An Algebraic Approach To Shape Computation (A Position Paper)*

Rudi Stouffs and Ramesh Krishnamurti

1 Motivation

There can be no argument that the use of computers in design is ratiocinative. Researchers and design practitioners have long been interested in questions on (i) what is design? (ii) how do we design? (iii) how can we use computers to support design without constraining either the process or the artifact? and, more importantly, (iv) how can we use computers to 'liberate' the process of design? To some extent, we attempt to address the latter two questions.

It is imperative that the role and use of the computer in design, - from our concerns - in particular in architectural design, be clearly recognized. Computers have long been hailed to offer speed of action and improve efficiency. As a result, most computer-aided design systems are driven by "the cut/copy/paste economies of digital data" [1]. Many architects or architectural firms are tricked into CAD by unfounded claims of increase in productivity. Six months later they wonder: what went wrong? Setting aside discussions on training and adaptation, there is a more powerful consideration. Efficiency does not necessarily improve effectiveness. Speed and accuracy are not synonymous with quality.

We consider the computer as a tool for design and presentation, as a *means of communication* between the design professional and client, among professionals and during the design process. Proverbially, a drawing is worth a thousand words. In design, especially architecture, a drawing is worth much more. In architecture, we use sketches, plans, elevations, sections, isometries and perspectives as the means of communication. Drawings are essential to the design process. The wealth of information contained in drawings justifies the time taken for their creation.

It is in the area of drafting and presentation that the computer chiefly shows its strengths: its capabilities for visualization are well known. In other areas, say in the early design phase, the use of computer lags far behind.

Visualization, in view of the current 'state of the art' in CAD, relates most commonly to the production of 'pretty pictures.' However, visualization is not about pretty pictures in the same way that architecture is not about pretty buildings. Instead, visualization has the potential of being a powerful tool in the evaluation of designs. Large amounts of information can be successfully displayed in computer-generated images in a way that a human being - designer - can readily process. Indicators of building performance such as lighting and acoustics are obvious examples; urban analysis and design is another in which visualization can assist the designer.

We argue that visualization encompasses every stage in the design process. It has even greater potential in the early design stages, where it enables the explicit formulation of three-dimensional models in the thought process. We have explored an alternative modeling approach, based on an algebraic model for shapes, that allows for shapes to be dealt with and manipulated in indeterminate ways. The algebraic approach to shape computation allows for an intuitive and powerful method of reasoning with shapes, in a visual interactive generative design environment.

*In workshop notes of *Reasoning with Shapes in Design. Artificial Intelligence in Design '94*, pp. 50-55, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, (15-18 August) 1994.

2 Emergent Shapes

Algebraic models for shapes have been adopted elsewhere in shape grammars [6] [7] [8]. In these models, a shape is specified as an element of an algebra that is ordered by a *part relation* and closed under the operations of sum, product and difference and the affine transformations. Fundamental to an algebraic model is that, under the part relation, *any* part of a shape is a shape and can be manipulated as such: thus, users can deal with shapes in indeterminate ways. This is quite distinct from the selection process in current CAD approaches where the only objects that can be selected correspond to those (prescribed minimal entities) that have been predefined by the data-structures.

The part relation for shapes leads naturally to the concept of *emergent shapes*: A shape defines an infinite set of shapes, all part of the original shape, that *emerge* under the part relation. Emergent shapes are not originally envisioned as such, they only become explicit when manipulated as such. Computationally recognizing emergent shapes requires determining a transformation under which a specified similar shape is a part of the original shape. A shape rule constitutes a formal specification of shape recognition and subsequent manipulation. A shape rule has the form $lhs \rightarrow rhs$; *lhs* (left hand side) specifies the similar shape to be recognized, *rhs* (right hand side) specifies the manipulation leading to the resulting shape. Then, shape rule application consists of replacing the emergent shape corresponding to *lhs*, under some allowable transformation, by *rhs*, under the same transformation. A shape grammar combines a set of (semantically related) shape rules into a formal rewriting system for producing a language of shapes.

The concept of emergent shapes is highly enticing to design search [4] [9]; the specification of shape rules leads naturally to the generation and exploration of possible designs. However, the concept of search in this context is more fundamental to design than its generational form alone might imply. Any mutation of an object into another one, or parts thereof, whether as the result of a transformation or operation, constitutes an action of search.

A rule constitutes a particular compound operation or mutation, that is, a composition of a number of operations and/or transformations that is recognized as a new, single, operation and can be applied as such. Under this algebraic model, any such composition defines a valid mutation. Similarly, a grammar is a collection of rules or operations that yields a certain set (or language) of designs from an initial design. As such, the creation of a grammar is only a tool that allows a structuring of a set of operations that has proven its applicability to the creation of a certain set of designs, rather than a framework for generation.

