MODELLING SHAPES FOR DESIGN AND COMPUTATION: Hybrid Algebras for the Designer

PAUL VARGHESE, ARIF MERCHANT
Department of Architecture & Regional Planning
Indian Institute of Technology Kharagpur 721302, India.
{varghese, merchant} at arp.iitkgp.ernet.in

Abstract. A scheme for using shapes from hybrid algebras as the basis for modelling is examined, which offers advantages over conventional design modelling methods, such as pure solid models.

1. Introduction

Note: The reference to 'shape modelling' as used here comes from the shape grammar formalism. It does not refer to other methods that might have a like name or similarities in the methods seen.

Conventional modelling methods such as solid modelling are in vogue within the design community. Without doubt, they serve a useful role in the development of computer-aided-design (CAD). However, there are limits that these processes possess, that aid the designer only to a certain extent. New methods for designing need to be explored and put to use. In this pursuit, we explore the alternative of using shapes (from the shape grammar formalism) as an advanced system for representing design.

1.1. DESIGNING

Architects and designers in their initial design scribbling make use of conceptual structures that start as squiggles, bubble-diagrams and/or functional relationships that are connections between functions, spaces, etc.

At the same time, architects and designers, especially those who use a pencil-and-paper approach, know that the freedom that this gives in the conceptual stages, probably cannot be matched when starting off on a modelling system.

Part of the reason is the freedom to work without defining the underlying 3-dimensional elements of the point, line, plane or solid which are trickier to represent and work with using conventional modellers, as opposed to 'cruder' methods like the tracing sketch paper or the cardboard model.

1.1.1. How Architects use CAD

Computer-Aided Design (CAD) is the gift of technological development from the computational sciences to the design processes, used from architectural design and mechanical engineering to the product or graphic standards.

Currently, it might seem that CAD is used extensively in architectural offices, but a closer look will reveal its application mostly for representation, production and documentation environments. The dissatisfaction with CAD in practice is that it has merely replaced pencil drawings, even if more speedily, accurately, or for various adaptations in representations.

Thus, the implication is that the underlying or primary processes of design have remained largely unaddressed. However, this might not be entirely accurate, and progress is seen as the preliminary design stages too. The available software, however, has not replaced the pencil-and-paper environment.

Thus, one needs to delve deeper into the fundamental processes of design practices to comprehend the manner in which architects design. However, the aid to this is the pencil on paper representing the primary elements from points to solids.

1.1.2. Modelling with CAD

The use of CAD has become ubiquitous in architecture. The best of systems use the basic functions. Operations on forms use the Euclidean (including translation, rotation, scaling and mirroring operations) and Boolean operations (which are the sum, the difference and the complement). With these basic operations in place, one can do much sculpting necessary for the creation of forms.

2. Designing with Solid Models

Solid modelling is one of the modern tools of industry, which has been developed from the 1970s onwards, primarily for the automobile and the aircraft engineering industries. However, its use has simultaneously crossed over to mechanical engineering, architecture and the construction industry. Its applications have also found use in all kinds of imaging requirements including medicine and the movie-making industry.

Modelling with 3D forms/solids normally utilized one of the basic classes of: (i) wire-frame representations (ii) instantiation or parametrized shapes (iii) sweep representations (iv) cell decompositions (v) constructive solid geometry (CSG) or (vi) boundary representation (B-rep). Hybrids and combinations of the above are also used. For a comprehensive review of the various types, uses, potential, etc., refer Mortenson (1985), Mantyla (1988).

Of the various classes used in solid modelling, Boundary Representation (B-rep) and Constructive Solid Geometry (CSG) have shown themselves to be the methods that are in proficient use, and have held promise for the design environment. However, they still have not fulfilled the requirements early design.

2.1. LIMITATIONS OF SOLID MODELLING

Solids models of the types listed are used in CAD systems to model shapes and forms. However, there are limitations to these systems.

