Accepted to *3rd International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Spa, Belgium, 18-21 August 1996
NOT FOR QUOTATION

The Extensibility and Applicability of Geometric Representations

Rudi Stouffs
Swiss Federal Institute of Technology Zurich
Architecture and CAAD
Zurich
Switzerland

Ramesh Krishnamurti
Carnegie Mellon University
Department of Architecture
Pittsburgh, Pa.
USA

Abstract

As designers pose new questions, within the context of computational design, that go beyond geometry and require other information to be included, there is, now more than ever, a need for extensible geometric representations. We believe that such can best be achieved using abstract data types defined over a set of basic operations, common to all data types. At the same time, we must consider the applicability of such a representation with respect to the functionalities of the application. In this paper, we explore these issues of extensibility and applicability as these relate to the questions of standardization and adaptability of representations. As a particular example, we consider an algebraic model, with a corresponding representation, that defines arithmetic operations that operate uniformly and consistently on geometries and on non-geometric attributes of various kinds.

1 Introduction

Design relies on effective models of geometry. Computational design of physical artifacts relies on solid models. Over the past twenty years, solid modeling has established itself as a major area of research. Several models and representations have been developed, based on a variety of representations, associated operations and underlying concepts. None, however, exists singly as a model of representation that applies to the solution of every problem. In the search for an appropriate solid model there is a consensus, namely, a solid model has to be complete. This means that the corresponding representation must be "adequate for answering arbitrary geometric questions algorithmically" (Mäntylä 1988). However, this statement becomes increasingly difficult to qualify as users and, in particular, designers pose new questions that go beyond geometry and require other information to be included.

An interesting example is the ever increasing concern for performance issues in building or engineering design, where it is vital that extended representations and manipulations of design geometries are developed. These must include form and material properties, specified at various levels, in a hierarchical manner with dynamic relationships. As an example, a wall may be composed of a number of material layers, each with fundamental thermal-performance related attributes like conductivity, thickness, specific heat, etc. The wall, as a composite entity, has thermal attributes like overall conductance, and, at a higher level, the building has an overall heat transfer value.

In the design of a multi-purpose geometric modeling system, it is generally infeasible to foresee each and every attribute the designer may want to employ. One approach to this problem is to consider abstract data types defined over a set of basic operations, common to all data types. Then, any user-defined attribute can be added if it fits within one of these types. If the basic operations are straightforward, then, these data types may themselves be user-definable. However, this set of basic operations must support the functionality of the application. To determine this qualification is straightforward for simple applications. In the case of computer-aided design (CAD) systems, on the other hand, determining all operations that both underlie the functionality of the system and apply to these data types, is a non-trivial task (Schindewolf 1995). In general, these operations become far too many and often depend on the particular data or their semantics. In order for such a scheme to be successful, it is necessary to conceive of only a limited set of operations that apply to all data types, even if in a different fashion, and suffice to model the functionality of the system.

In this paper, we consider these issues and their relative importance with respect to the efficacy of geometric representations. We consider the *extensibility* of a representation as its ability to allow for new, user-defined attributes and data types. We consider the *applicability* of a representation, with respect to the functionalities of the application, to denote its ability to provide these functionalities through a common interface of operations and queries over the data types of the representation. These issues of extensibility and applicability have received little research attention in comparison to other issues that also relate to the efficacy of a representation, such as efficiency and robustness. Furthermore, research in these latter issues has mostly led to solutions that are generally applicable to many representations. In contrast, we argue that the issues of extensibility and applicability are strongly depended on the particular representation. We explore these issues through a particular example. We consider an algebraic model, with corresponding representation, that offers a uniform and consistent approach for handling mixed-dimensional geometries and non-geometric elements and attributes.

2 Motivation

CAD systems provide prime examples of the development to include non-geometric information. Generally, these take a database approach to the problem, creating record structures for geometric entities that allow for the inclusion of non-spatial attributes, e.g., layer information, color, texture, and so on. In principle, the record structures present a static view of the world, where attributes are defined upon construction of the database, or, software. While software users often are given the ability to extend these record structures with one's own user-defined attributes, this alone does not allow for a dynamic, user-centered view of the world to be presented by the system. Beyond the establishment of these information holders – it is straightforward to attach attribute information to design objects through entries in the record structure – current systems provide no further support for even this user-modified view. Once an attribute is assigned to a design object, this information is fixed and remains unaltered under any modeling operations, unless through explicit and additional user action.

