Workshop Notes, Visual Representation, Reasoning and Interaction in Design, Fourth International Conference on Artificial Intelligence in Design, Stanford University, USA, 23 June 1996

Representing designs with logic formulations of spatial relations

Scott C. Chase

Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

scott.chase@nist.gov

Abstract

A new method of describing designs by combining the paradigms of shape algebras and predicate logic representations is presented. Representing shapes and spatial relations in first order predicate logic provides a natural, intuitive method of developing complete computer systems for reasoning about designs. Shape algebraic representations provide several advantages over more traditional geometric representations. The method described involves the definition of a large set of high level design relations from a small set of simple structures and spatial relations.

1 Introduction

The problem of predetermination

Research in the development of design modeling systems has identified the need for evolutionary models which support dynamic schema modification (Eastman et al., 1991). However, the development of current design systems does not easily support such a goal. They tend to be constructed in a bottom-up manner, with the design of low level data structures and operations first. This can be seen as a 'kit-of-parts' approach, and is often done in order to develop efficient operations for object manipulation. What this generally does is force the designer/user into a specific manner of representing and manipulating objects. Thus, the structure of a model must be decided at the start. Essentially this is akin to the philosophy of reductionism, which considers the universe to be composed of separate parts which, in various combinations, make up the whole:

> It is a natural human tendency to separate a whole into its parts, to categorize and classify, to draw boundaries between parts, and to define classes on the basis of rigidly defined boundaries. Boundaries so defined may be useful for some purposes, but they may badly confuse the accomplishment of other purposes. (Robinove, 1986 p. 15)

The decision to classify and structure up front may preclude the possibility of other desirable forms and structures in the future. It is extremely difficult, if not impossible, to anticipate all possible ways in which one might wish to view or classify parts of a model. This often requires an unmanageable amount of information. The problems with this approach were among the causes of the failure of early CAD building modeling systems in the 1970's and early '80s, which often required the predetermination of all types of information of interest, and for this information to be stored in a single model (Hoskins, 1973).

On the other hand, the philosophy of holism considers the universe to be a whole rather than the sum of its parts. A system which forces no preconceived structure upon the user, but rather, allows one to find all sorts of emergent features and properties from within the whole, would be extremely desirable. This might enable an easier development path in a top-down fashion, from the abstract to the specific.

Shape algebras and shape grammars in design

The algebras of shape (Stiny, 1991) can support both holistic and reductionist views. By considering shapes as finite sets of elements which can carry fixed properties, a reductionist view is supported. The real power of such algebras, however, lies in the fact that the elements of a shape and their properties may be defined in such a manner as to enable the emergence of features which are not apparent in the initial formulation of a shape. In addition, the generality of their representations, their reliance upon a minimum of structure, and their use in combination can provide the semantic richness needed for design generation and analysis. The practicality of these algebras has been demonstrated with their use in shape grammars (Stiny, Gips, 1972), production systems which generate languages of designs.

The shape grammar formalism has been used in the construction of a wide variety of design grammars, generating such designs as Wright's Prairie houses (Koning, Eizenberg, 1981) and Hepplewhite chair back patterns (Knight, 1980). While remaining true to the formal representations, these tend to be paper and pencil exercises, as the computational issues in computer implementation are great. Due to these limitations, computer implementations of shape grammars (and indeed, design systems in general) tend to simplify the formal representations in order to solve the computational problems. Attempts at implementing a shape grammar system supporting emergence tend to have other restrictions limiting their use in practical applications (Chase, 1989; Krishnamurti, 1980; 1981; Krishnamurti, Giraud, 1986; Tapia, 1996).

Logic as a specification tool

Previously, most of the shape grammar literature has dealt with the development of grammars, focusing on specific languages of designs. The representations used tend to describe shapes and spatial relations simply by drawing them, thus limiting much of the description to non-parametric shapes. With few exceptions, discussion of parametric shapes and grammars has been limited to natural language descriptions of the conditions placed upon a shape and very general descriptions of rule application.

Representing shapes and spatial relations in first order predicate logic provides an easy way to develop complete computer systems for reasoning about designs. The use of logic provides a natural, intuitive method of generating precise definitions of parametric shapes and high level spatial relations. Its use as a specification and programming tool has become widespread over the past two decades, providing advantages over traditional procedural programming methods, among those the ability to specify the knowledge to be encapsulated in a model (description) without the need to specify data manipulation procedures (prescription) (Kowalski, 1979). The use of logic can facilitate a top-down method of development, from the abstract to the specific. The symbolic abstractions of logic formulations enable one to denote entire classes of data structures and procedures while ignoring their details. This can be a more natural method of development than having to deal with often nonintuitive formulations.

