ACADIA '96 Proceedings, Tucson, AZ, 31 October–3 November, 1996

Design Modeling With Shape Algebras and Formal Logic

Scott C. Chase
National Institute of Standards and Technology
Manufacturing Engineering Laboratory

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 logic provides a natural, intuitive method of developing complete computer systems for reasoning about designs. The advantages of shape algebra formalisms over more traditional representations of geometric objects are discussed. The method employed involves the definition of a large set of high level design relations from a small set of simple structures and spatial relations. Examples in architecture and geographic information systems are illustrated.

1 BACKGROUND

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)

> As soon as you perceive an object, you draw a line between it and the rest of the world; you divide the world, artificially, into parts, and you thereby miss the Way. (Hofstadter, 1979 p. 251, in discussing Zen's struggle against dualism)

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 (Eastman, 1978; 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, more flexible design development path in a top-down fashion, from the abstract to the specific.

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. How these representations support emergence is demonstrated in Section 2.

Shape Grammars in Design

Shape grammars have over the past two decades been shown to be a powerful means of generating and analyzing languages of designs. A wide variety of grammars have been constructed which encapsulate styles of designs in areas such as fine arts (Kirsch, Kirsch, 1986), architecture (Flemming, 1987; Heisserman, Woodbury, 1994) and landscape design (Stiny, Mitchell, 1980). Recent work includes the development of grammars which generate *new* languages of designs (Knight, 1992; Knight, 1995).

While remaining true to the formal representations, these tend to be paper and pencil exercises. 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. The dearth of computer implementations can be seen as due to problems of computational complexity. Because of this, computer implementations of shape grammars (and indeed, design systems in general) have tended to have many restrictions and simplifications of the formal representations in order to solve the computational problems, thus limiting their use in practical applications (Chase, 1989; Krishnamurti, 1981; Krishnamurti, Giraud, 1986; Tapia, 1996).

Logic as a Specification Tool

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, initially with the Prolog language and continuing with constraint logic programming languages (Benhamou, Colmerauer, 1993). These provide 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. This is possible because 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 unintuitive 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, 1992; 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.

Approach

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. These relations can be used to describe designs in more ways than simply geometrical composition: they have the potential to represent behavioral, psychological and cultural issues. 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.

The remainder of the paper is organized as follows. Section 2 offers an introduction to the shape algebraic representations used and illustrates how they support feature emergence. Section 3 shows how shapes and spatial relations can be formally defined in logic. Section 4 offers examples from the domains of architectural plans and geographic information systems. Section 5 discusses the problems inherent in a computer implementation of such a model, and Section 6 offers some conclusions.

2 SHAPE ALGEBRAS AND EMERGENCE

With a small set of compositional rules, shape grammars can generate a large, rich variety of designs and can support shape emergence, allowing new shapes to be found in existing compositions. The properties of the algebras of shape are what provide this richness.

Maximal Line Algebras

Shapes are finite arrangements of basic geometric elements. The types of basic elements we consider here are points, lines, two dimensional regions, and three dimensional solids. Each has its own algebra in which it can be manipulated. We describe here algebras of lines and shapes composed of lines.

A line has finite, nonzero length. It may be decomposed into its parts, which consist of other lines embedded within it. There are an infinite number of lines which may be part of a given line. We define the *part-of* relation ≤ to describe this situation, e.g., $l_1 \leq l_2$ means that line l_1 is embedded in or part of line l_2.

Two collinear lines which overlap in any part or who have an endpoint in common may combine via reduction rules for + (sum), – (difference) and • (product) to form new lines (Chase, 1989). A shape consists of a set of lines which are *maximal*, i.e. no line in the shape contains any parts which are part of any other line in the shape. Consequently, no two lines in the shape can be combined to form a single line. As shapes are simply finite sets of maximal lines, the relation ≤ and the operations +, – and • may also be defined for shapes by repeated application of the reduction rules for lines (Figure 1a). The relation ≤ between shapes is called the *subshape* relation.

It should be noted that this representation is very different from the shape representations typically used in vector based computer graphics systems (with the possible exception of brep solid models). There, shapes are often represented as sets of lines which cannot be further decomposed, i.e. no parts of lines can be easily recognized (Figure 1b). This difference is significant, as use of the maximal line representation allows the recognition of emergent subshapes. Here we define an emergent subshape as a subshape containing at least one maximal line which is not maximal in the original shape. Figures 2 and 3 illustrate this difference. Considering a maximal line representation, there are 22 instances of Euclidean transformations (compositions of translation, scaling, rotation or mirroring) of shape A which are subshapes of S in Figure 2. If, instead, we consider A and S as sets of *non-decomposable* line segments corresponding to those in the maximal line representation, there are *no* subsets of transformations τ of A in S. In order to identify the same subshapes, A and S have to be represented by different sets of non-decomposable line segments A' and S', using a larger number of line segments (Figure 3). If we wish to find some other subshape of S', we can do so using the maximal line representation, but may need to modify the non-decomposable line segment representation of S'. For example, the shape B in Figure 3, represented with maximal lines, is a subshape of S . If, however, it is represented with non-decomposable lines, it is not a subshape of S'. Thus, with a non-decomposable representation, it is necessary to predict what possible subshapes might be desired *before* deciding upon a shape’s representation.

It is this combination of non-predetermination of structure with minimal representation producing maximum expression which makes the maximal line representation so appealing. In effect, one does not have to worry about schema modification, as an infinite number of features can be found at any time during the design process. With traditional representations, the potential for schema modification is a major concern.