One drawback in particular is that solids modelling can be used only for creating or manipulating solid forms. Solid forms enclose volume. Most solid models are formed from point-sets that validate the presence of enclosed volumes only. In such systems, planes or lines cannot exist independently, and are then removed from the modelling space.

These inadequacies have been known, especially in the context of design environments, and the efficacy of such systems is partial. The need of architects and designers are tools for the design processes, where modelling with all the elements (of points to solids), can be simultaneously and independently allowed.

2.1.1. Progress in the Theory of Solids

Rossignac and Requicha (1991) considered the possibility of bringing in the other lower dimensions by not regularizing the solids, termed constructive non-regularized geometry (CNRG). It was primarily as a means for interrogating additional properties of solids such as its boundaries. It however, remains incomplete for use in design. As much as this might not have been crucial for modelling or in a production environment, the real need is for multi-dimensional elements. However, there is still scope for development in this direction.

3. Shape Grammars and Algebras

Shape grammars are rigorous formal and mathematical structures that generate and manipulate shapes in design environments. Shapes can be composed using the various shape operations particularly the Euclidean and Boolean functions (Stiny, 1980). Further, as theoretical structures, they go beyond the mere issues of representation to those of rule-based generation, manipulation of shapes, shape recognition, etc.; some are still limited by theoretical considerations of time, memory, complexity, etc. (Chase 1987, Gips 1999).

3.1. SHAPE ALGEBRAS

Stiny (1991) proposed the use of shape algebras, within the framework of the shape grammar (SG) formalism, as a redefined partitioning of the formal elements that compose shapes. Shape algebras are used to compose shapes from the elements of points (U_0), lines (U_1), planes (U_2) and solids (U_3).

Theoretical shape algebras allow compositions of shapes from these elements in any combination and without restriction. They allow interactions within their respective algebraic structures by the Euclidean and Boolean operations.

Shape algebras hold promise for the designer interested in doing design in multi-dimensional environments. Though this is a representational consideration as part of the shape grammatical formalism, it is one required step which will initiate the process of computing with shapes into the design environment. Future systems could look at shape-algebraic structures to model shapes in any dimension. The need for multidimensional computing essentially promotes the development of non-manifold algebras within SGs, and in the overall design process.

The uses/need of boundaries and carriers serve the design process (Stouffs, 1994) sometimes in the representation and might vary between designers, but could be essential for designing.

3.1.1. Multi-dimensional Modelling

The importance of computing with shape algebras has been expounded as a key opportunity of the designer, it is perhaps the article by Knight (2003) on computing with ambiguity that it became clear (at least to one author), that the need for inter-algebraic computation was critical and possible for the designer, i.e., that it would serve well as part of the user interface. However, it should be clarified that there is mention of cross-algebraic computation, without any elaboration, by Stouffs (1994) earlier.

Understanding this, we could compute with shape algebras such that elements from across the algebras can be operated upon. In reality, there is no real procedure that would allow these operations. Therefore, it becomes necessary for the modeller to simulate these operations for the users' convenience.

3.2. SHAPE MODELLING

The possibility of using solid modelling methods for shapes was examined (Varghese, 1993). It concluded that even considering the state of the art in computational methods and technology, only limited implementation would be possible because of the design restrictions. However, any kind of real implementation would be an advantage over conventional systems.

The implementation of shape grammars as a practical technology is still some way away. Varghese (2002) further proposed the setting up of design environments using the shape (grammar and) modelling environment. This, however, would be a long-term endeavour, which cannot be looked at in isolation, but needs progress along several fronts. Only partial implementation possibilities or strategies were perhaps realizable (Gips, 1999).

4. Computing with Ambiguity

“Computing with shapes (in shape grammars) is filled with ambiguities.” (Stiny, 2001).

Ambiguity can be of several types (perceptual, nonsense objects, ambiguous grammars) outside the SG formalism and/or within it. In the traditional design process, one constantly uses a combination of differing elements in differing ways, and computations with shapes regularly deal with ambiguous considerations.