The use of logic in design is not new; general surveys of how logic may be used in design include (Coyne et al., 1990) and (Mitchell, 1990). Some examples include its use in shape grammar and reasoning systems (Chase, 1989; Damski, Gero, 1996; Heisserman, Woodbury, 1994; Krishnamurti, 1992a; 1992b; Krishnamurti, Giraud, 1986). The weaknesses of the shape grammar implementations are in their limitations dues to computational problems; those of other logic based systems are in their representations of design objects, which—with few exceptions—cannot support emergent features.

Rather than attempt to solve all of these problems, we focus here on the representations rather than the search and control issues inherent in any production system. The approach taken is of modeling designs using spatial relations based upon shape algebraic representations. This entails the construction of a formal, hierarchical model of shape, spatial relations and non-spatial properties from first principles of geometry, topology and logic. The shape algebra formalism is extended by using logic to make more precise, generalized, parametric definitions of shape and spatial relations than has been previously possible. The value of such a model and the advantages of the representations used over more traditional 'kit-of-parts' models can be demonstrated by the use of these generalized spatial relations for solving typical problems involving spatial reasoning.

2 Shape algebras, emergence and spatial relations

Shapes and emergence

Shapes composed of sets of elements such as points, lines, regions and solids can be manipulated in algebras, as described by Stiny (1991). These are constructed by *a*) describing an element by its boundary (e.g., points bound lines, lines, bound regions, etc.) and an additional descriptor (often an analytic equation); and *b*) defining the operations +, –, • (product) and the part-of relation ≤ on pairs of elements and shapes (Figure 1) [1]. Thus, an element is defined by its boundary (a set of maximal elements, i.e. points, lines, etc. which can't further combine with each other) and descriptor:

$$e = \langle descriptor, \{e_B \mid e_B \text{ is a boundary element of } e\}\rangle$$

and a shape S is defined by its set of maximal elements:

$$S = \{e \mid e \text{ is a maximal element of } S\}$$

It should be noted that (with the possible exception of brep solid models) these representations are very different from the traditional shape representations used in computer graphics systems. There, shapes are often represented as sets of atomic elements which cannot be further decomposed, e.g., no parts of lines can be easily recognized. With the representations described in this paper, the part-of relation enables the recognition of any element which is embedded in (part of) another element, thus allowing all sorts of emergent properties of a shape, i.e. subshapes, to be found (Figure 2). Thus, in Figure 2, the emergent subshapes of S shown (and any others which are not *subsets* of the lines of S) cannot be found in a non-decomposable representation of the lines of S. That representation would require modification in order to find these subshapes. This requires the prediction of what possible subshapes of S might be desired *before* deciding upon its representation. It is this combination of non-predetermination of structure with minimal representation producing maximum expression which makes the maximal element representation so appealing.

---

[1] It should be noted that the definition of an algebra of solid elements in 3-space is similar to the boundary representation and Boolean operations for solid models commonly used today. However, the definitions of shapes and relations described here apply, in general, to elements of *any* dimension.

The shape algebra formalism has been extended to include algebras of shape with non-geometric properties such as color, texture, material properties, and function and cost (Stiny, 1992). The author's research has included development of some examples using such algebras (Chase, 1996), but they are beyond the scope of this paper.

Fig. 1 Shapes *A*, *B* and *C* composed of lines and results of operations upon them.[2] a) *A, B* and *C* are represented as sets of maximal lines; b) *A, B* and *C* are represented as sets of non-decomposable line segments.

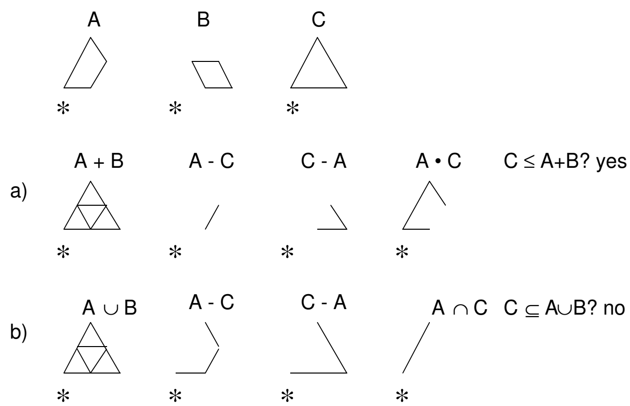
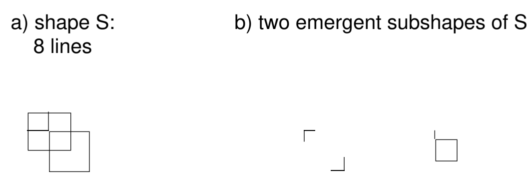


Fig. 2 *a)* shape *S*, consisting of 8 lines. *b)* Two possible emergent subshapes of *S* which cannot be found in a representation of *S* consisting of 8 *non-decomposable* lines.