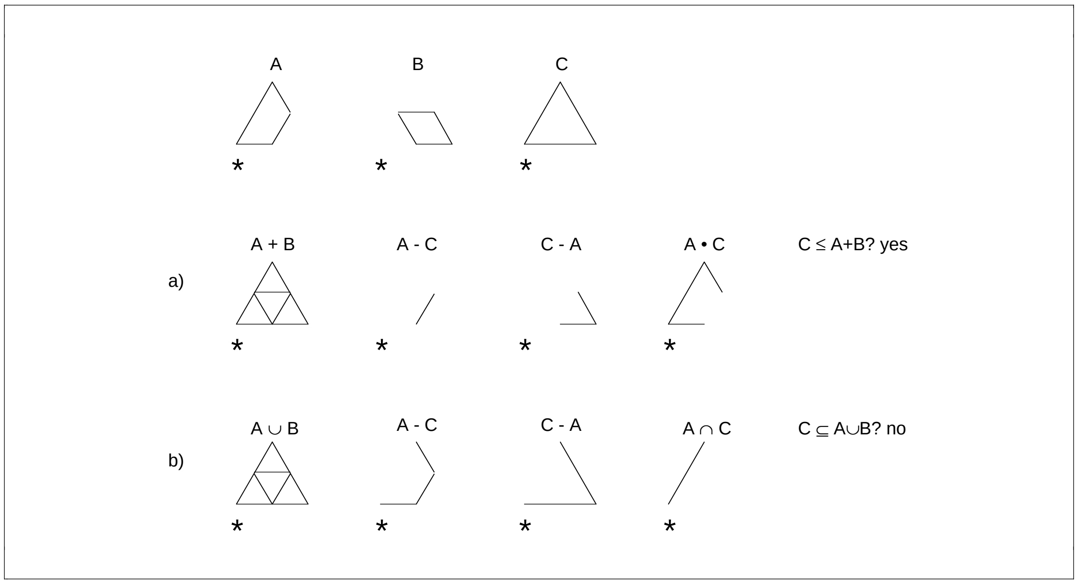

Figure 1. Shapes *A*, *B* and *C* composed of lines and results of operations upon them. a) *A*, *B* and *C* are represented as sets of maximal lines; b) *A*, *B* and *C* are represented as sets of non-decomposable line segments. Note: the symbol * shown here and in other figures represents a position reference marker for use in comparing two shapes. It and any accompanying text are not considered part of the shape itself.

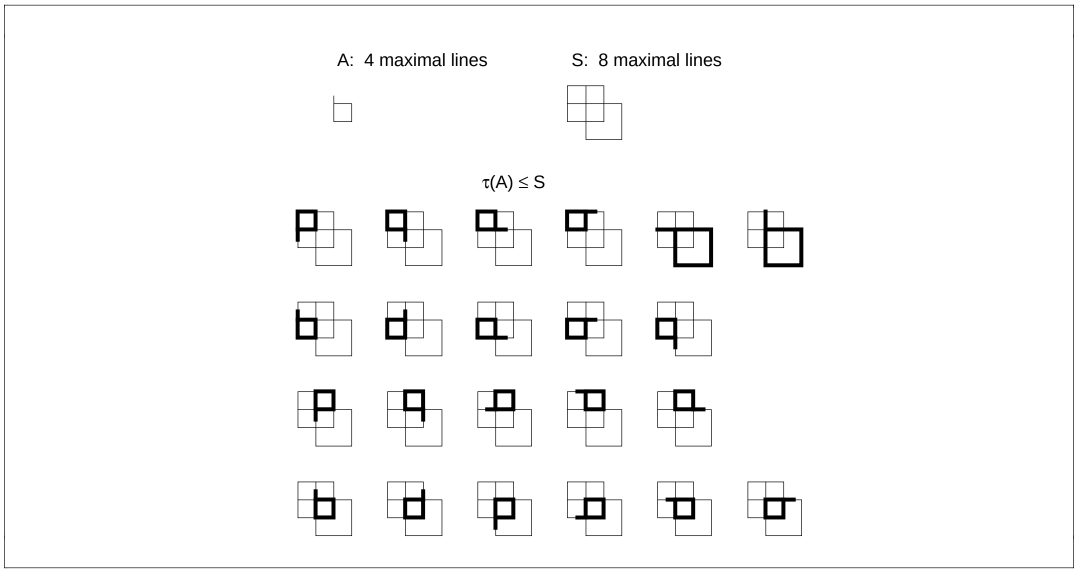

Figure 2. The 22 possible subshapes of *S* that can be produced by transformations of *A*.

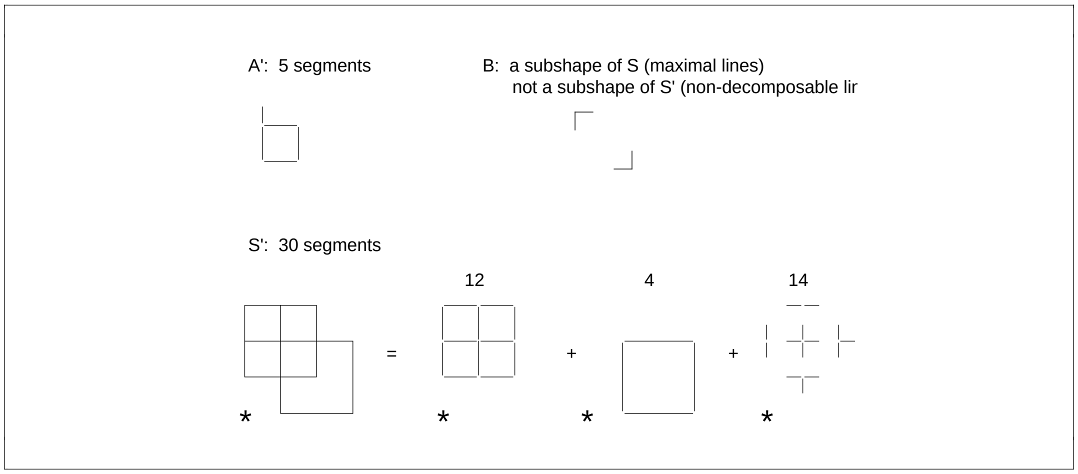
A': 5 segments
B: a subshape of S (maximal lines)
not a subshape of S' (non-decomposable lir
S': 30 segments
12
4
14
=
+
+
*
*
*
*

Figure 3. Line segments needed for a possible representation of the shapes in Figure 2 using non-decomposable lines.

