AN ONTOLOGY FOR CONCEPTUAL DESIGN IN ARCHITECTURE

SAMIR S. EMDANAT AND EMMANUEL-G. VAKALO
Doctoral Program in Architecture
College of Architecture and Urban Planning
The University of Michigan
Ann Arbor, Michigan, 48109-2069, USA
Email: emdanat@umich.edu

Abstract. This paper presents ongoing efforts to formulate an ontology for conceptual design on the basis of shape algebras. The ontology includes definitions for spatial elements such as points, lines, planes, and volumes, as well as, non-spatial elements such as material properties. The ontology is intended to facilitate sharing knowledge of shapes and their properties among independent design agents. This paper describes the formulation of the ontology and discusses some of its underlying classes, axioms, and relations.

1. Introduction

Form-making is a collaborative activity. In almost all stages of form-making, members of a design team need to share their form-making decisions, preferences, proposals, and assessments with other team members. Conceptual design is the part of form-making in which designers formulate the initial parameters for an artifact. At this stage, design ideas are set but not finalized. It can be argued that conceptual design representations should facilitate multiple interpretations of design elements, while, at the same time, allow these elements to be modified in a variety of ways.

Algebras of shape (Stiny, 1991; Stiny, 1994) represent shape elements as continuous entities because of the maximal element representation and the sub-part relation. Compared to other forms of computer-based representations, shape algebras provide a unique representation for a shape yet without pre-ordering its elements (Stiny, 1994; Chase, 1997). Shape emergence, a characteristic of the shape algebraic approach, allows a particular shape to be interpreted in a variety of ways. Therefore, shape algebras are appropriate for representing and sharing design ideas.

CAADRIA '98 : Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia.
eds. T. Sasada, S. Yamaguchi, M. Morozumi, A. Kaga, and R. Homma
April 22-24, 1998. Osaka University, Osaka, Japan. Pp. 425-434

This paper presents ongoing efforts to develop an ontology based on shape algebraic definitions that can be used for sharing design knowledge. The ontology is written in Ontolingua (Gruber, 1993), a meta-knowledge representation language. This paper introduces some of the classes, axioms, and definitions of the ontology and discusses the factors influencing its design and the selection of its representational abstractions.

2. What is an Ontology

An ontology is a formal declarative specification of concepts and relations intended for use in knowledge sharing in a particular domain (Gruber, 1993). Ontolingua, is a language for describing ontologies in a form that is interchangeable with other knowledge representation languages such as LOOM, EPIKIT, and CLIPS. The syntax of Ontolingua is based on the syntax of the Knowledge Interchange Format (KIF), which, in turn, is an extended form of first-order logic (Gensereth and Fikes, 1992). KIF is based on declarative semantics (i.e., it is not dependent on a particular computer implementation) and is logically comprehensive (i.e., it can represent any statement in first-order logic). Consequently, Ontolingua is an expressive, declarative, and system-independent language for describing ontologies. There are numerous examples of ontologies targeting a variety of domains including engineering mathematics, matrix algebra, abstract algebra, and configuration design. Such ontologies are available on the web on Stanford's KSL Network Services (http://www-ksl-svc.stanford.edu).

3. Shape Algebras

Shape algebras were first introduced to facilitate the generative specification of shapes using shape grammars (Stiny and Gips, 1972). In the literature, numerous shape grammars were formulated including grammars for generating Palladian villas (Stiny and Mitchell, 1987), Queen Anne houses (Flemming, 1987), and Wren church designs (Buelinckx, 1993). Recent work on the computability of shape algebras explored the potential of the algebraic approach for representing architectural form. Theoretical issues pertaining to shape continuity and closure resulted in defining shapes as sets of continuous elements based on the sub part relation and the maximal element representation (Stiny, 1994). The introduction of algebras of weights (Stiny, 1992) extended the definition of shapes to include visual physical, and functional properties among others.

Krishnamurti (1980; 1981) addressed the issues pertaining to the computability of shapes composed of points and lines. Later research extended this work to algebras of planes and solids (Krishnamurti, 1992; Krishnamurti and Stouffs, 1997). A number of shape grammar interpreters for architectural design were implemented with varying levels of support of shape algebraic concepts and functionality (Krishnamurti and Giraud, 1986; Heisserman, 1991; Tapia, 1996; Chase, 1996; Emdanat, 1997). Moreover, restricted forms of shape algebras were utilized to solve product optimization problems in other design-related domains including mechanical and structural engineering (Cagan and Mitchell, 1993; Brown and Cagan, 1996).