Spatial relations

Spatial relations useful in design can be developed using the basic constructs of shape definition and shape operations. The non-graphic symbolism of logic is a powerful tool in extending the formal definitions of shape algebras and spatial relations.

It is assumed that an underlying data structure is implementable for the geometric description of shape. The descriptions here of shape and spatial relations deal mainly with the topological properties of shape. We recognize that the problems of low-level geometric computation have been researched by others and that solutions exist which are adequate to support investigation of the issues here. The amount of research in computational geometry and computer graphics over the past thirty years tends to support this argument.

The basic definitions of shape (by boundaries and descriptors) and shape operations described above are used to construct definitions for spatial relations which apply to

[2]The symbol * shown here represents a position reference marker for use in comparing two shapes. It and any accompanying text are not considered part of the shape itself.

multiple element types. In this way, spatial relations are parameterized and can apply to shapes in any dimension. For example, a single definition of the relation *share_boundary* can be constructed for both lines and regions (and indeed, elements of higher dimension) by examining the product (•) of their boundaries (Figure 3):

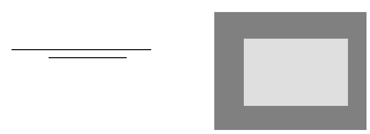
$$\forall A \forall B\ [\mathit{share\_boundary}(A,B) \leftrightarrow \mathit{boundary}(A) \bullet \mathit{boundary}(B) \neq \emptyset]$$

This relation can then be used in the definition of other relations, e.g. *surrounded_by* (Figure 4), in which an element A is surrounded by an element B if A is a part of B and they don't share boundaries:

$$\forall A \forall B\ [\mathit{surrounded\_by}(A,B) \leftrightarrow A \leq B\ \&\ \neg \mathit{share\_boundary}(A,B)]$$

Fig. 3 *share_boundary* relation. The lines share a boundary (endpoint); the regions share a portion of their boundary lines.

Fig. 4 *surrounded_by* relation. The bottom line is surrounded by the top one (assuming collinearity); the smaller region is surrounded by the larger one.



In the course of this research, a large set of spatial relations and operations has been constructed from a base set of primitive definitions. These include standard geometric concepts such as *parallel*, *distance*, *projection*, as well as interesting extensions to a basic *intersection* operation. These general relations and operations can be used to fashion domain specific relations and queries.

3 Applications

Here we offer a few examples of how this shape algebra/formal logic method can be used in reasoning systems for specific problems in design and other spatially oriented domains. The examples shown in GIS; following is a brief description of other work in an architectural domain.

Geographic information systems

Geographic information systems (GIS) provide a good testing ground for the spatial relations developed. Much of GIS deals with two dimensional maps containing simple relations between points, lines and regions. There is a large body of GIS research dealing with topology and spatial relations. However, planners often tend to focus on specific features of interest, and design data structures to represent this closed set of features. Thus, the possibility of emergent features and properties is limited or nonexistent.

As an example of emergent features, the *continuous* relation [3] (Figure 5), in which a shape is continuous if all of its points are 'connected' (i.e. in the transitive closure of a 'neighborhood' relation), can be useful for determining connectivity and accessibility between points or regions. Given a shape consisting of roads (lines) and properties (regions), one can see that there is access between discrete properties *A* and *B* if there is a continuous shape which contains lines which are within *A* and *B* (Figure 6a). The following formula illustrates the necessary logical conditions (slightly simplified by the omission of some implied conditions and universal quantifiers):

$$\begin{aligned}&accessible(A,B) \leftrightarrow A \bullet B = \emptyset \ \& \ \exists C, C_A, C_B \ (continuous(C) \ \& \ l \in C \rightarrow \\ &\quad C_A \leq C \ \& \ C_B \leq C \ \& \ within(C_A, A) \ \& \ within(C_B, B))\end{aligned}$$