Particular to a rule is that the transformation is not specified but is selected from a body of transformations, e.g., the similarity transformations, according to a (specified) constraint. For a shape rule, the constraint specifies that the object of mutation is a part of the given shape. The process of determining one (or all) transformation for which this constraint holds is termed subshape detection; it relies fundamentally on the algebraic model for shapes and the corresponding maximal element representation.

The maximal element representation for shapes [3] formalizes the part relation that underlies the algebraic model. This representation is particularly suited to answer the following two questions: (i) are two shapes identical? and (ii) is one shape a part of another shape? Together, the algebraic model and maximal element representation define a representation scheme for geometric modeling that is both unique and unambiguous. The algebraic model is mathematically uniform for shapes of all kinds, including curved shapes, and applies to non-geometric elements or attributes as well. The proofs of these assertions can be found in Stouffs [10].

3 Mixed-Dimensional Models

The model provides a natural and intuitive framework for mixed-dimensional shapes. The creative process of design is not so concerned with the dimensionality of each and every individual, even if the final product is a composition of (purely) three-dimensional solid elements. Even in the evaluation of a design, an abstraction is often more valuable. Structurally, walls can be represented as plane segments of aliquot parts and attributes; a true three-dimensional model would, unless appropriately approximated, needlessly complicate a structural evaluation. In general, even a single component may be represented as different elements of mixed dimensionality, each element projecting information for a specific application.

Mixed-dimensional models have found recent support in solid modeling [2] [5]. By adhering to Boolean set operations in a Euclidean space, these mixed-dimensional approaches come at the expense of intuition in operations and ease of conception. For example, the notion of solids touching is completely absent.

The algebraic model is based on a part relationship that can be freely defined as long as it constitutes a partial order relation. It is advantageous to define the part relation between elements of the same type, e.g., between elements of the same dimensionality. Elements of the same dimensionality belong to the same algebra. A shape may consist of more than one type of element, in which case it belongs to the algebra given by the Cartesian product of the algebras of its element types.

The algebraic model for shapes is independent of an underlying geometric framework. We distinguish the geometry of shapes from Euclidean geometry, instead, consider the Euclidean space to embed the elements in their final representation [10]. An element of an algebra given by the Cartesian product of the algebras of its spatial element types is compositely embedded in a Cartesian product of Euclidean spaces. These may ultimately be visualized as either a single space or a multiplicity of spaces. When embedding elements of different types in the same Euclidean space, these elements cohabit without interference. By virtue of the Cartesian product, algebras can be used and applied in combination without mutual interference. This parallel construction allows for disparate objects not only to coexist peacefully in a single (spatial modeling) world, but also to be conceived of as one at the same time as being handled and operated on quite differently, in this conceptually unified approach.

The Cartesian product of algebras is not restricted to a composition of different dimensional algebras. Generalized as such, compound shapes can be regarded either as complex shapes that are composed of segments from different algebras or as consisting of shapes that are coordinated or related. For example, consider a set of drawings from amongst plans, elevations and sections of a same building. Each drawing can be considered a shape, as an element of a two-dimensional algebra, while the set of drawings can be regarded as a single shape that is an element of a Cartesian product of algebras (here, all of the same dimensionality).

4 GRAIL

The algebraic model is more than a model for the representation of shapes. It is a concept for spatial modeling that applies as well as a model of interaction for computational design. As such, it has the advantages of being clear, straightforward, yet inclusive: many, if not the most common computational design activities can be expressed using the algebraic model. Upon defining the appropriate algebras (e.g., for design and selection), the operations of creation and deletion, but also of selection and deselection, can be expressed readily in terms of the shape operations of sum and difference in these algebras.

These explorations into shape computation and modeling form a part of the GRAIL project. In it, we explore the challenges and the implications that the algebraic model for shapes poses to computational design in an interactive generative design environment. GRAIL, currently, has the following objectives:

- exploring different ways of interacting with shapes, as detailed above. We are developing shape selection methods that derive from cross-algebra operations, where a selector shape may be in a different dimensional algebra to the shape that is ultimately selected (see figures 1 and 2).