Stouffs, Krishnamurti, and Eastman (1996) discussed in detail the advantages of the algebraic framework over other approaches to shape representation. In particular, the algebraic approach is favored for its uniform, extendible, and flexible shape representation. Shape algebras do not impose a pre-ordering on shape elements, at the same time, they allow an infinite number of shapes of the same kind to be embedded within a given shape. Thus, they allow the same shape to have multiple interpretations (*Figure 1*).

A shape algebra defines shape elements such as points, lines, planes, and volumes, as well as, the algebraic relations between these elements. The relations are in the form of shape addition, subtraction, and intersection. Each shape element defines a level of shape description. Points are the simplest and solids the most complex. Shapes at higher levels are defined in terms of shape boundaries (Earl, 1997) using shapes from a lower level.

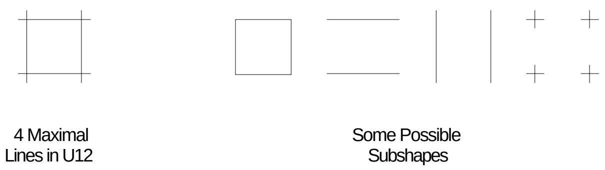


Figure 1: A Shape and Some of Its Possible Subshapes

Despite the possibilities that shape algebras offer in sharing design knowledge, implementations of shape algebras remain in the form of stand-alone systems. Sharing knowledge of shapes through shape algebras facilitates knowledge reuse. It reduces the amount of effort involved in building practical shape interpreters and the cost associated with building complex form-making systems. Moreover, a shared ontological representation facilitates the incorporation of other kinds of knowledge from other disciplines. It is anticipated that with the increasing availability of shape interpreters for a variety of form-making activities, there will be a need for these systems to interact in a shared environment to solve complex form-making problems.

4. An Ontology for Shape Algebras

Typically, for the creation of a system that uses the shape algebraic theory, a number of concepts have to be defined including spatial dimension, shape, spatial elements, spatial relations, and shape properties. In addition, other specific concepts such as the sub part relation and shape boundaries will have to be defined explicitly. In the literature, many of these issues have been explored and efficient algorithms for their definition have been proposed. In the remaining sections, this paper proposes a framework for treating these definitions as shared conceptualizations represented formally in a shape algebra ontology.

A shape is considered an element in a shape algebra that defines the properties of shapes in a spatial dimension. A spatial dimension imposes constraints on how shape primitives are defined and interpreted. *Figure 2* shows a conceptual representation of commonly used spatial dimensions in discussing architectural composition. Usually, elements of one spatial dimension are used to restrict the organization of elements of other dimensions. For example, a composition of spaces can be characterized as linear or centralized depending on whether they are organized along a line or around a point.

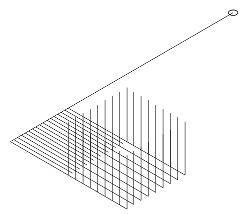


Figure 2. Possible Spatial Dimensions: Point, Line, Plane, and Volume (Baker, 1989)

A shape algebra is expressed with reference to its elements and the spatial dimension within which they are defined. A shape algebra U_{ij} indicates that its shapes are defined using elements of type i belonging to spatial dimension j. The index i is a number from the set {0,1,2,3} that corresponds to the set {point, line, plane, solid}, and the index j is a number from the set {0,1,2,3} that can be either discrete, linear, planar, or volumetric. Note that i is always less than or equal to j.

In the shape algebra ontology, spatial dimensions are defined based on the number of allowable coordinate axes. In U_{i0} no coordinate axes are allowed and only one shape (i.e., a single point) is possible. In this case, the index i must be equal to zero (i.e., representing a single point). In U_{i1}, only one coordinate axis is permitted. Therefore, collinear arrangements of points or lines are possible.

In U_{i2}, two coordinate axes are allowed. Therefore, coplanar arrangements of points, lines, and planes are possible. Finally, in U_{i3}, three coordinate axes are allowed and arrangements of points, lines, planes, and solids are possible.

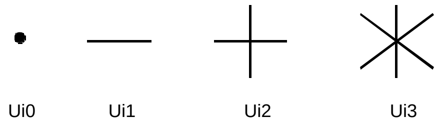


Figure 3. Possible Spatial Dimensions

This conceptualization allows for the representation of the same shapes in different spatial dimensions in a manner that is consistent with the concepts of world coordinate systems and entity coordinate systems that are commonly used in geometric modeling systems. Thus, by default, all shape entities are defined as elements in U_{33}. Appropriate coordinate system transformations map these shape entities to corresponding entities in other spatial dimensions.