Furthermore, we often want access to this design information from outside the particular application. In practice, a translation process provides third-party access to the information contained in the database. Generally, a translation between applications requires a two-step process, where the source application provides the information in a formatted file that can, subsequently, be read and interpreted by the destination application into its own data representation. The adopted file format can be considered expressing a more generalized representational model than the system's internal representational structure. For enhanced efficiency and in order to remove any ambiguity that otherwise may arise, the information in the file is kept to a minimum: all general information is absent and instead is referenced. This results in a compact file, but, in the case of user-defined attributes, in a cryptic data representation. Therefore, when users decide to define their own attributes, they forfeit any standardization, as third parties are no longer able to interpret the translated data properly unless explicitly notified of the specific attribute additions. An example is provided by AutoCAD™'s DXF drawing interchange file format (Autodesk 1992), a popular data exchange format for visual and design applications. Attributes are assigned identification numbers and a few numbers are reserved for user-defined attributes. The DXF file contains only the identification number and value for each attribute – it does not contain any information that may allow for a proper interpretation of this data.

In general, the problem is one of extensibility while maintaining some form of standardization. In the extreme, this concerns the design of an all-encompassing standard. It is obvious that the latter is practically infeasible, perhaps even impossible. By definition, a standard is a formalism that specifies general data classes and ways of manipulating this data. Whatever representation is defined, one can always find a piece of data that does not fit within the formalism. Nonetheless, when we view the problem in terms of extensions to existing data representations for dealing with user-designed attributes, a solution may prevail. By pushing the boundaries of such a solution, we may even come close to – without actually reaching – a solution for the extreme case.

It suffices to consider a standard approach for manipulating non-standard information. Common data types can be recognized, and common operations that apply to them defined for each type. When the data to be represented fits one of these data types, the corresponding attribute(s) can be defined without any risk of loss of data integrity. When the same information, that is, the data types with their corresponding operations, is made available to other software, these will also be able to represent and manipulate the same data, even if the attributes' semantics are unknown to the software.

In order to substantiate this claim, it is important that we make a distinction between *manipulative* operations that may change the value of an attribute, and *evaluative* operations that interpret the attribute but keep its value unchanged. The former can often be defined syntactically, that is, the result of an operation is dependent only on the attribute's value and type. Evaluative operations mostly depend on the attribute's semantics. For instance, the visualization of the attribute data, e.g., color or line thickness, is attribute specific; a presentation of the attribute's name and (alphanumeric) value is not. Data integrity only relates to the correct application of manipulative operations. Therefore, we consider only these operations. That is, we consider a formalism or representation to be successful when the data of one program can be read, manipulated and rewritten by another program, even if it may not be able to visualize the data, or otherwise evaluate it. An evaluation of this new data may then be provided for by the original program.

As an example, we consider an algebraic model that defines arithmetic operations on both geometries and non-geometric attributes of various kinds. This model presents a uniform and consistent approach for dealing with spatial objects including non-spatial elements and attributes, as well as geometries of mixed-dimensionality. We believe that this uniform approach provides the key for achieving an extensible yet standardized data representation for geometries and non-geometric attributes. At the same time, this model allows for powerful new ways of manipulating objects in the context of design exploration (Stouffs and Krishnamurti 1994). Its uniform and consistent application to geometries of different dimensionality has been proven (Stouffs 1994); its application to attribute-weighted geometries has been explored for certain attribute types (Stiny 1980, 1992; Knight 1989).

3 Extensibility and Applicability

Requicha (1980) presents a list of properties or characteristics for the evaluation and comparison of geometric representations (see also Mäntylä 1988). Some are formal properties, while others influence the design and implementation of a modeling or design environment in a practical way. Formal properties include the scope or expressive power of a representation, the closure of operations, the uniqueness, unambiguity and validity of the representations. It is straightforward to assess these properties for a given representation. As such, these serve as a primary basis for the qualification and comparison of geometric representations.