Other Algebras

Algebras which support feature emergence in a manner similar to the algebras of maximal lines can be defined for other geometric objects, namely, points, planar regions (Figure 4), and solids (Stiny, 1991). In addition, algebras have been defined to support non-geometric attributes of shape, which include material properties, function, cost, etc. (Stiny, 1992).

In design, there is a strong need for the simultaneous use of multiple representations (Chase, 1993). To that end, these algebras can combine to produce new algebras of shape containing multiple types of elements, and grammars written to manipulate shapes in these compound algebras (Stiny, 1992).

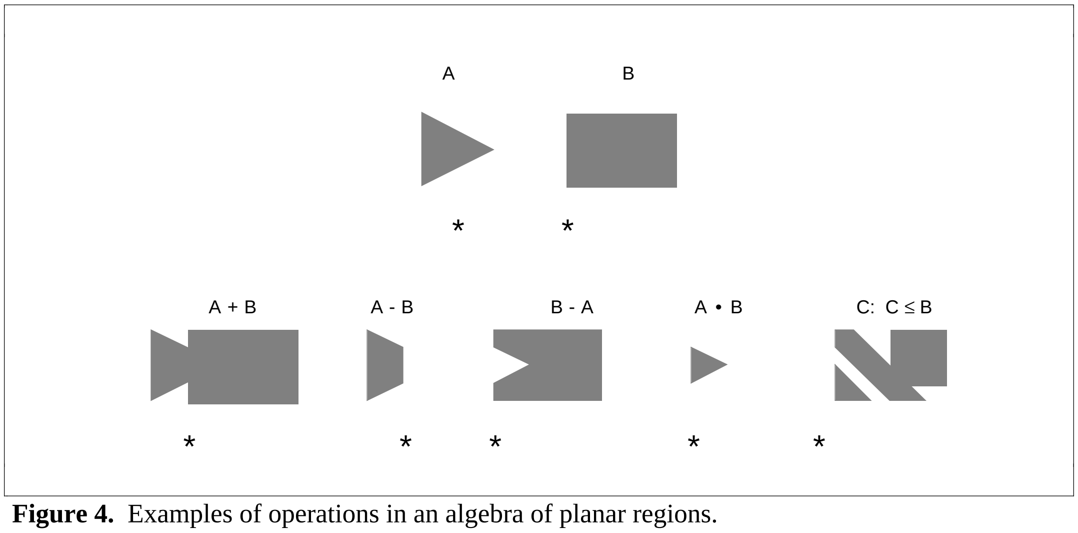

Figure 4. Examples of operations in an algebra of planar regions.

3 FORMAL DEFINITIONS OF SHAPE AND SPATIAL RELATIONS

Here, we extend the formal definitions of shape algebras and spatial relations by using the non-graphic symbolism of logic. In doing so, a solid mathematical foundation is established which enables generalized but precise definitions of shape and spatial relations applicable to shapes of any dimension and facilitates reasoning about these shapes in a design context. This also provides an extensible framework for adding new shape algebras and spatial relations to capture additional properties of designs.

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 bulk of the author's research has been spent in the development of logic specifications for definitions of shape and a large set of spatial relations. Due to the technical nature of these definitions, only a few simple ones are illustrated here. Complete, detailed definitions may be found in (Chase, 1996).

Basic Elements

Shapes are composed of finite *basic elements* (or simply, *elements*), which are manipulated in algebras U_{ij}, indicating elements of dimension i in a space of dimension j. For example, U_{02} and U_{12} describe, respectively, points and lines in the plane, U_{33} describes three dimensional (solid) elements in 3-space.

A basic element in U_{ij} is finite and can be distinguished by its *boundary* and a *descriptor*. The boundary divides the design space between an element's interior (finite) and its exterior (infinite). It consists of a set of elements in the algebra of next lowest dimension, e.g., points bound lines, lines bound planes, and so forth. The descriptor provides additional information about an element, including its *carrier*, the infinite element in which it is embedded. Only elements with equal descriptors (considered *cohyperplanar*) may interact in the operations +, –, • and the relation ≤. The most common examples of cohyperplanarity are collinearity and coplanarity.

Geometric 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.

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 multiple element types. In this way, spatial relations are parameterized and can apply to shapes in any dimension.

Boundary Relations. Much of design involves establishing relations between objects based upon their boundaries, e.g., adjacency and abutting. We illustrate here two such relations, share_boundary and surrounded_by.