- exploring different ways of creating and organizing shapes. The algebraic model is extensible to non-geometric elements and attributes. Elements may be layered and grouped according to both spatial and non-spatial features. In addition, shapes may have derivational dependencies. A uniform method for interactively and programmatically organizing and manipulating shapes through a forest-like organizational structure is being explored. We eventually hope to handle more general graph relationships.
- exploring generative approaches to spatial modeling. We are exploring ways of interactively defining spatial rules and rule application. Part of the GRAIL project subsumes spatial grammar interpreters within a three-dimensional spatial representational environment for modeling, analysis and visualization.

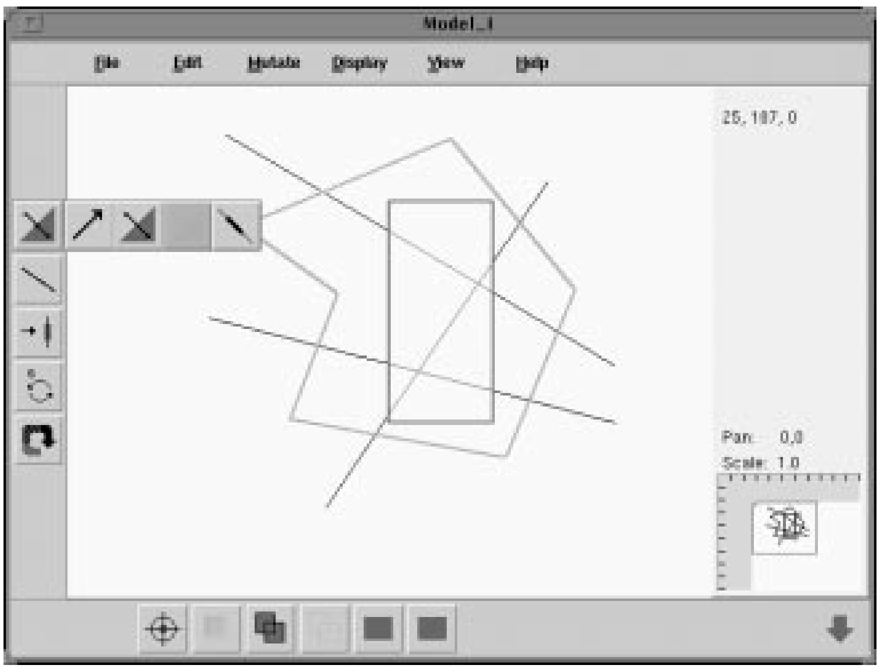

Figure 1: A two-dimensional (rectangular) selection shape allows for the selection of both one-dimensional and two-dimensional spatial elements (snapshot from the current GRAIL interface).

In addition, a programming language interface is planned for GRAIL so that a number of applications can be built upon it. The design is modular and data integrity is ensured by having each module exercise sole control over the creation, update and maintenance of all data under its purview. Robustness is achieved by ensuring that all operations are algebraically sound. GRAIL will be designed to serve as a modeling and design system as well as a functional toolkit for application developers.

5 Where Next?

The algebraic approach is general and complete; however, it lacks the necessary expressive power to gain acceptance without further extension. An appropriate representation for algebraic surfaces and curves needs to be specified that fits within the algebraic model. Part of the basic attractiveness of the algebraic model is its ability to include non-geometric attributes within the model. We can easily add constant, symbolic or numerical information to shapes. In each of these examples, the augmented shapes have been derived from shapes by associating symbols, labels or properties, with the spatial elements. Labeled points, for example, play an important role in both spatial representation and mainpulation. They serve to guide the rule matching process through identification and classification