Informal properties are generally harder to qualify or assess. These relate to issues of conciseness of the representation, ease of creation, and efficacy of the algorithms. The conciseness of a representation is often constrained by the efficacy of its algorithms because efficacy in this instance is related to the amount of information that is available, its ready accessibility, and its organization. The ease of creation of representational instances or model objects is mainly related to the design and objective of the modeling environment and its interface, than to the geometric representation and its algorithms. For instance, a solid modeling environment may adopt a constructive modeling approach even though the representation is a boundary representation. Therefore, the issue of efficacy is often the most important.

This representational efficacy relies on a combination of different properties that relate to the algorithms for creating, manipulating and transforming representations, such as correctness, efficiency, robustness and extensibility (Requicha 1980). The issue of correctness is straightforward but dependent on precise specifications. The issue of efficiency is prominent and has received much attention within the field of computational geometry. Generally, the efficiency for a boundary representation is less dependent on the particular representation than on the type of algorithms used. For example, algorithms for the algebraic model for geometries (Stouffs 1994) require little topological input in comparison to most other boundary representations that are often topologically complete. Even then, these can be specified with efficiencies comparable to some of the most efficient boundary representations (Stouffs 1994). The issue of robustness has also received fair attention (Hoffmann 1989). In contrast, the issue of extensibility, while it relates directly to the problem of standardization, has received little attention in such research. Standardization projects are generally guided by other issues that receive their prominence because of external interests.

Another issue, and one that is mostly overlooked, is the applicability of a representation with respect to certain functionalities. This is basically an issue of scope, though in terms of the applicability of the algorithms describing the operations on the representation to common practices and functionalities in computational design environments or CAD systems. We denote this, in short, as applicability, even though we note that such applicability is dependent on the functionality of the environment that the representation is used in. The issue of applicability is as much related to the efficacy of the representation as the issues above, but differs in the fact that it also introduces aspects of specific functionality. Whereas most other issues have received much attention and research, and have resulted in solutions that are often generally applicable to many representations, the issues of extensibility and applicability have received little research and are, we argue, very dependent on the particular representation. These issues relate directly to the questions of standardization and adaptability of representations. As such, a commitment to these issues will not only facilitate the work of developers, but also extend software capabilities and user choices.

Instead of attempting an evaluation or comparison of common representations, we examine a particular representational model, i.e., the algebraic model, that we believe is particularly suited to illustrate the importance and strength of both issues. Below, we present the model and its corresponding representation, explore and describe the extensibility and applicability of this representation in the context of computational design environments.

4 An Example

For our purpose, the 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 design objects by associating symbols, labels or properties with the corresponding geometries. Such augmented geometries allow for the exploration of new design problems and may improve upon the methodology for solving known problems. For example, highlighting points through the use of labels (i.e., labeled points) plays an important role in both spatial representation and manipulation. In the context of spatial manipulation through rules, these serve to guide the rule matching process through identification and classification of the rules, while these can also be viewed as semantic extensions to what are fundamentally syntactical expressions, i.e., rules.

The applicability of this extensible model to geometric modeling and computational design relies on the existence of arithmetic operations that operate uniformly and consistently on geometries and non-spatial attributes. We argue that these arithmetic operations, together with the ability to augment geometries with non-geometric attributes, suffice to support the functionality of a CAD system or a computational design environment. We explore the application of the arithmetic operations on augmented geometries to common design interactions, such as creation, selection, and grouping or layering of design objects. First, we present the algebraic model and consider the supporting *maximal element* representation for augmented geometries (Krishnamurti 1992; Stouffs 1994).

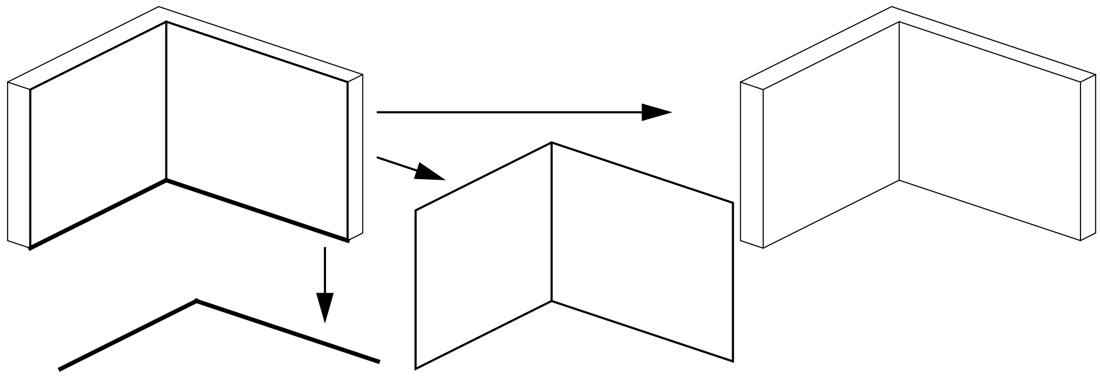
4.1 Extensibility and the Algebraic Model

