three dimensions' Environment and Planning B: Planning and Design 19 pp 585-603

Krishnamurti R, Earl C F 1998 'Densely packed rectangulations' Environment and Planning B: Planning and Design 25 pp 637-792

Krstic D 1999 'Constructing algebras of designs' Environment and Planning B: Planning and Design 26 pp 45-57

Stiny G 1994 'Shape rules: closure, continuity and emergence' Environment and Planning B: Planning and Design 21 pp s49-s78

Stiny G 1999 'Commentary: Shape' Environment and Planning B: Planning and Design 19 pp 7-14

Appendix

____

Commentary

____

Shape

I have been working on a book called *Shape* for a long time. My book is about shapes, and how they work and what they show when they're used to calculate. Every time I thought I knew enough about shapes and calculating to finish, I would find something new that needed thinking about or that was just fun to figure out. I still want to know a lot more about shapes and how to calculate with them. I will probably never understand everything I would like to. But I probably know enough now to tell a good story. It is a story told mostly with drawings punctuated with words. The hard part is to use words, so that shapes do not always look the same. The story is as multifaceted and changeable as shapes themselves.

This paper is a commentary on *Shape*. As you will see, I sometimes look at the shapes of things. But mostly, I am interested in shapes as things themselves. Shapes as things are numerous and easy to find. They come in many different kinds, yet their underlying properties are nearly all the same. Shapes contain basic elements. These include points, lines, planes, and solids. They combine in any dimension at least as big as their own. And they have like elements embedded in them, and divide and fuse freely.

Shapes combine basic elements. Some shapes combine basic elements on the page, as they do in my book. These shapes may be arrangements of zero dimensional points, or they may be made up of one dimensional lines or two dimensional planes. And shapes on the page may contain basic elements of several kinds. But not all shapes fit on the page. Shapes made up of solids are three dimensional and extend farther than any page or other surface goes. Nevertheless, solids combine to make most of the shapes found in everyday experience, the shapes of things that are easy to hold and bump into. And there are many more shapes that extend beyond the page: these include, for example, shapes made up of points, lines, and planes that are located in space.

My interest in shapes is both perceptual and computational. I am concerned with what I can see, and how I can use this to calculate. To many people today, this means complicated models of three dimensional solids, and the imposing technology of computer graphics and visualization. Experts in these fields tell me that any serious study of shapes must begin with solids. Shapes containing solids are plainly more useful than shapes made with basic elements of lesser rank. Even so, I have something else in mind. Computer models of three dimensional solids are apt to behave like arrangements of zero dimensional points! Shapes are prestructured in both cases. They have definite components -- ‘points’ -- that are independent in combination and simple beyond analysis. This is unavoidable when shapes are actually made up of points because embedding implies identity. But shapes of other kinds are not so limited. Computer models of solids ignore this crucial fact. Logic, I am told, demands it. Still, calculation is free to work in ways this kind of logic is normally loath to try. It is enough to start with common lines to see why: embedding no longer requires identity. This fixes the key properties of shapes whichever basic elements are combined, and makes calculating with shapes different.

The first chapter of my book does in fact begin with lines. But there are five other chapters, too, that elaborate the central insights presented there in many more ways. The chapters of my book are as follows:

1. Shape
2. Algebra
3. Rules and Computations
4. Structure
5. Design
6. Ambiguity and Process

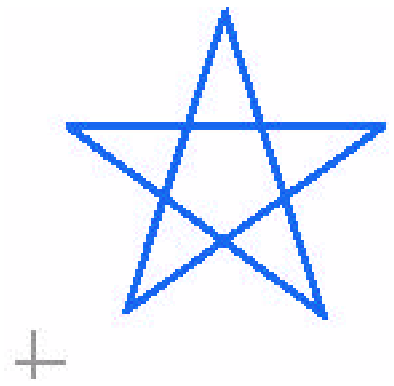
1. Shape

In this chapter line drawings are used to introduce shapes, and to show how shapes work in computations. Shapes are ambiguous. The parts they have depend entirely on what rules there are and how these rules are applied in computations. Parts may change freely any time rules are tried. The way I use rules now may conflict with the way parts were divided in the past, and has no claim on the future. This makes it difficult for computers to apply rules in computations. Shapes are moving targets. They are not defined in terms of components that are given in advance, at least as components are normally conceived as combinatorial units, fixed primitives, or permanent parts. Components are free to vary in any way whatsoever as computations unfold. They are never determined once and for all.