A single definition of the relation *share_boundary* can be constructed for both lines and regions (and indeed, elements of higher dimension). Two basic elements A and B share a boundary if the product of their boundaries is non-empty, i.e. their boundaries overlap (Figure 5). It should be noted that the logical descriptions presented here are notated in an informal manner in order to facilitate reading. Variables are assumed to be universally quantified unless otherwise specified.

$$\begin{aligned}&\mathit{share\_boundary}(A,B) \leftrightarrow \\ &\quad \mathit{boundary}(A) \bullet \mathit{boundary}(B) \neq \varnothing\end{aligned}$$

For elements A and B of the same type, A is surrounded by B if and only if A is a part of B and they don't share a boundary (Figure 6):

$$\begin{aligned}&\mathit{surrounded\_by}(A,B) \leftrightarrow \\ &\quad A \leq B \ \& \ \neg\mathit{share\_boundary}(A,B)\end{aligned}$$

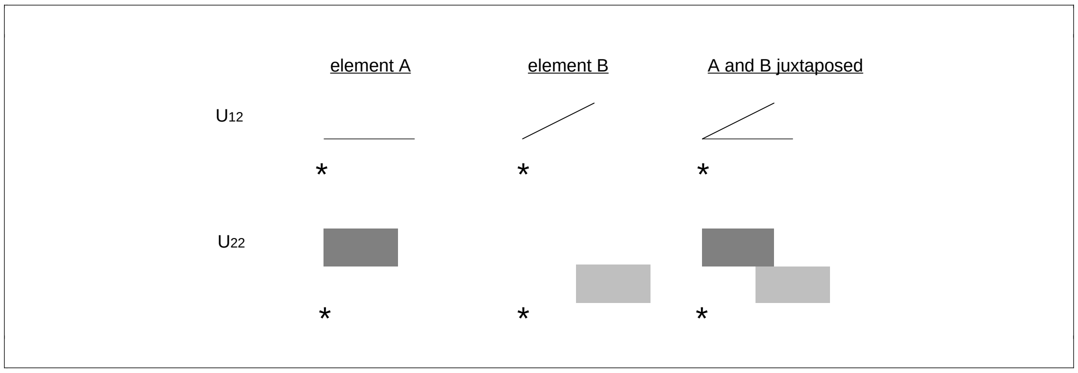

Figure 5. *share_boundary* relation. The lines (in U_{12}) share a boundary (endpoint); the regions (U_{22}) share a portion of their boundary lines.

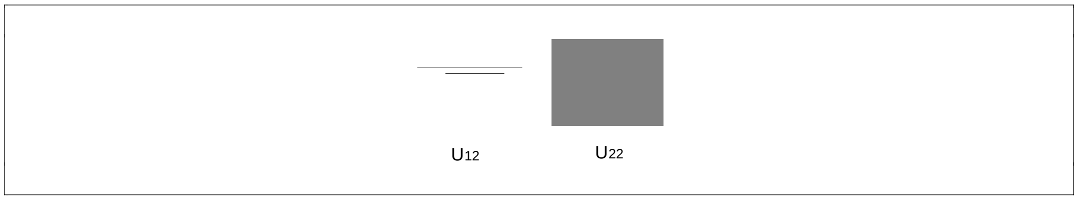

Figure 6. *surrounded* relation. The bottom line is surrounded by the top one (assuming collinearity); the smaller region is surrounded by the larger one.

Dependency Network of Relations. A large set of spatial relations and operations has been constructed by progressively defining common spatial relations based upon primitive definitions and other relations. With this a dependency network can be diagrammed which can provide some insight into how spatial relations interact and the issues involved in a computer implementation of such a network (Figure 7).

It turns out that there is often no one correct way to define a spatial relation, but rather, multiple possibilities for a definition. How a relation is used in practice can be a factor in how one chooses to define it. Since the relations are all interrelated in a network containing recursive relations, there may be multiple ways to define relation *A*, given relations *B*, *C*, etc.

4 APPLICATIONS

Illustrated here are examples from the domains of architecture and geographic information systems, which demonstrate the relative ease with which the spatial relations and operations described above can be used for generating and reasoning about designs, including the ability to represent emergent features.

Architectural Floor Plans

As typical representations in architectural design, we illustrate examples of wall centerline generation and the inference of spaces and emergent subspaces using algebras U_{i2} (points, lines and regions in the plane).

Identification of Wall Centerlines. There has been much research in wall and location identification from plan drawings. The problem can be quite complex; many different methods have been developed to facilitate this process, which generally involves techniques such as line following (Koutamanis, 1990) and constraint management to construct a structural hierarchy of spatial objects such as walls, doors and rooms (Cherneff, 1990). Here, the power of the shape algebras and the logic specification of relations allows us to easily specify the spatial relations needed for such interpretations without focusing on control mechanisms.

Figure 8 illustrates spatial relations resulting from a few of the possible configurations of intersecting walls. A wall segment can be constructed by using two node types as the end conditions of the segment. Although the conditions illustrated here are of perpendicular segments, this is not a necessary condition, as demonstrated in Figure 10.

Figure 9 illustrates a rule which can generate wall centerlines from wall outlines for case 1-2 in Figure 8 (L-shaped intersection), and a design which can be generated using this rule and similar ones. Figure 10 and the formulas following illustrate some of the relations necessary for the construction.