We propose an arithmetic model with operations of sum, difference and product on both geometries and non-spatial attributes, because the conceptual simplicity of these arithmetic operations makes an applicative extension to new data types straightforward. We limit the arithmetic model to an algebraic model based on a part relation. This part relation can be freely defined as long as it constitutes a partial order relation. For practical purposes, it is advantageous to define the part relation between elements of the same *sort*, i.e., geometries of the same dimensionality or attributes of the same type. We consider sorts as collections of design elements of a same type, where the boundary of each sort is given by the applicative boundary of the part relation that is defined over this sort. Thus, while a minimal classification of sorts may be specified from an algebraic and representational point of view, the final decision on a more in-depth classification may lie with the user or designer, allowing her to define attribute sorts corresponding to the requirements or intentions of the design.

4.1.1 *Sorts of Geometries*

An important aspect of computational design representation is the ability to represent geometries of different dimensionalities. Consider the example of building design,

Figure 1: **A wall represented at once as a volume, planes and lines.**



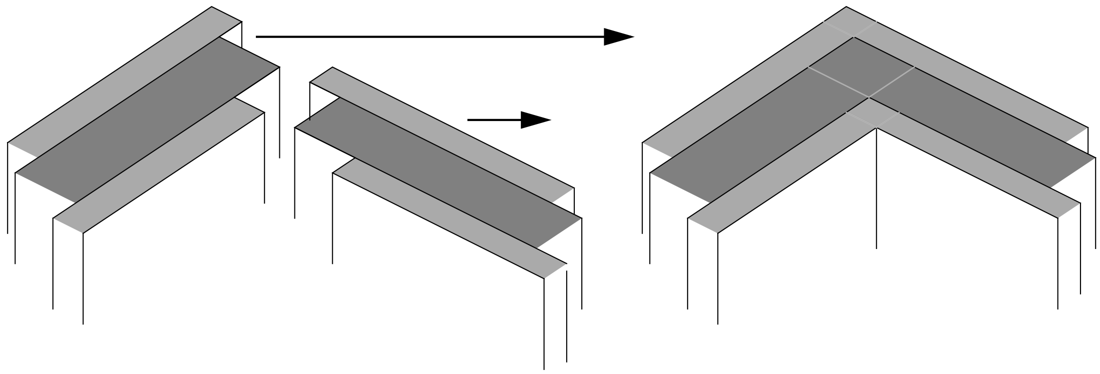
where the final product is a composition of purely solid elements or building components. Yet, in the design and evaluation process, the dimensionality of each individual component is not essential for all purposes. Instead, an abstraction is often more valuable. In the structural evaluation of a design, for example, walls can be represented as planes with simple integral attributes - a true three-dimensional model would be too complicated, unless approximated. In general, even a single component may be multiply represented as elements of different dimensionalities, each element projecting information for a specific application, such as structural or performance evaluations in building design. (figure 1)

Mixed-dimensional models have found recent support in solid modeling (Rossignac and Requicha 1991; Gursoz et al. 1991). However, these mixed-dimensional approaches come at the expense of intuition in operations and ease of conception, by adhering to Boolean Set operations in a Euclidean or point space. Instead, the algebraic approach provides an altogether new and different notion: a composite geometry. Under the algebraic model, geometries of different dimensionality coexist without interference; the components of a mixed-dimensional object are operated upon in parallel by the algebraic operations. This parallel construction allows for objects of varying disparity not only to coexist peacefully in a single (geometric modeling) world, but also to be conceived as one at the same time as being operated on quite differently, yet under a conceptually unified approach. Consequently, a composite geometry can be regarded either as composed of geometric components with different dimensionality or as consisting of multiple geometries that are coordinated or related. For example, consider a set of drawings from amongst plans, elevations and sections of a same building. Each drawing may be considered a spatial object, or, the set of drawings as a whole can be regarded as a composite object - its components may be visualized in the same, as in the case of a composite drawing, or in different spaces.