Ambiguity in shape grammars has been classified into two main types (Knight, 2003)— part ambiguity and representational ambiguity. In part ambiguity, we are concerned more with the computational effects of rule application, where the intent, the parts and resultants may not emerge as expected from the computation.

In computing with representational ambiguity, we have differing types of algebras that Knight proposes, depending on the purpose and manner of use. However, they provide rich environments for understanding and working in design. Knight (2003) discusses the idea that many types of computations are possible depending on the interaction between the elements, as also the environment. She divides the possibilities into several types like composite algebras, ordered algebras, hybrid algebras, medial algebras, etc. Of these categories, we are particularly interested in *hybrid algebras* for our purposes.

4.1. DESIGNING WITH AMBIGUITY

From another viewpoint, the range of shapes and its operations as described by Knight really relates to individual preferences or choices for operations. For the sake of computation, all possibilities in the algebra can be used, but strictly, the designer may prefer to bias his design process using only a few of them, depending on his/her style of working, whim or spur-of-the-moment consideration. Any way it goes, this really points to the designer as the person who makes/takes the choice of process, if/when all the processes are open for use in the computation. It translates to the biases of the designer, better termed ‘customization’.

4.1.1. Hybrid Algebras

Hybrid algebras are those, which make elements from differing algebras, interact with one another, as classified by Knight (2003).

Knight further discusses the possibility of looking at shapes operating as point-sets as used in point-set topology, where all elements can be viewed as composed of points. Under such circumstances, the elements can be made computationally interactive within the limitations of such operations.

4.1.2. Operations using Hybrid Algebras

From the concept that hybrid algebras have a direct possibility of use, we understand that the implications are quite vast. Whereas previously there were very limited operations that were possible using pure solid modelling (refer Table 1), i.e., purely the operations between solids, it was felt that the normal shape grammar formalism would offer additional operational capabilities. However, if hybrid algebraic operations were to be put in place, the options for interaction multiply vastly. The comparative chart shown in Table 1 indicates that the number of operations has increased. However, this does not complete the story.

TABLE 1. Operation interaction table.

Operation Interaction Table — Models & Operations	*Solid Modelling*					*Normal Shape Grammars*					*Hybrid Shape Algebras*				
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$U_0 \times U_0$	n	n	n	n	n	y	y	y	y	y	y	y	y	y	Y
$U_0 \times U_1$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	Y
$U_1 \times U_1$	n	n	n	n	n	y	y	y	y	y	y	y	y	y	Y
$U_0 \times U_2$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	y
$U_1 \times U_2$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	y
$U_2 \times U_2$	n	n	n	n	n	y	y	y	y	y	y	y	y	y	y
$U_0 \times U_3$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	y
$U_1 \times U_3$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	y
$U_2 \times U_3$	n	n	n	n	n	n	n	n	n	n	n	y	y	y	y
$U_3 \times U_3$	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y

A look at Table 2 reveals briefly shows the operations and their possible resultants indicated. In actuality, the number of possibilities in terms of resultants are far more than those shown, and would depend on the manner in which those of the operations have been set up. The list in Table 2 is merely suggestive, and *not actually exhaustive*. It does not enumerate all possibilities.

Thus, it can be summarized that the number of operations are possibly much more than the user probably requires, and for that matter, can keep track of, and hence make useful.

4.1.3. Modelling the Shape

A schematic structure for the representation of shapes is shown in Figure 1. The structure is made dynamic so that it can accommodate differing elements, from points to solids, by a process of *type conversion*. The point transforms into a line, a line into a plane, etc. The basic process is of conversion from one element type

TABLE 2. Operations and Potential Resultants using Hybrid Algebras.