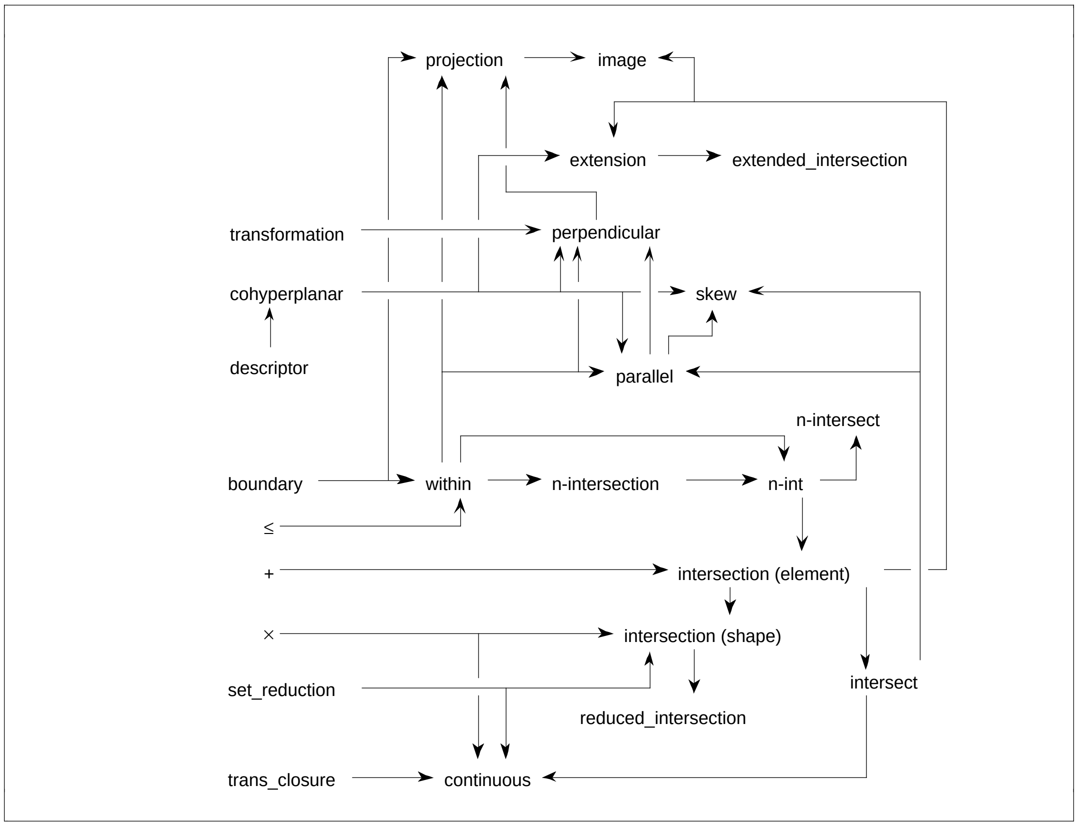

Figure 7. A portion of a dependency network of spatial relations and operations. An arrow from *A* to *B* indicates that the definition of *B* is dependent upon *A*. The recursive nature of relations is not shown here. Definitions with no incoming arrows (on the left of the figure) can be considered primitives.

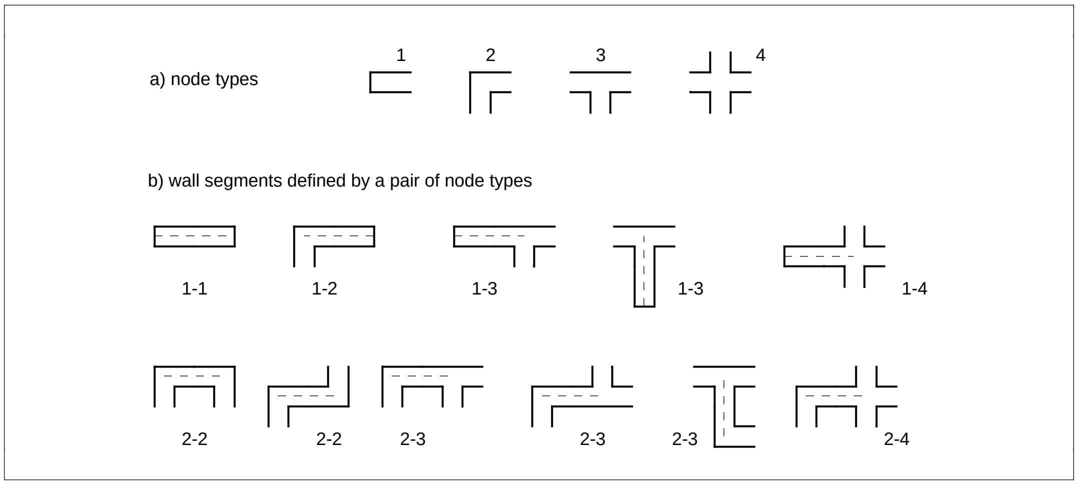

Figure 8. a) Wall segment intersection nodes (up to four). b) Some of the configurations of wall segments defined by pairs of nodes. The dashed line represents the centerline of the wall segment in question.

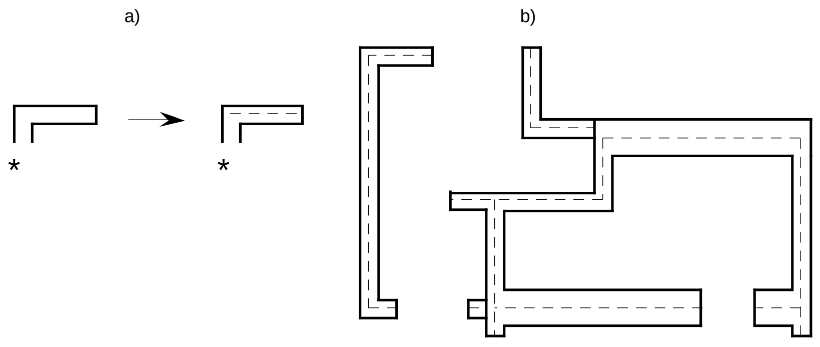

Figure 9. a) Shape replacement rule for wall centerline generation (spatial relations described in Figure 10 and the text). b) A possible floor plan shape generated from multiple invocations of rules similar to a), based on the shapes in Figure 8b.