4.1.2 *Attribute Sorts*

Even when considering geometries of the same dimensionality, it may be important for these to coexist and be operated upon in parallel. Consider, for example, a

Figure 2: **A solid composition of walls as layers with different material properties.**



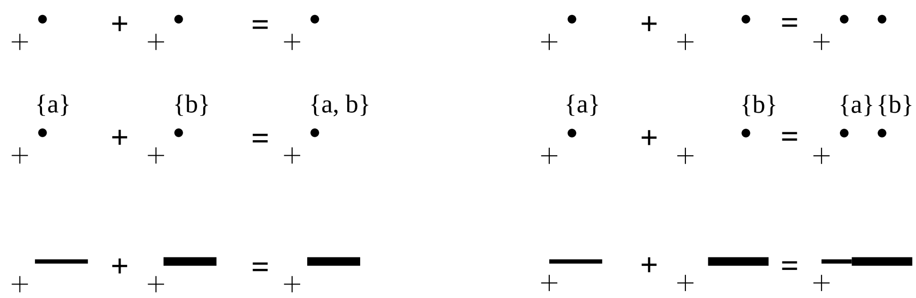
composite material composed of several regions with different material properties (Rossignac and Requicha 1991), e.g., a wall composed of a number of material layers. Such a decomposition into layers should not only be representable, but also preserved during manipulation. Commonly, this is not the case. In this respect, the algebraic model offers a straightforward solution: using non-geometric attributes to denote such properties as material, color, thermal-performance, etc. Here, different regions or layers are represented by elements with different-valued (material) attributes. The algebraic operations preserve this structure while operating simultaneously on all layers. These operations no longer obliterate the internal boundaries between these different regions. (figure 2)

Attribute sorts may be conceived for many different applications and functionalities. It is straightforward to consider attributes for the representation of graphical information, e.g., line thicknesses and colors, some design knowledge, e.g., sets of labels, and design evaluation data, e.g., numerical as well as non-numerical properties. The same or similar attribute sorts can also be used for other purposes. For example, label attributes also provide for some CAD functionality, such as grouping and layering. At the same time, we can also envision more advanced practices within the same conceptual framework. For example, one can assign functions to geometries for the deferred evaluation of physical aspects such as lighting, heating and cooling. In general, we can consider attribute sorts for any data on which we can define a partial or total ordering relation.

4.1.3 *Sorts as Algebras*

This part relation defines the set of all values for each sort as a partially ordered set or lattice with the consequent algebraic properties. As such, a sort defines an algebra that is ordered by a part relation and closed under the operations of sum, product and difference, and the similarity transformations (i.e., translation, rotation, reflection and scale) for sorts of geometries. Although, strictly speaking, the arithmetic operations can be arbitrarily (within limits) defined on the attribute values, a definition based on a part relation is most intuitive. As Stiny (1992) illustrates with line thicknesses, a single line drawn multiple times, every time with different thickness, appears as it

Figure 3: **Exemplar applications of the operation of sum on augmented geometries.**



was drawn once with the largest thickness, even though it assumes the same line with other thicknesses. If we consider the thickness of a line as a numeric value, the corresponding operation of sum would not be equivalent to the arithmetic sum operation, instead, would be defined as the least upper bound of the set of values, as specified by the ordering relation. The same reasoning holds for many other attribute sorts. On sets (of labels), obviously, we consider the set operations of union, intersection and difference as algebraic operations. For colors, a ranking may be specified that maps the colors to real values or, otherwise, a three-dimensional color coding (e.g., RGB or intensity, saturation and hue) may be considered with an appropriate part relation. Another example are functional attributes where the attribute's value is a function defined as an element of some functional space.