To maintain continuity with other geometric and design related concepts in the KSL library, entity spaces are treated as instances of the geometry class defined in the "quantify-space" ontology. Quantity space is defined as the set of all elements that belong to that type of space and in which a distance can be measured between any pair of its elements (*Figure 4*).

```
(Defrelation Quantity-Space (?S)
  := (And (Set ?S)
          (Forall (?X1 ?X2)
                  (=> (And (Member ?X1 ?S) (Member ?X2 ?S))
                      (Exists (?D)
                              (And (= ?D (Distance ?X1 ?X2))
                                   (Scalar-Quantity ?D)))))))
```

Figure 4: The KIF Definition of a Quantity Space in the Quantity-Spaces Ontology (The KSL Ontology Server)

The shape algebra ontology introduces a special sub class of the quantity space class in case additional concepts need to be incorporated for entity spaces that are not applicable for quantity spaces. The four different spatial dimensions are defined as entity space instances as shown in *Figure 5*.

```
Entity-Space
   Entity-Space-0d
   Entity-Space-1d
   Entity-Space-2d
   Entity-Sapce-3d
```

Figure 5. Defining Entity Spaces

Axioms and relations define the kinds of elements that belong to each dimension. Since U_{i3} is the most general spatial dimension, all other entity spaces are defined as specializations thereof. Thus, elements in U_{i2} (i.e., points, lines, and planes) are defined using two-dimensional points (i.e., one of their coordinate values is restricted to zero). Similarly, elements in U_{i1} (i.e., points, lines) are defined using one-dimensional points (i.e., two of their coordinate axis are restricted to zero). Finally, the only allowable element in U_0 is the point of origin (0,0,0). For example, to state that all elements of U_{i2} are points, lines or planes, one could write using KIF notion:

```
(<=> (Entity-Space-2d ?X)
       (Forall (?E)
              (<=> (Member ?E ?X) (Shape-Entity-2d ?E))))
```

Figure 6. Defining a Two-Dimensional Entity Space

Commonly used shape entities are points, lines, planes, or solids. It is also possible to define algebras for curves, surfaces, and other kinds of shapes. To allow for the future addition of other kinds of shapes, the ontology defines the commonly used shape entities in terms of abstract shape entities (*Figure 7*). For example, lines are generalized into linear entities, which may also include rays. Two-dimensional lines are generalized to curved entities, which may also include arcs.

```
Shape-Entity
   Shape-Entity-0d
      Point-0d
   Shape-Entity-1d
      Point-1d
      Linear-Entity-1d
         Line-1d
   Shape-Entity-2d
      Point-2d
      Curved-Entity-2d
         Circular-Arc-2d
         Linear-Entity-2d
            Line-Segment-2d
      Planar-Entity-2d
         Plane-2d
      ...
   Shape-Entity-3d
      ...
```

Figure 7. Part of the Shape Entity Conceptualization

Each kind of shape entity is defined using an appropriate point type. For instance, two-dimensional shape entities are defined using two-dimensional points. In the ontology, a three-dimensional point (Point-3d) is defined as a set of elements, each of which is a member of the entity space: Entity-Space-3d. In order to indicate membership of an element to an entity space, the relation Element-In is defined. The domain of this relation is a shape entity and its range is an entity space.

```
(<=> (Point-0d ?p) (Element-In ?p Entity-Space-0d))
(<=> (Point-1d ?p) (Element-In ?p Entity-Space-1d))
(<=> (Point-2d ?p) (Element-In ?p Entity-Space-2d))
(<=> (Point-3d ?p) (Element-In ?p Entity-Space-3d))
```

Figure 8. Relating Points to Spatial Dimensions

Once the basic classes have been defined, it would be possible to define functions and relations between them. For instance, points are partially ordered according to the < relation. In the ontology, the algebraic < relation is re-written to order points in each of the spatial dimensions. In Entity-Space-2d a point is less than another, if and only if, the x coordinate of the first is less than the x coordinate of the second, or if the two are the same and the y coordinate of the first is less than the y coordinate of the second. This can be captured in the following KIF expression (*Figure 9*):

```
(< ?X ?Y)
    (And (Point-2d ?X)
       (Point-2d ?Y)
          (Or (< (X-Coord ?X) (X-Coord ?Y))
                 (And (= (X-Coord ?X) (X-Coord ?Y))
                      (< (Y-Coord ?X) (Y-Coord ?Y))))))
```