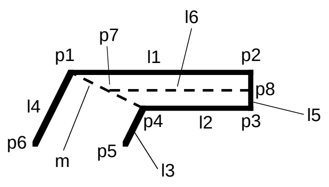

Figure 10. Shape (with elements labeled) for the 1-2 case of wall centerlines.

We illustrate here a few of the necessary spatial conditions for the L-intersection. It is assumed that the points in Figure 10 are in the algebra U_{02}, and that the lines are in the algebra U_{12}. First, there are conditions of parallelism and perpendicularity (specified formally in logic in (Chase, 1996)) in which must be satisfied by certain wall segments:which must be satisfied by certain wall segments:

$parallel(l_1,l_2)$ $perpendicular(l_1,l_5)$
$parallel(l_3,l_4)$

The following conditions set a range of allowed wall thicknesses, i.e. the distance between the two parallel lines. These are conditions of architectural context, based upon the design and drawing scale. Here, *distance* is a function whose value is the perpendicular distance between the two parallel lines:

min_wall_thickness ≤ $distance(l_1,l_2)$ ≤ max_wall_thickness
min_wall_thickness ≤ $distance(l_3,l_4)$ ≤ max_wall_thickness

Some elements and relations can be inferred from the explicit conditions. For example, the centerline l_6 can be constructed as shown below. The points p_7, p_8 and the line m are also inferred. The formula for l_6 indicates that it is constructed in an algebra V_{12} (labeled lines), with endpoints p_7, p_8 and label “centerline”:

$center(l_5) = p_8$ $m = e_{U12}(\{p_1, p_4\})$
$center(m) = p_7$ $l_6 = e_{V12}(\{p_7,p_8\}$, “centerline”)

The *contiguous* relation (which here describes lines which share an endpoint) is used to construct the sequence l_4,l_1,l_5,l_2,l_3, forming a chain of connected line segments:

$contiguous(l_4,l_1)$ $contiguous(l_5,l_2)$
$contiguous(l_1,l_5)$ $contiguous(l_2,l_3)$

The following can also be proven from the given conditions:

$$0 \leq interior\_angle(l_1,l_4) = interior\_angle(l_2,l_3) \leq 90°$$

There are a number of additional required and inferred spatial relations which hold in this example; they have been omitted here for the sake of brevity.

Identification of Spaces. An important function of an architectural design interpretation system is the recognition of spaces. Since architectural spaces are three dimensional, plan drawings cannot always capture their volumetric qualities. However, they do in general serve as adequate notational devices for representing space. The example here simplifies the problem by treating elements embedded in a wall (such as doors and windows) as discontinuities in the wall (where appropriate).

The advantage of the shape algebras over more traditional representations becomes evident here, as generation of the open spaces involves no more than the basic shape algebraic operations performed on the wall segments identified in the previous example. Some emergent features are also easily found, a difficult or impossible task with other representations.

There are several steps to this process, resulting in a figure-ground of wall vs. open space (Figures 11–14). First, wall segments defined from the construction in Figures 8–10 are transformed into individual regions, then reduced to a shape *W* of maximal regions. The shape *W* is subtracted from its convex hull (the smallest convex region enclosing the shape), producing a shape *S* representing the open space in a figure-ground relationship with *W*.

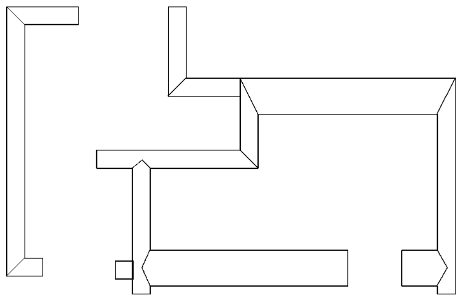
Figure 11. Boundary lines (U_{12}) for all wall segment regions.

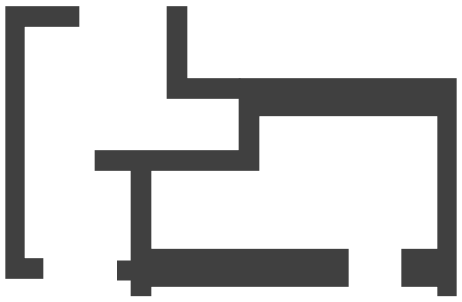
Figure 12. Reduction of wall segment regions into a shape *W* of maximal elements (U_{22}).

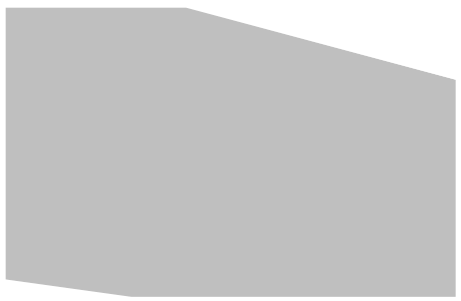
Figure 13. Convex hull of the shape *W* in Figure 12.

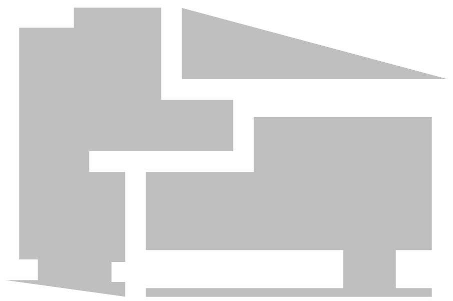
Figure 14. The shape *S* representing open space constructed by *convex_hull*(*W*) – *W* .

Emergent subspaces. While the shape S does represent continuous spaces in terms of maximal regions, in general we are interested in subspaces of these larger spaces. Although with plan drawing alone we cannot get a complete picture of all possible subspaces of interest, we can identify some of them. It should be noted that any subshape of S which contains non-maximal parts of S as maximal elements qualifies as an emergent subspace. However, there are an infinite number of these subshapes (Figure 15), most of which are not of interest. Rules therefore need to be established for the identification of meaningful subspaces.