Computers calculate with shapes in terms of their representations. The way symbolic (combinatorial) devices are ordinarily used to represent shapes is illustrated in a simple example. A shape containing polygons -- a triangle and a separate chevron -- is reflected in a surprising way when a single rule is applied to translate triangles.

The example highlights the deficiencies of the symbolic approach with its complacent appeal to units and primitives and permanent parts before there is any kind of computation at all. But none of this is necessary. An alternative method of representation is proposed, and its details are carefully sketched. The new method keeps shapes ambiguous throughout every computation no matter how it is done. The computation may be performed automatically by computer, or with my active participation by hand and eye. The embedding relation and a handful of ancillary devices (maximal elements, reduction rules, and so on) provide the formal resources that make this possible.

It is important to note also that my example is generic. It is the leading term in two infinite series of computations in which rules are used to exploit ambiguity in shapes. Consider the computations in the more recent series. I haven't described these computations in print before. Each depends on an individual superstar like this one



to change a shape containing a triangle and multiple chevrons in an unlikely way. Rules are used to pick out triangles and chevrons, and to translate them. But the polygons the rules move to make superstars may not be the ones they find later. The rules divide superstars in many different ways. Exponentially many! The surfeit of possibilities shows just how uncertain calculating with shapes can be.

The chapter -- like all of the other chapters -- has a background section in which key ideas are compared with ideas in other areas. These include, for example, some in computer science in classics by Ivan Sutherland, Herbert Simon, Marvin Minsky, and John McCarthy, and some in the writings of the American pragmatists: thinkers like Pierce, James, and Dewey. Many people today are surprised to learn that standard ideas in computer science have a lot less to do with shapes and their computations than ideas found elsewhere.

2. Algebra

Shapes made up of lines show how shapes work, but do not exhaust the kinds of shapes there are. Shapes are defined in different algebras. These are listed in the following table

$U_{0\,0}$	$U_{0\,1}$	$U_{0\,2}$	$U_{0\,3}$
	$U_{1\,1}$	$U_{1\,2}$	$U_{1\,3}$
		$U_{2\,2}$	$U_{2\,3}$
			$U_{3\,3}$

An algebra $U_{i\,j}$ contains shapes. Every shape is made up of a finite number of basic elements of dimension $i \geq 0$ that are combined in dimension $j \geq i$. Thus, for example, shapes in the algebra $U_{1\,2}$ consist of lines arranged in the plane, and shapes in the algebra $U_{2\,3}$ are made up of planes arranged in space. In the algebra $U_{3\,3}$, shapes are arrangements of solids in space just like the shapes of physical objects in everyday experience.

The algebras in the table are ordered recursively via basic elements. The ordering highlights their underlying properties in terms of the embedding relation, finite content (nonzero length, area, and volume) for lines, planes, and solids, and definite boundaries for these elements that are shapes but not parts. In fact, the relationship is crucial. It makes computer implementations of shapes possible, ultimately with points that have neither content nor boundaries. And it lets computers calculate with shapes as if this were done by hand and eye.

The formal properties of the different algebras in the table are about the same, and they are described in detail. The algebras are all defined uniformly: relatively complemented, distributive lattices (Boolean rings) are augmented with operators (at least the group of Euclidean transformations). But more importantly, the algebras are compared in several different ways. The key distinction is drawn between algebras where $i = 0$, so that shapes just contain points and embedding implies identity, and algebras where $i > 0$. In the first case, there are atoms. Shapes behave as if they were symbolic objects like words formed from letters of the alphabet. In the second case, there are no atoms. Shapes have underlying spatial properties that explain why they are so ambiguous. The contrast is reinforced when the closure properties of the algebras are compared, and topologies for individual shapes -- Stone spaces in which parts are disconnected -- are defined for $i > 0$. This shows neatly why my approach to shapes is fundamentally different: in the point set approach normally followed in solid modeling and computer graphics, parts of shapes stick together. And it helps to understand why naive notions of computation on the one hand and creative expression in art and design on the other hand contrast so sharply.

The algebras U_{ij} can themselves be extended and combined in many ways. First, labels and weights can be introduced to define new algebras V_{ij} and W_{ij}. Labels in shapes classify basic elements, keeping ones in different categories separate when shapes are compared and combined, and do not interact themselves. Alternatively, weights are used to give basic elements additional properties, for example, physical or functional ones. Weights have their own underlying algebras, and interact as basic elements do. But this is not all. The possibility for other extensions of the algebras U_{ij} is always left open.