$U_{1,3}$	U_a x U_b	Operations & Potential Resultants				
		a + b	a M b	a / b	b / a	a D b
POINT	point x point	*1 point*	*1 point*	*0*	*0*	*0*
LINE	point x line	*1 point + 1 line*	*1 point*	*1 point*	*1 divided line*	*1 divided line*
-	line x line	*2 lines*	*1 point*	*1 divided line*	*1 divided line*	*2 divided lines*
PLANE	point x plane	*1 point + 1 plane*	*1 point*	*1 point*	*1 plane*	*1 plane*
-	line x plane	*1 line + 1 plane*	*1 partial line*	*1 partial line*	*1 plane*	*1 partial line + 1 plane*
-	plane x plane	*1 combined plane*	*1 partial plane*	*1 partial plane*	*1 partial plane*	*2 partial planes*
SOLID	point x solid	*1 point + 1 solid*	*1 point*	*1 point*	*1 solid*	*1 solid*
	line x solid	*1 line + 1 solid*	*1 partial line*	*1 partial line*	*1 solid*	*1 partial line + 1 solid*
	plane x solid	*1 plane + 1 solid*	*1 partial plane*	*1 partial plane*	*1 solid*	*1 partial plane + 1 solid*
	solid x solid	*1 combined solid*	*1 partial solid*	*1 partial solid*	*1 partial solid*	*2 partial solids*

to another. In going from a point to a line, the process is the addition of another point and of a line joining the two. Similarly, the plane is built by the addition of a point, two connecting lines and a filling plane. And a solid is created by the addition of another non-co-planar point (vertex), three additional lines (edges), three planes (faces) and an infilling solid.

At the same time, it is also possible that a line, plane or solid is added to another by a process of addition, which can be called as a *type-editing* process. These procedures are added/edited by simple rules.

4.1.4. An Integrated Data Structure for Shapes

The data-structure for the development of shapes is shown in Figure 2. The structure is dynamic such that it accommodates differing elements, from points to solids by type conversion.

At the same time, it is also possible that a line, plane or solid is added to another by addition, here called *type-editing*.