A common way of identifying these subspaces is to consider the extension of wall segments (and also ceilings) which are the boundaries of a space. Rules such as those in Figure 16 can be used to construct "virtual walls", which are then used as boundary elements for emergent subspaces (Figures 17 & 18). Any two contiguous subspaces can be combined into a single larger subspace. An additional benefit of this representation of spaces as regions in U_{22} is that a room perimeter is represented by the boundary elements of its spatial region. Perimeter wall lengths and surface areas can easily be computed by forming the product (•) of a region's boundary elements and the shape consisting of wall edges.

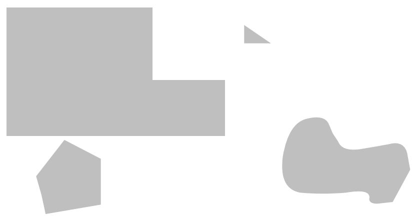

Figure 15. Some emergent subspaces of the shape *S*, most of which are not of interest.

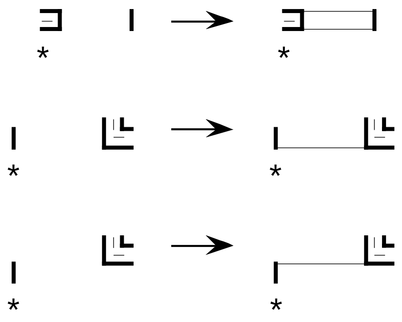

Figure 16. Rules for the extension of wall segments into "virtual walls".

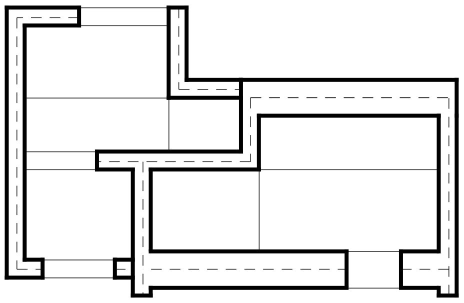

Figure 17. Example with "virtual wall" extensions added.

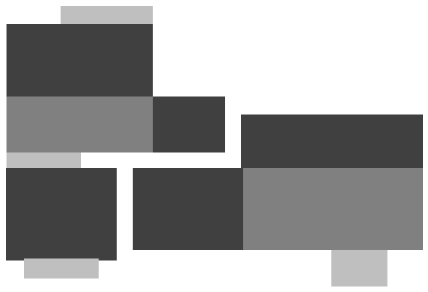

Figure 18. Subspaces constructed from possible new boundary lines.

Views between spaces. Among the emergent features that could be identified using the representations described above is that of views between spaces. Figure 19 illustrates this by defining this feature as a parameterized shape consisting of portal jamb lines, "virtual wall" lines at these portals, and a region (shaded) which is part of a space. By varying restrictions upon the parameters, one can identify general vistas between spaces as well as more classical enfilades characteristic of the designs of Palladio (Stiny, Mitchell, 1978).

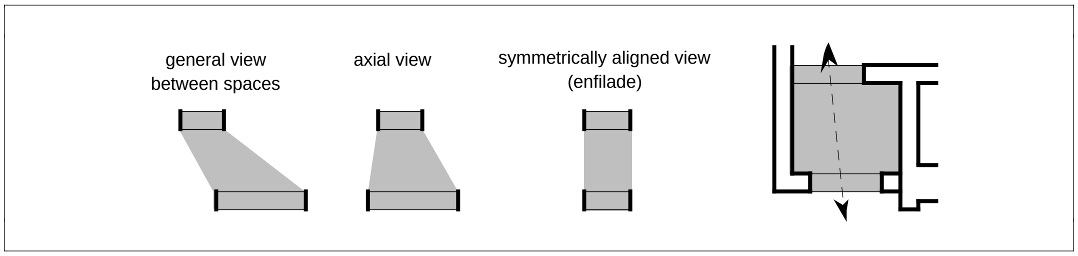
general view between spaces
axial view
symmetrically aligned view (enfilade)

Figure 19. Emergent views between spaces defined by portals. Parametric conditions are not detailed here.

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 their data structures to represent this closed set of features. Thus, the possibility of emergent features and properties is limited or nonexistent.

A simple example illustrating the emergent feature *accessible* uses the definition of a *continuous* shape, one in which all elements are "connected" (Figure 20). In other systems, accessibility problems are generally handled by connectivity graphs which are explicitly constructed for this purpose. Thus, the structure of the map, i.e. its connectivity relations, must be known before drawing it; emergence is not possible. Here, the graph emerges from the implicit connectivity relations among the various elements of the shape.