Figure 9. Defining the Point Less Than Relation

Any shape entity defines a spatial dimension that is appropriate to its kind. Lines define linear spatial dimensions, planes define planar spatial dimensions, and volumes define volumetric spatial dimensions. Any set of n non-collinear lines in U_{i3} defines n lines in n different spatial dimensions U_{11}. Similarly, two coplanar lines in this set define a planar spatial dimension in which other planar entities can be added using an appropriate shape rule. *Figure 10*-A shows four planar shapes defined in spatial dimension U_{23}. The two dark shapes taken in isolation (*Figure 10*-B) define an emergent spatial dimension U_{22} in which specific shape rules written for this space may operate. A transformation T maps elements in U_{22} to corresponding elements in U_{23}. Similarly, *Figure 10*-C shows another emergent spatial dimension U_{22} in which the other two shapes are coplanar. The advantage of this conceptualization is that it facilitates the definition of shape rules for one dimension and their reuse in other emergent spatial dimensions after adding more shape elements.

The advantage of this becomes apparent through the following example. It has been argued (Curtis, 1923; Clark and Pause, 1985) that architects use plan and section to reason about compositional qualities. In many cases, it is believed that the rules that apply in plan inform what may be used in section or elevation. By allowing emergent spatial dimensions to be recognized, the same shape rule can operate on entities in spatial dimensions that were not defined originally. In *Figure 11*, Rule 1 is defined in the spatial dimension U_{12} x U_{22} and Rule 2 is defined in U_{12} x U_{23}. After applying Rule 1 to the starting shape shown in *Figure 12*-A the resulting shape (*Figure 12*-B) is still in U_{12} x U_{22}. Given that all the shape elements defined in U_{12} and U_{22} can be mapped onto corresponding elements in other spatial dimensions, an emergent spatial dimension U_{23} allows Rule 2 to apply and generate the shape shown in *Figure 12*-C in the spatial dimension U_{12} x U_{23}.

A B C

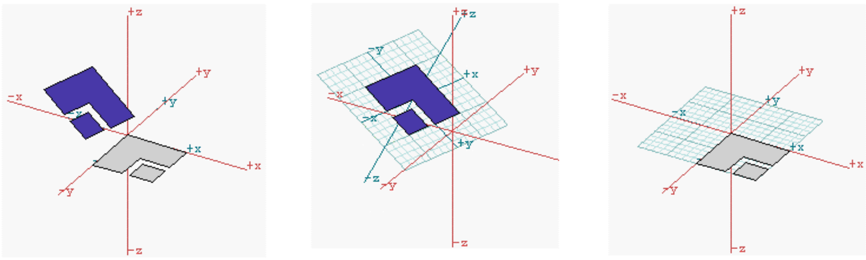
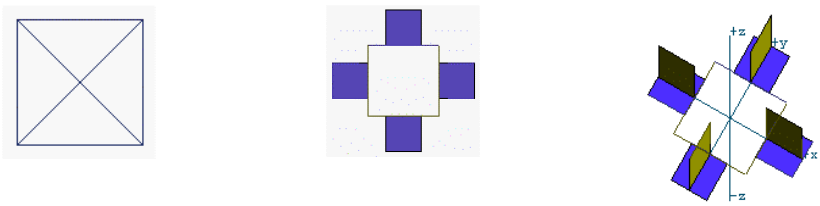


Figure 10. Spatial Dimensions

Rule 1 Rule 2

Figure 11. Example of Two Shape Rules

A B C



Initial Shape | Applying Rule 1 Four Times in U_{22}. | Applying Rule 2 Four Times in U23.

Figure 12. Example of Shape Derivation

5. Implications and Future Work

The purpose of this paper was to introduce a new way of approaching shape representation through the definition of a shared ontology of shape algebras and shape related concepts. It is anticipated that with the increasing number of shape interpreters, a need will arise for these systems to integrate their knowledge of shapes and their behavior in order to solve complex form-making problems. The shape algebraic framework is suitable for sharing design knowledge because it does not pre-order shapes, at the same time, it allows for various interpretations to emerge from the same shape description depending on the preferences of the designer. Work is being conducted on extending the initial efforts outlined in this paper to build a set of conceptualizations of shapes in shape algebras and translate these conceptualizations into a set of ontologies grouped together under the shape algebra ontology.