The algebras U_{ij}, V_{ij}, and W_{ij} can be combined in different operations, including, for example, appropriate sums and products (direct products), to obtain new algebras. These algebras typically contain compound shapes with one or more components in which basic elements of various kinds, labels, and weights are mixed and interact. The algebras U_{ij}, V_{ij}, and W_{ij}, and their combinations facilitate the definition of formal devices that correspond very closely to traditional media in art and design. This provides a computational foundation for these expressive activities with unlimited creative scope.

3. Rules and Computations

Shapes are meant for computation, and the algebras in which they are defined contain the means to calculate. These work together in shape rules. People love to hate rules, but the reasons they give for their disdain -- ‘rules are obtuse; ideal [practice] is flexible and based on the situation at hand’ -- miss the mark completely when shape rules are applied in computations. Shape rules provide the key to understand shapes; they restructure them freely and change them in the course of their use.

Shape rules are defined by ostension: any two shapes that are shown one after the other determine a rule. Suppose these shapes are A and B in a given algebra. Then, the rule they define is

$$A \rightarrow B$$

The details of using the rule and saying what it does are remarkably simple. The rule applies to any shape C in the algebra in a two stage process: (1) find a part of C that looks like A, and (2) replace this part with a new part that looks like B. The stages are always linked, so that the use of the rule completes a spatial analogy. The part like A is to the part like B just as A is to B. The part relation in the algebra and its Boolean and Euclidean operations are enough to tell when parts are alike, and to perform the twin actions of finding and replacing parts. The routine is the same regardless of the algebra involved, and exploits the resources of the algebra fully.

The use and classification of shape rules, and the coordinated actions of multiple rules in shape grammars are explored in depth. Simple rules for symmetrical patterns, fractals, and similar kinds of rudimentary geometrical figures are easy to define. More generally, shape rules are introduced to add and subtract shapes in accordance with spatial relations: the rules are inferred from sets of examples and defined entirely from scratch. But inference always has its hazards, and these are illustrated. Why inference is never conclusive is discussed, too. Then Lagrange's theorem is used to classify the possible actions of rules $A \rightarrow B$ in terms of the symmetries of the shapes A and B. The use of indeterminate rules is described. And rules that produce the shapes in different algebras are considered. This leads very naturally to the extension of rules in shape rule schemata. Schemata define families of rules under new transformations. The unlimited flexibility makes rules comfortable adjuncts to the unfettered actions of hand and eye. The technical part of the chapter ends with a brief discussion of some formal ways to modify shape grammars and to combine the languages they define, including, for example, the use of Boolean operations for sets and shapes.

Rules are conceived in different ways. Shape rules are compared with rules of these various kinds in the background section of this chapter. Some important historical precedents are traced in logic (plainly, Post's production systems, but also crucial insights from Pierce, James, and Dewey), and how rules are defined and used in adjacent fields are considered: in linguistics ('creativity', meaning and composition, Chomsky et al., the trouble with fixed categories, and the need for a dynamic taxonomy), computer science (picture languages, syntactic pattern recognition, expert systems, the physical symbol hypothesis, agents, cellular automata, artificial life, and so on), and elsewhere (the oft-heard assertion that rules are obtuse, for example, comes from a well known Chicago law professor and legal pragmatist). The direct contact shape rules have with concrete (unstructured) visual and spatial material in their underlying algebras gives them many special properties that are highlighted in this review. Their ability to handle ambiguity and the novelty it brings goes far beyond what rules are normally thought to do. It lets redescription and inference (deduction) -- perception and thought -- meld in a continuous process, and opens fresh possibilities for computation. I am unaware of any other computational device with this ability already built in.

4. Structure

Shape and structure (meaning) are tied, but neither permanently nor ineluctably. Shapes have no intrinsic structure; they come without divisions of any kind. Structure is an artifact of computation: it depends on the rules there are and the way these rules are used to pick out parts. Parts may vary freely across computations, or they may change dynamically within a single computation as it is performed. Alternative ways to describe these variations, and what they show are the subject of this chapter.