The advantage of such a data structure is that a single structure type can

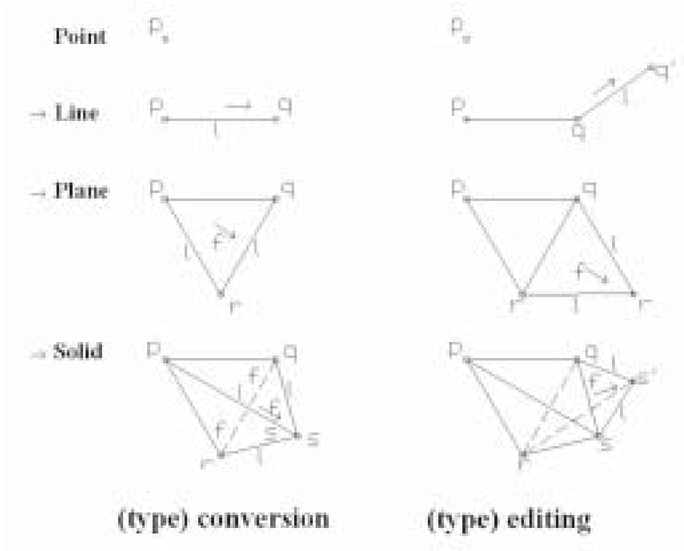

Figure 1. Development of the form

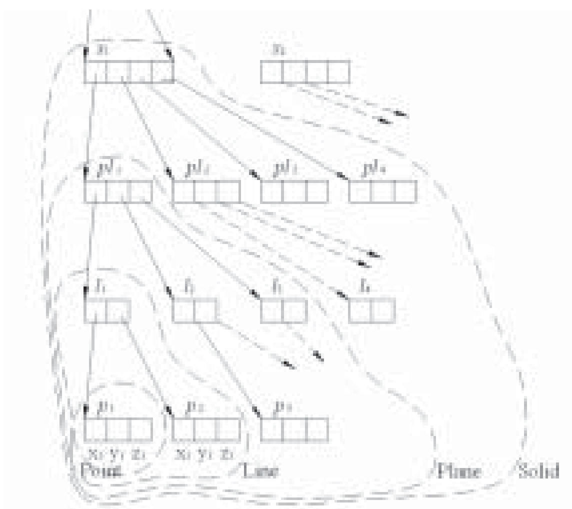

Figure 2. Schematic data-structure

accommodate the different elements using object-oriented languages. This can be expanded dynamically as and when necessary. The underlying principle is a process of triangulation that ensures that the forms are consistent. This type of structure also ensures that the elements stay independent.

5. The User Interface

As indicated in Section 4.1.2, the use of the hybrid algebra has many benefits particularly for the user-designer. By extending the range of the operations, the user has an option of several new manoeuvres as compared to the other modelling environments. The range of operations is wide that the user will have difficulty in deciding the actual choice necessary. Thus, probably an expert would be able to master the choices. This is a long way from the simple four/five operations that is possible with solid modelling systems (even today!).

It might be advisable to keep the operations within manageable numbers. In fact, it is felt that one of the concerns with the new operation table would be that with the number of operations available to the user, there might be much confusion regarding the operations. Hence, it could be advised that the operations be made available in stages, or by a customization menu that can be controlled.

5.1. DESIGNING WITH THE USER INTERFACE

An advantage that might not be obvious is that the method of operations is very intuitive and similar to the methods used by people using pencil-and-paper, or in the manner that designers work things in the head. The mental computations used by the designer become so innate and spontaneous that working with external devices seems cumbersome.

6. Conclusion

It can be seen that the adoption of shape modelling using hybrid algebras offers many advantages over conventional modellers especially for the initial design stages. It is known that conventional design system lets one represent and depict forms and shapes, but is limited in its possibilities.

The use of dynamic data-structures using object-oriented languages should lead to better ways of modelling shapes. However, the actual functioning of these has not been tested, but show promise.

It is hoped that the use of these techniques will also aid in the development of the shape modelling systems in the long run. What is mentioned is part of the overall work that will go towards the making of dedicated machines for shapes.

The main aim of these efforts is to make the design process as good as the process of designing in intuitive manners.

References

Chase, S.C. 1987, Computer implementations of shape grammars. *Master's thesis*. GSAUP, University of California Los Angeles.

Gips J. 1999, "Computer implementation of shape grammars". www.shapegrammar.org

Knight T. 2003, "Computing with ambiguity". *Environment and Planning B: Planning and Design.* vol. 30, pp. 165-180.

Mantyla M. 1988, An introduction to solid modelling, Computer Science Press, Rockville, Maryland.

Mortenson M. 1985, Geometric modelling. John Wiley, New York.

Rossignac, J.R. and Requicha, A.A.G. 1991, "Constructive non-regularized geometry", *Computer Aided Design*, vol. 23, pp. 21-32.

Stiny G. 1980, "An introduction to shape and shape grammars" *Environment and Planning B: Planning and Design.* 7: 343-351.

Stiny G. 1991,"The algebras of design" *Research in Engineering Design*, vol. 2, no. 3, pp. 171-181.

Stiny G. 2001, "How to calculate with shapes" in *Formal Engineering Design Synthesis*. Eds. E.K. Antonsson, J Cagan. Cambridge University Press, New York. pp. 20-64.

Stouffs R. 1994, The algebra of shapes, *Ph.D. Dissertation*, Department of Architecture, Carnegie Mellon University, Pittsburgh.

Varghese, P. 1993, Solids, shapes and modelling. *Master's thesis*. GSAUP, University of California Los Angeles.

Varghese, P. 2002, "Shape studies: towards the description of a shape modelling environment and partial modelling possibilities". ITbuilt: Conference held at Dept. of Architecture & Reg. Plng., Jan. 2002, Indian Institute of Technology, Kharagpur, India.