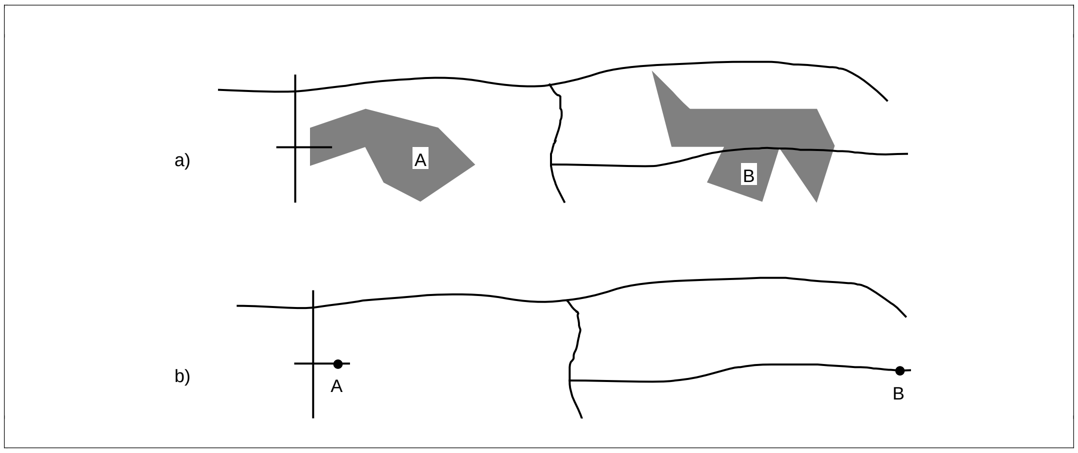

Figure 20. Accessibility. a) between regions; b) between points. *A* is accessible to *B* because there is a continuous shape containing *A*, *B* and a set of (road) lines.

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 21a). 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 complexity of computation has been moved from the query language to the inference mechanism. The queries prove to be relatively easy to formulate using such intuitive relations as *boundary*, *within* and *intersection* (Figure 21b), and allow for the possibility of emergent features, impossible in the more traditional relational representation.

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:

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

Figure 21 Comparison of queries on a GIS database. *a*) Relational algebraic query; *b*) the same query using relations based on shape algebras.

5 IMPLEMENTATION ISSUES

The model introduced in this work has been developed in a descriptive rather than operational manner. By doing so, 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. If one wishes to develop a computer implementation of the model, the issues of data structure and computational complexity must be considered.

Computations using shape grammars involve the subshape recognition problem, i.e. the determination of applicable rule invocations by searching for occurrences of a rule pattern within a design. The number of applicable subshapes can range from none to an infinite number, in general growing combinatorially with the number of maximal elements in a shape. Therefore, a goal is to reduce the set of possible rule applications to a finite, manageable number. Research by Tapia (1996) focuses on presenting a manageable number of choices to the designer when computing with shape grammars. Several ways to accomplish this include placing restrictions on the computational mechanism, numerical representation of shapes, and rules and their application.

Thus, implementation invariably requires 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.

It is expected that a deductive database (Gallaire, Minker, 1978) will be used 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 model.

6 CONCLUSIONS

The combination of shape algebras and symbolic logic has been shown to be a powerful tool in the specification of design systems. This was done by constructing a model of shape and spatial relations from first principles of geometry, topology and logic. The model 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 here provides a generalized parameterization schema for shapes and spatial relations 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.
- Spatial relations are constructed in a layered approach, beginning with base primitive operations, and building upon them to generate high level relations and operations which are natural to design. This method allows one to examine formal properties of relations and identify issues which may affect implementation. It also proves to be a natural and intuitive way of development.
- 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.

In summary, this work demonstrates that 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 minimal modification of the model semantics appears to be great. It is hoped that future research will adopt this approach (focus on the power of the formal model), thereby overcoming the traditional bias of favoring implementation at the cost of representation.

REFERENCES

Benhamou F, Colmerauer A, 1993 *Constraint Logic Programming: Selected Research* (MIT Press, Cambridge, Mass.)

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, 1993, "The use of multiple representations to facilitate design interpretation" in *ARECDAO '93,* Barcelona, Spain, March 30-April 1, 1993

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

Cherneff J, 1990, "Knowledge Based Interpretation of Architectural Drawings", report, R90-13 (IESL 90-05), Intelligent Engineering Systems Laboratory, Dept. of Civil Engineering, Massachusetts Institute of Technology

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, 1978, "The representation of design problems and maintenance of their structure" in *Artificial Intelligence and Pattern Recognition in Computer Aided Design* Ed J C Latombe (North-Holland, New York) 335-357

Eastman C M, Bond A H, Chase S C, 1991, "A formal approach for product model information", *Research in Engineering Design* **2** 65-80

Flemming U, 1987, "More than the sum of parts: the grammar of Queen Anne houses", *Environment and Planning B: Planning and Design* **14** 323-350

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

Hofstadter D R, 1979 *Gödel, Escher, Bach: an Eternal Golden Braid* (Basic Books, New York)
Hoskins E M, 1973, "Computer aids in system building" in *Computer-aided Design* Ed J Vlietstra, R F Wielinga (North-Holland, Amsterdam) 127-140
Kirsch J L, Kirsch R A, 1986, "The structure of paintings: formal grammar and design", *Environment and Planning B: Planning and Design* **13** 163-176
Knight T W, 1992, "Designing with grammars" in *CAAD futures '91* Ed G N Schmitt (Vieweg, Wiesbaden) 33-48
Knight T W, 1995, "Constructive symmetry", *Environment and Planning B: Planning and Design* **22** 419-450
Koutamanis A, 1990, *Development of a Computerized Handbook of Architectural Plans* Ph.D dissertation, Delft University of Technology
Kowalski R A, 1979 *Logic for Problem Solving* (North-Holland, New York)
Krishnamurti R, 1981, "The construction of shapes", *Environment and Planning B* **8** 5-40
Krishnamurti R, 1992, "The arithmetic of maximal planes", *Environment and Planning B: Planning and Design* **19** 431-464
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
Stiny G, Mitchell W J, 1978, "The Palladian grammar", *Environment and Planning B* **5** 5-18
Stiny G, Mitchell W J, 1980, "The grammar of paradise: on the generation of Mughul gardens", *Environment and Planning B* **7** 209-226
Tapia M A, 1996, *From Shape to Style, Shape Grammars: Issues in Representation and Computation, Presentation and Selection* Ph.D dissertation, University of Toronto