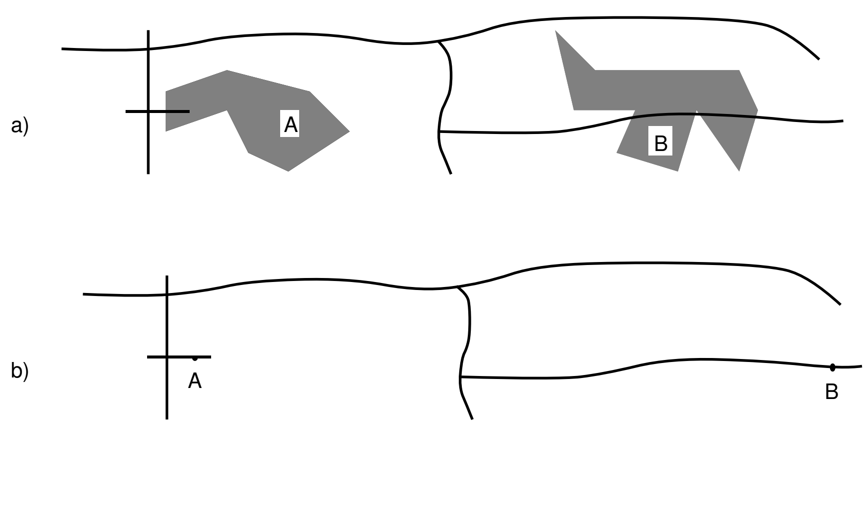
A similar situation exists to determine whether two points are accessible by road (Figure 6b). Points *A* and *B* must be within a continuous shape representing a road system:

$$\begin{aligned}&accessible(A,B) \leftrightarrow \exists C \ (continuous(C) \ \& \ l \in C \rightarrow \\ &\quad within(A,C) \ \& \ within(B,C))\end{aligned}$$

In other systems, accessibility problems are generally handled by connectivity graphs which are explicitly constructed for this purpose. Here, the graph emerges from the *implicit* connectivity relations among the various elements of the shape.

Fig. 5 *continuous* relation. *A* is continuous because all of its points are connected; *B* is not as there are two distinct classes of connected elements.

Fig. 6 Accessibility. a) between regions; b) between points. *A* is accessible to *B* because there is a continuous shape containing *A*, *B* and a set of (road) lines.



[3] A formal definition for *continuous* can be found in (Chase, 1996).

Other work by the author in the area of GIS (Chase, 1996) has involved a comparison with a typical relational database implementation of a geographic information system. This study showed that in a typical GIS relational database system, all information (i.e. features) of interest (roads, rivers, intersections, bridges, and districts) had to be explicitly entered into the relational tables for points, line segments and polygons. Queries using the relational algebra tend to be nonintuitive and difficult to formulate (Figure 7a). While the actual computations using the shape algebraic formulations described here may be similar to those of the relational representation, they tend to be hidden from the user. The queries prove to be relatively easy to formulate (7b), and allow for the possibility of emergent features, impossible in the more traditional relational representation.

Fig. 7 Comparison of queries on a GIS database. *a*) Relational algebraic query; *b*) the same query using relations based on shape algebras.

Name the intersections that are entirely within districts (not on district boundaries).

a) Relational algebra:

$A = \pi_{point\_label}(\sigma_{feature\_type=\text{‘intersection’}}(point))$
$B = \sigma_{left\_polygon \neq right\_polygon}(line)$
$C = \pi_{beginning\_pt}(B) \cup \pi_{ending\_pt}(B)$
$D = A - C$
$E = \pi_{D.point\_label, line.left\_polygon}(D \bowtie_{D.point\_label=beginning\_pt}\ line) \cup$
$\quad \pi_{D.point\_label, line.right\_polygon}(D \bowtie_{D.point\_label=ending\_pt}\ line)$
$F = \pi_{E.point\_label, area.feature\_name}(E \bowtie_{E.polygon=area.polygon\_label}\ area)$

Answer =
$\quad \pi_{point.feature\_name, F.area.feature\_name}(F \bowtie_{F.point\_label=area.point\_label}\ point)$

b) Shape algebra:

$$\text{Ans} = \{ \langle District, Int \rangle \mid district_{22}(District)\ \&\ Int = \{ I \mid intersection_{02}(I)\ \&\ within(I, District)\ \&\ \neg within(I, boundary(District)) \} \}$$

Architectural design

Spatial relations have also been constructed which can be used for various tasks involving architecture plan representations. These include identification and generation of wall centerlines, identification of interesting emergent subspaces, and recognition of views between spaces (Chase, 1996).

4 Implementation issues

The method for modeling designs introduced here is descriptive, rather than operational in manner. In this way, we have deliberately not specified data structures or algorithms to manipulate the data, but focused instead on the logic of the relations between objects. This permits the later modification of a data structure without altering higher level procedures.

The next phase of this research will involve the development of a computer implementation. Naturally, issues of data structure and computational complexity must be considered. This may involve compromises in the areas of model soundness and completeness by restricting the types of queries and data objects permitted. In addition, the generality of some relations may be sacrificed for algorithm efficiency. Despite these potential problems, it is the author's belief that developing a model using abstract data structures has great potential.