We adopt a maximal element representation for augmented geometries that supports this algebraic behavior. This canonical representation defines a (maximal) geometry as a set of disjoint spatial elements, each of which is connected. These disjoint elements are the *maximal elements* of the geometry. For instance, consider the algebra of points. A point is maximal if it is not coincident with any other point. Thus, the sum of two points is the set of both points if these are not coincident and is the single point otherwise (figure 3). The operation of sum combines two geometries, that are not necessarily maximal with respect to each other, into a single maximal geometry – this holds for geometries of all dimensionalities. Similarly, the results of the operations of product and difference are maximal if the operands are maximal. When considering geometries augmented with attributes, we associate with each spatial element a second element that is a value taken from the attribute sort. The result is a pair composed of a geometry and an attribute. Such pairs may be viewed as elements of a new, augmented, algebra.

For example, consider labels as attributes. Then, we associate with each spatial element a label, or, a set of labels if we allow a single spatial element to have more than one label. We can consider algebraic operations on labeled geometries similar to these operations on simple geometries. Similar to the sum of two points, the sum of two labeled points equals the set of both labeled points if the points are not coincident. If these are coincident, the sum equals the single point with the sum, or set union, of both sets of labels (figure 3). The operations of product and difference apply in a similar way to labeled points, using the set operations of intersection and difference on sets of labels. Then, a labeled point is a part of another labeled point, if the points are coincident and the labels to the first point form a subset of the labels to the second point. This definition is intuitive. Similar definitions hold for augmented geometries of higher dimensionalities, except that when combining overlapping geometries with different attributes, the result is depended upon three parts: the difference of the first geometry with the second, the difference of the second geometry with the first, and the common part of both geometries. The attribute value of the common part is derived in the same way as for points. The other two parts receive their attribute value from the original geometry each is a part of (in the case of sum) (figure 3).

4.1.4 *Extensibility*
There exists indeterminately many possible and user-preferable attribute sorts. However, many of these can be expressed, syntactically, in an identical manner. That is, while their meaning may be different, their representational form may be identical, and the operations of sum, product and difference may be defined identically on this form. As an example of the latter, the operation of sum is often defined as the least upper bound of the set of operand values. Thus, we can envision a number of basic data types, e.g., numeric values and sets of labels, with corresponding arithmetic operations defined over each data type. These data types can serve as frameworks for different attribute sorts. If appropriate, different versions of the same data type can be envisioned with the arithmetic operations defined differently, in order to cope with a larger variety of different sorts. Then, when the user conceives a new attribute sort, the values of which fit into one of these data types, and the corresponding arithmetic operations operate as expected and preferred, this sort may be defined as an instance of the corresponding data type. We may also give users the opportunity to define their own data representation and/or arithmetic operations on a data representation, thereby creating a new data type that, subsequently, can be used as the framework for new attribute sorts.

4.2 Applicability and Design Interactions

We argue that the algebraic model for geometries augmented with attributes may be sufficient to support the functionality of most CAD systems. We consider the algebraic operations of sum, product and difference on augmented geometries together with the similarity transformations on geometries, to express this functionality. We also consider the distinguishing of sorts and compositions of sorts as algebras, thereby specifying a parallel but uniform application of the operations on elements of different algebras, for this purpose. Below, we explore this application to common computational design interactions and CAD functionalities.

4.2.1 *Creation and Selection*
Consider the most common operations of creation and deletion, selection and deselection of objects: A user may create a mental model consisting of two worlds,

one is the design, the other the selection. Selecting an object means moving it from the design world to the selection world, deselecting an object does the opposite. In both cases, an object is first removed from one world and subsequently inserted into the other. If we consider two algebras, one for the design and another for the selection, the operations can be translated into a difference in one algebra followed by a sum in the other. The operations of creation and deletion work in the same way, except for the fact that there is only one algebra involved, the other world is the empty space.