References

Baker, G. (1989) *Design Strategies in Architecture: An Approach to the Analysis of Form*. London: Van Nostrand Reinhold.

Brown, K. and J. Cagan (1996) “Modified Shape Annealing for Optimally-Directed Generation: Initial Results.” *Advances in Formal Design Methods for CAD: Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for Computer-Aided Design*, (eds) J. Gero and F. Sudweeks, 67-80. London: Chapman & Hall.

Buelinckx, H. (1993) “Wren's Language of City Church Designs: A Formal Generative Classification. *Environment and Planning B: Planning and Design* 20: 645-76.

Cagan, J. and W. Mitchell (1993) “Optimally Directed Shape Generation by Shape Annealing.” *Environment and Planning B: Planning and Design* 20: 5-12.

Chase, S. (1997) “Logic-Based Design Modeling with Shape Algebras.” *Automation in Construction* 6, no. 4: 311-22.

Chase, S. (1996) *Modeling Designs with Shape Algebras and Formal Logic.* Ph.D. Dissertation, University of California.

Clark, R. and M. Pause (1985) *Precedents in Architecture*. New York: Van Nostrand Reinhold.

Curtis, N. (1923) *Architectural Composition*. Cleveland: J. H. Jansen.

Earl, C. (1997) “Shape Boundaries.” *Environment and Planning B: Planning and Design* 24: 669-87.

Emdanat, S. (1997) "Shape Grammars, Some Basic Implementations and New Results: Introduction to Building Grammars." College of Architecture and Urban Planning, University of Michigan, Ann Arbor, MI, Unpublished.

Flemming, U. (1987) “More Than the Sum of Parts: The Grammar of Queen Anne Houses.” *Environment and Planning B: Planning and Design* 14: 323-50.

Gensereth, M. and R. Fikes (1992) “Knowledge Interchange Format, Version 2.0” *Reference Manual, Logic Group*: Logic-91-1. Computer Science Department, Stanford University, Stanford, California.

Gruber, T. (1993) “Translation Approach to Portable Ontology Specifications.” *Knowledge Acquisition* 5, no. 2: 199-220.

Heisserman, J. (1991) *Generative Geometric Design and Boundary Solid Grammars Department of Architecture.* Ph.D. Dissertation, Carnegie Mellon University.

Krishnamurti, R. (1980) “The Arithmetic of Shapes.” *Environment and Planning B: Planning and Design* 7: 463-84.
Krishnamurti, R. (1981) “The Construction of Shapes.” *Environment and Planning B: Planning and Design* 8: 5-40.
Krishnamurti, R. (1992) “The Maximal Representation of a Shape.” *Environment and Planning B: Planning and Design* 19: 267-88.
Krishnamurti, R. and C. Giraud (1986) “Towards a Shape Editor: The Implementation of a Shape Generation System.” *Environment and Planning B: Planning and Design* 13: 391-404.
Krishnamurti, R. and R. Stouffs (1997) “Spatial Change: Continuity, Reversibility, and Emergent Shapes.” *Environment and Planning B: Planning and Design* 24: 359-84.
Stiny, G. (1991) “The Algebras of Design.” *Research in Engineering Design* 2: 171-81.
Stiny, G. (1994) “Shape Rules: Closure, Continuity, and Emergence.” *Environment and Planning B: Planning and Design* 21: s49-s78.
Stiny, G. (1992) “Weights.” *Environment and Planning B: Planning and Design* 19: 413-30.
Stiny, G. and J. Gips (1972) “Shape Grammars and the Generative Specification of Painting and Sculpture.” *Information Processing*. (eds.) C. Frieman, 1460-1465. North-Holland, Amsterdam: North-Holland Publishing Company.
Stiny, G. and W. Mitchell (1987) “The Palladian Grammar.” *Environment and Planning B: Planning and Design* 5: 5-18.
Stouffs, R., R. Krishnamurti, and C. Eastman (1996) “A Formal Structure for Nonequivalent Solid Representations.” *Proceedings of IFIP WG 5.2: Workshop on Knowledge Intensive CAD II*, (eds.) S. Finger, M. Maumlntylä, and T. Tomiyama. International Federation for Information Processing.
Tapia, M. (1996) *From Shape to Style, Shape Grammars: Issues in Representation and Computation, Presentation and Selection.* Ph.D. Dissertation, University of Toronto.