A deductive database (Gallaire, Minker, 1978) is under consideration for a prototype implementation of the model described here. Deductive databases combine aspects of logic based systems as well as database systems. A cursory examination of the relations developed here and the anticipated query types indicates that the limitations of deductive databases may be acceptable with only minor modifications to the current state of the model.

5 Conclusions

The combination of shape algebras and symbolic logic can be a powerful tool in the specification of design systems. A model of shape and spatial relations based on first principles of geometry, topology and logic improves upon previous efforts in several ways:

- Shape algebra representations are superior to those of more traditional 'kit-of-parts' systems in that they require minimal predetermination of structure and support direct manipulation of emergent features.

- The model provides a method for defining parameterized spatial relations on shapes of any dimension and description. This improves upon previous, less parameterized representations of shape algebras by the use of logic in a precise manner, rather than by drawing or natural language description of parameters.

- Use of a logic formulation allows one to focus on high level knowledge, not on low level data structures and implementations.

- Logic formulations are amenable to computer implementation: logic programs and deductive databases are examples of programming paradigms which support subsets of first order logic formulations.

By concentrating on the knowledge to be modeled and not directly on implementation, more powerful models of design can be developed. The potential for implementing these models with little sacrifice in functionality appears to be great. It is hoped that future research will adopt this approach and overcome the traditional bias of favoring implementation at the cost of representation.

References

Chase S C, 1989, "Shapes and Shape Grammars: From Mathematical Model to Computer Implementation", *Environment and Planning B: Planning and Design* **16:2** 215-242

Chase S C, 1996, *Modeling Designs With Shape Algebras and Formal Logic* Ph.D dissertation, University of California, Los Angeles

Coyne R D, Rosenman M A, Radford A D, Balachandran M, Gero J S, 1990 *Knowledge-Based Design Systems* (Addison-Wesley, Reading, Mass.)

Damski J C, Gero J S, 1996, “A logic-based framework for shape representation”, *Computer-Aided Design* **28:3** 169-181

Eastman C M, Bond A H, Chase S C, 1991, “A Formal Approach For Product Model Information”, *Research in Engineering Design* **2** 65-80

Gallaire H, Minker J, 1978 *Logic and Databases* (Plenum Press, New York)

Heisserman J, Woodbury R, 1994, “Geometric design with boundary solid grammars” in *Formal Design Methods for CAD* Ed J S Gero, E Tyugu (North-Holland, Amsterdam) 85-105

Hoskins E M, 1973, “Computer Aids in System Building” in *Computer-aided Design* Ed J Vlietstra, R F Wielinga (North-Holland, Amsterdam) 127-140

Knight T W, 1980, “The generation of Hepplewhite-style chair-back designs”, *Environment and Planning B* **7** 227-238

Koning H, Eizenberg J, 1981, “The language of the prairie: Frank Lloyd Wright’s prairie houses”, *Environment and Planning B* **8** 295-323

Kowalski R A, 1979 *Logic for Problem Solving* (North-Holland, New York)

Krishnamurti R, 1980, “The arithmetic of shapes”, *Environment and Planning B* **7** 463-484

Krishnamurti R, 1981, “The construction of shapes”, *Environment and Planning B* **8** 5-40

Krishnamurti R, 1992a, “The arithmetic of maximal planes”, *Environment and Planning B: Planning and Design* **19** 431-464

Krishnamurti R, 1992b, “The maximal representation of a shape”, *Environment and Planning B: Planning and Design* **19** 267-288

Krishnamurti R, Giraud C, 1986, “Towards a shape editor: the implementation of a shape generation system”, *Environment and Planning B: Planning and Design* **13** 391-404

Mitchell W J, 1990 *The Logic of Architecture* (MIT Press, Cambridge, Mass.)

Robinove C J, 1986, “Principles of Logic and the Use of Digital Geographic Information Systems”, Circular 977, Dept. of the Interior, U.S. Geological Survey

Stiny G, 1991, “The Algebras of Design”, *Research in Engineering Design* **2** 171-181

Stiny G, 1992, “Weights”, *Environment and Planning B: Planning and Design* **19** 413-430

Stiny G, Gips J, 1972, “Shape grammars and the generative specification of painting and sculpture” in *Information Processing 71* Ed C V Freiman (North-Holland, Amsterdam) 1460-1465

Tapia M A, 1996, *From Shape to Style, Shape Grammars: Issues in Representation and Computation, Presentation and Selection* Ph.D dissertation, University of Toronto