This similarity between the operations of creation and deletion, on one hand, selection and deselection, on the other hand, is another strong advantage of the algebraic model. Fundamental to the algebraic model is that, under the part relation, every element of a sort specifies an indefinite set of elements that are each part of the original element. Applied to geometries, it means that *any* part of a spatial object is an object and can be manipulated as such; thus, users can deal with objects in indeterminate ways. This is quite distinct from the selection process in conventional CAD approaches where the only objects that are selectable correspond to those prescribed minimal entities that have been predefined in the data-structures. Under the algebraic model, in contrast, the part relation for geometries leads naturally to the concept of *emergent geometries*: spatial objects that are not a priori defined, but emerge under the part relation. Emergent geometries play an important role in design search and exploration (Mitchell 1993; Stiny 1993). Recent research suggests that phenomena of emergence can be used to explain, on the basis of continuity and articulate consistency, why descriptions of design are post-rationalized in that descriptions of design explain the precedents that justify the designs (Stiny 1994; Krishnamurti and Stouffs forthcoming). This fits and leads into issues of cognitive psychology, though not explored in this paper, are essential to an understanding of human (user) behavior in connection with design and CAD environments.

Since, in CAD systems, the objects of manipulation are a priori defined in the data-structures, the selection of an object can be achieved by a straightforward search in the database, following a lead by the user, e.g., the position of the cursor at the moment of the activation of the search. The algebraic model does not impede such searches. However, since we consider an object as a definite description of indefinitely many parts, the action of selecting a part may include some more powerful means of describing what is to be selected. Since any part can be selected, in the extreme this may require the user to create the selection as an object that is a part of the design. Therefore, the operations of creation and selection may invoke the same action sequence in terms of describing the resulting object, whether it is a newly created object or a selected existing part. Thus, these operations may be presented to the user in a similar way as to emphasize the common dialogue in achieving these results, even if the results themselves are conceptually quite different.

4.2.2 *Layering and Grouping*

We consider another common aspect of CAD systems: layering of the data. Layering allows the user to introduce a classification of the data in terms of user-defined concepts. Such a classification enables the user to hide or exclude data temporarily from the design and facilitates the searching of entities. The layering can be viewed as a differentiation between algebras: geometries only combine within the same algebra, algebras can be included or excluded in the visualization or editing, etc. Or these can be viewed as attributes such as labels. The result is slightly different. When considering layers as attribute values, it may seem intuitive to allow only for a single label (or layer) for each element. After all, most CAD systems do not allow a same object to exist in multiple layers at the same time. However, there is no notion of a layer ordering that can be used to define the algebraic operations on layer labels. If we want to restrict a spatial element to a single layer, considering different algebras for the different layers makes most sense.

Next, consider what would happen if we allow a set of layer labels to be assigned to a geometry. This may be interpreted as identical copies of the same geometry existing in different layers, and can be represented as such to the user. In this case, the result to the user is identical as in the algebras version. Or, the geometry can be visualized to the user as belonging to multiple layers at the same time. In order to manipulate a single-layer copy, that is, a copy of the geometry with only a single label (or a singleton set of labels) assigned to it, we need to differentiate it from the other copies of the same geometry existing in the other layers. Under the algebraic model any part of a geometry is a geometry and can and ought to be manipulated upon as such. A single-layer copy is a part of a multi-layer geometry if the single label to the first element is a member of the set of labels to the second element. Selecting it includes taking the difference of the first element from the second. As such, the other copies remain as a single element of the algebra, but the single-layer copy is no longer a part of this geometry.

Other grouping operations can be achieved in a similar way. Note that selection is also a form of grouping. Algebras and attributes can also be used to relate or coordinate different elements into a single entity. For example, when different views of a same object exist in different layers, e.g., a two-dimensional and three-dimensional version for the sake of visualizing 2D and 3D views, these may be coordinated so that selecting all views of a single object may be facilitated.

5 Conclusion

The issues of extensibility and applicability for geometric representations relate to the efficacy of these representations. In comparison to other aspects of efficacy, these issues have received little attention so far. However, we believe that, now more than ever, a commitment to address these issues is needed in order to support the quest for adaptable computational design environments. At the same time, the adoption of a uniform approach for handling distinct data sorts can ease the work of developers, giving them the opportunity to provide support for new and extended computational design functionalities. The algebraic model presented here is particularly supportive to achieve such. As an additional example, the same model may provide for a powerful query language that allows a designer to retrieve design information that may not be explicitly available or provided (Stouffs and Krishnamurti 1996a). The support of the algebraic model for adaptability in knowledge intensive design and engineering is the focus of Stouffs and Krishnamurti (1996b).