Continuity in computations and the topologies of shapes that make it possible open the main avenue of investigation. The use of shape rules in computations appears discontinuous when shapes are divided into parts in incompatible ways, when there are surprising gestalt switches between rule applications. A sense of continuity, however, can be restored retroactively both to structure the shapes in the computations and to explain exactly what happened. Some simple ideas from topology framed in terms of closure relations are readily used for this purpose. The ideas work for rules of any kind, even obviously useless identities of the form

$$A \rightarrow A$$

In fact, identities are enough to show a lot. Whenever an identity is applied to a shape, it leaves it alone. But the complete lack of constructive force is balanced by an unabated observational capacity. This illustrates the double action of rules -- recognizing parts and then replacing them -- and stresses the crucial distinction between shape and structure that is at the very heart of my kind of computation. The use of different identities to structure the shapes in computations is a good counterpart of seeing in alternative ways. It shows plainly that no observation is passive or inconsequential. This could only be if shapes were prestructured! Identities act with as much vigor as rules of any other kind. Rules are necessary for both division and change whatever the emphasis on shape or structure in computations.

The lack of intrinsic structure in shapes has some other implications that are elaborated at the end of this chapter. Take just one example. There is a lot of real excitement today about complexity, and how computers help to understand it in extreme cases. It is easy to agree that complexity is a correlate of structure -- how are parts divided for counting? -- and that it is impossible without it. But as a result, complexity is undefined before rules are applied in actual computations. It may vary widely as computations unfold, and may be irrelevant once they are completed. There are other ways to describe shapes and to talk about how they work.

5. Design

The approach to shape in my book comes from thinking about design, especially in fields where shape is important: in the visual arts, architecture, and engineering. Shape and design are separate subjects that stand alone, but each includes a useful window on the other. It is worth seeing what design looks like.

Design means different things to different people. But mostly, there is talk of process and product, without saying what designs really are. On the one hand, the emphasis is on methods and behavior that may lead to designs, while on the other hand, designs and the actual things that answer to them are confused. Designs, however, may be studied in their own right. They are used to describe things for making and to show how they work, and so link process and product. Shape grammars and the rules they contain provide the means to explore these relationships.

So, what is a design? My answer goes like this:

<u>A design is an element is an *n*-ary relation among drawings, other kinds of descriptions, and correlative devices as needed.</u>

This is just shape grammar talk for the familiar idea that designs and designing coordinate a host of descriptions to specify a lot of different activities and things that interact.

A design may be complicated and multifaceted. To begin with, drawings -- either singly or multiply -- are used to describe form. For example, they may establish three dimensional relationships in plans, sections, and elevations, or describe the separate parts in an assembly. Drawings, however, rarely give everything contained in a design. Sometimes, drawings are augmented with models and other kinds of geometrical representations. More generally, they are combined with labels, weights, and samples to describe form further and to provide details of function, use, material, construction, and so on. And drawings themselves may be described in many ways when they are divided into parts to clarify intention from different points of view, and for analysis, explication, and evaluation. These ends are also served when drawings are connected with other descriptions, including diagrams, graphs, and networks, and numbers and analytical expressions. The use and scope of the various descriptions in a design may depend on special instructions and documents, or be elaborated in additional commentary. Many devices of many kinds connect to make a design.

The definition of designs in computational processes is what algebras of shapes and shape grammars are all about. The algebras U_{ij}, V_{ij}, and W_{ij}, and others as well provide an ever expanding repertoire of expressive devices that make descriptions possible. And shape grammars defined in these algebras generate designs of things in certain styles and of certain types. Examples of this are presented in this chapter: introductory ones in which multiple descriptions are defined and connected in various ways, historical examples in which new work is produced in known styles or for established designers, examples of stylistic change, and examples of original designs made from scratch.

Of course, there is no guarantee that a design process will also be a computational process. The key is to show that computation is as flexible and as open-ended as design. A good sense of this flexibility and open-endedness is captured in Donald Schon's flattering account of the reflective practitioner. Schon puts 'reframing' and 'back talk' at the center of ideal practice in design and in other professions. This is the human ability to converse with your work: to reconfigure the situation at hand before and after you act, to respond rapidly and smoothly as your perceptions change, and to introduce new ideas in an ongoing process whenever you like. But I can do all of this and much more with shape rules! Shape rules meld redescription and inference, so that ambiguity and a confluence of shifting interests and goals can be handled together. Multiple perspectives provide the impetus for creative expression in new situations. These perspectives interact in many intricate ways in the flux of practice. The use of shape rules to perform computations in algebras of shapes formalizes this kind of process without loss.