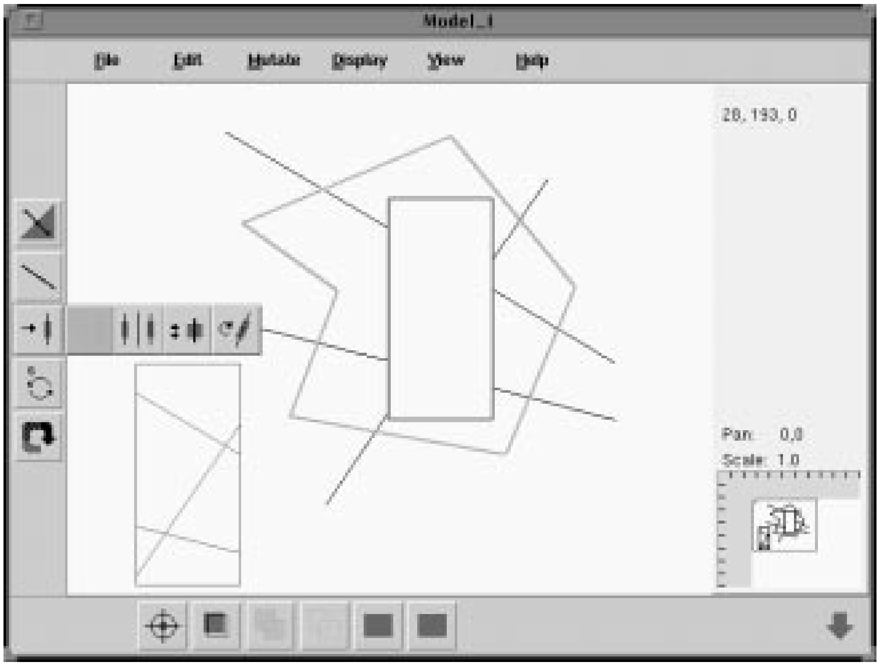

Figure 2: A translation of the selected shape illustrates the unified manipulation of one-dimensional and two-dimensional spatial elements (snapshot from the current GRAIL interface).

of the rules. They could also be viewed as semantic extensions to what are fundamentally syntactical expressions.

It follows that the algebraic operations, and the underlying part relation, need to be redefined in order to deal correctly with the associated information. When considering labels, this can be achieved by an ordinary set approach: the sum of two identical points, each with a set of labels, is the single point with the union of both sets associated to it. It may seem less intuitive for segments of a different dimensionality. Instead, consider shapes augmented with weights or colors. For weights (e.g., numerical attributes), the part relation is obvious and the algebraic operations follow naturally (the sum of two identical segments with different weights is the single segment with the maximum of both weights). For colors, a ranking may be specified that maps the colors to real values similar to weights or, otherwise, a three-dimensional color coding (e.g., RGB or intensity, saturation and hue) may be considered with an appropriate part relation.

Stouffs [10] introduces a characteristic function to a shape that, in the case of weights, assigns to any infinitesimally small shape a weight as a single value. Upon embedding weighted shapes in the Euclidean space, the result is similar to a spectral (discontinuous) function that specifies a 'height' for every point. The characteristic function of two weighted shapes equals the 'sum' of the characteristic functions of both shapes, where the operation of sum is defined for the algebra of weights, that is, the sum of two weights is the maximum value of both weights.

In general, when augmenting the shapes with non-spatial information, we only need to redefine the range for the characteristic function, e.g., we create one or more extra dimensions (e.g., three in the case of colors) that define the range space for the characteristic function. In the case of labeled shapes, for a given set of labels L, the range of the characteristic functions is the power set of L and the algebraic operations correspond to the set operations.

A similar treatment can be applied to procedural or functional weights attached to shapes. Here, a function, such as a daylight or structural analysis, is invoked only when certain conditions hold, which are governed by appropriate characteristic functions associated with the shapes in question. However, the details of this scheme remain to be worked out.

References

[1] C. E. Dagit, 1993. "Establishing Virtual Design Environments in Architectural Practice," in *CAAD Futures '93* (eds., U. Flemming and S. Van Wyk), pp. 513-522, North-Holland, Amsterdam.

[2] E. L. Gursoz, Y. Choi and F. B. Prinz, 1991. "Boolean Set Operations on Non-Manifold Boundary Representation Objects," *Computer Aided Design* 23, 33-39.

[3] R. Krishnamurti, 1992. "The Maximal Representation of a Shape," *Environment and Planning B: Planning and Design* 19, 267-288.

[4] W. J. Mitchell, 1993. "A Computational View of Design Creativity," in *Modeling Creativity and Knowledge-Based Creative Design* (eds., J.S. Gero and M.L. Maher), pp. 25-42, Lawrence Erlbaum Associates, Hillsdale, N.J.

[5] J. R. Rossignac and A. A. G. Requicha, 1991. "Constructive Non-Regularized Geometry," *Computer Aided Design* 23, 21-32.

[6] G. Stiny, 1980. "Introduction to Shape and Shape Grammars," *Environment and Planning B: Planning and Design* 7, 409-462.

[7] G. Stiny, 1986. "A New Line on Drafting Systems," *Design Computing* 1, 5-19.

[8] G. Stiny, 1991. "The Algebras of Design," *Research in Engineering Design* 2, 171-181.

[9] G. Stiny, 1993. "Emergence and Continuity in Shape Grammars," in *CAAD Futures '93* (eds., U. Flemming and S. Van Wyk), pp. 37-54, North-Holland, Amsterdam.

[10] R. Stouffs, 1994. "The Algebra of Shapes," Ph.D. dissertation, Dept. of Architecture, Carnegie Mellon University, Pittsburgh, Pa.