6. Ambiguity and Process

I define shapes rigorously in my book. But the implications of this extend beyond the mere technicalities of computation. I can think about how a host of different things work in terms of unfolding processes in which my actions resolve ambiguity and change what I see. This is a lot like calculating with shapes. And the equivalence pays off handsomely. It opens some new ways to explore some serious questions. Aspects of this are easy to find in the preceding chapters: in Chapter 1 and Chapter 3, for example, when I compare what rules do to change shapes and symbols in computations; in Chapter 2, when I distinguish shapes with points and shapes without these basic elements; in Chapter 4, when I discuss complexity and why it depends on prior computation; and in Chapter 5, when I show how shape rules allow for the same kind of reframing and back talk that inform ideal practice. But there is plenty to say to finish the story. Calculating with shapes provides a view of the way things work that is more generous and that sweeps farther than most people are ready to believe. Nearly all of the people I've asked -- especially artists and designers -- find the idea that computation can be really informative unsettling. To them, it is presumptuous and irredeemably odd.

In this final chapter, I trace what a few thinkers say who have made it their job to know about the way things work, and show where this is and isn't the same for shapes. I start with the American pragmatists -- mainly James and Dewey -- and go on to Wittgenstein and contemporary writers with similar views. The problem with this sequence is that there is too much agreement about its importance. I don't want to rehearse what everyone knows, and I don't want to give the impression I am jumping on the latest bandwagon. Decorum says stop. Shapes stand firmly on their own! And they have for a long time. Yet there is no reason to shy away from novel connections if they give fresh insights. From my spatial point of view, these thinkers are plainly concerned with computation. They talk about the way things work as if they were talking about shapes and how rules apply to change them. This may be disorienting. Other writers begin where I do in this chapter only to argue that rules are always limiting. They say rules won't work otherwise. If they did, it wouldn't be calculating. But I want you to see rules in a new way. This may take a gestalt switch like ones for shapes with competing descriptions. Today, neopragmatists are eager 'to exploit the possibilities of massive redescription.' This is Richard Rorty's phrase, and the purpose of his 'ironist's' dialectic. The method is just the same when rules are applied to shapes, and in my account of this process. Calculation doesn't look the same.

A brief example gives a quick idea of what's involved. Hilary Putnam says a lot about 'conceptual relativity': there are competing descriptions of the same facts that are equivalent in one sense and incompatible in another. He illustrates this phenomenon with three individuals. (These may as well be labeled points. Unlike designers, professional philosophers seem happy with algebras where $i = 0$.) Putnam finds the phenomenon everywhere in today's science. It is 'pervasive' and 'strikingly non-classical', and something that philosophers should heed if they are to provide '. . . meaningful, important, and discussable images of the human situation in the world.' Putnam's problem is that contemporary logicians and meaning theorists pay no attention to it in their work. So much the worse for logic and meaning. But there is serious interest elsewhere, and there are useful results. The preceding five chapters are clear evidence of this. Conceptual relativity as Putnam describes it is an unavoidable fact in all computations with shapes, and a good account of it is necessary to say what happens when rules are applied. This responds directly to Putnam's concerns, and concedes nothing to logic. The norms of technical precision and rigor stay the same.

Of course, there are some important contrasts that are worth making. Consider one. Putnam appeals to 'conceptual systems' to decide what things there are. Conceptual systems determine the descriptions we can use without answering our questions. We still have to work to find out about things. Questions like 'How many things are there?' have answers that do not depend entirely on us. Answers within a conceptual system seem to be definite once they are given. But across conceptual systems, answers may vary. (This reminds me of the 'aesthetic systems' I explored in my book *Algorithmic Aesthetics*.) In a like way, shape rules are used to decide what parts shapes have. Yet these parts are always provisional. Their numbers may stray as computations proceed. I can find answers to questions about what parts there are now, and count them. My results, however, are never final. These are competing images of the way things work. I sometimes think that shape rules and the freedom I have to try new ones when I like make the whole idea of conceptual systems a little wobbly. But this is not at stake. Both images can be discussed, and that's all that matters. One of my aims in this concluding chapter is to show how calculating with shapes contributes to useful conversation.

George Stiny
Massachusetts Institute of Technology