The algebraic model and the maximal element representation compare favorably with other geometric representation schemes on most comparative issues: the model extends to mixed-dimensional shapes, the algebraic operations are closed within the algebras defined and the asymptotic running times of the algorithms for the description and manipulation of shapes are comparable to similar algorithms for many geometric representations (Stouffs 1994). For example, Stouffs et al. (1995; also Stouffs and Krishnamurti forthcoming) compare its scope with other boundary representations to illustrate the issues of translation between representations. However, on its own, this comparison only places the maximal element representation on a par with established representations and does not give the merit a new representation scheme requires to become a prominent contender. On the other hand, we believe its value to the issues of extensibility and applicability may achieve this.

6 Acknowledgment

We would like to thank David Kurmann for his comments on an earlier version of this paper.

7 References

Autodesk. (1992) Drawing interchange and file formats, Chapter in *AutoCAD Release 12 Customization Manual*. Autodesk, Sausalito, Calif., pp. 243-284.

Gursoz, E.L., Y. Choi and F.B. Prinz (1991) Boolean set operations on non-manifold boundary representation objects, *Computer Aided Design* **23**, pp. 33-39.

Hoffmann, C.M. (1989) The problems of accuracy and robustness in geometric computation, *Computer* **22**, pp. 31-39, 41.

Knight, T.W. (1989) Color grammars: designing with lines and colors, *Environment and Planning B: Planning and Design* **16**, pp. 417-449.

Krishnamurti, R. (1992) The maximal representation of a shape, *Environment and Planning B: Planning and Design* **19**, pp. 267-288.

Krishnamurti, R. and R. Stouffs (forthcoming) Spatial change: continuity, reversibility and emergent shapes, To appear in *Environment and Planning B: Planning and Design*.

Mäntylä, M. (1988) *An Introduction to Solid Modeling*. Computer Science Press, Rockville, Md.

Mitchell, W.J. (1993) A computational view of design creativity, in Gero, J.S. and Maher, M.L. (eds.), *Modeling Creativity and Knowledge-Based Creative Design*. Lawrence Erlbaum, Hillsdale, N.J., pp. 25-42.

Requicha, A.A.G. (1980) Representations for rigid solids: theory, methods and systems, *Computing Surveys* **12**, pp. 437-464.

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

Schindewolf, M.W. (1995) (CADKEY Inc.) Personal conversation with the first author.

Stiny, G. (1980) Introduction to shape and shape grammars, *Environment and Planning B: Planning and Design* **7**, pp. 343-351.

Stiny, G. (1992) Weights, *Environment and Planning B: Planning and Design* **19**, pp. 413-430.

Stiny, G. (1993) Emergence and continuity in shape grammars, in Flemming, U. and Van Wyk, S. (eds.), *CAAD Futures '93*. North-Holland, Amsterdam, pp. 37-54.

Stiny, G. (1994) Shape rules: closure, continuity and emergence, *Environment and Planning B: Planning and Design* **21**, pp. s49-s78.

Stouffs, R. (1994) The Algebra of Shapes, Ph.D. Dissertation, Department of Architecture, Carnegie Mellon University, Pittsburgh, Pa.

Stouffs, R. and R. Krishnamurti (1994) An algebraic approach to shape computation (a position paper), in Workshop notes of *Reasoning with Shapes in Design. Artificial Intelligence in Design '94*, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, pp. 50-55.

Stouffs, R. and R. Krishnamurti (1996a) On a query language for weighted geometries, In *Third Canadian Conference on Computing in Civil and Building Engineering*, Montreal, 26-28 August 1996.

Stouffs, R. and R. Krishnamurti (1996b) An architecture for adaptability for knowledge intensive design and engineering, Submitted to *Knowledge Intensive CAD Second Workshop*, Pittsburgh, Pa., 16-18 September 1996.

Stouffs, R. and R. Krishnamurti (forthcoming) An algebraic approach to comparing representations, Manuscript, Architecture and CAAD, Swiss Federal Institute of Technology, Zurich and Dept. of Architecture, Carnegie Mellon University, Pittsburgh, Pa.

Stouffs, R., R. Krishnamurti, C.M. Eastman and H. Assal (1995) Non-"standard" solid representations, Technical Report, Dept. of Architecture, Carnegie Mellon University, Pittsburgh, Pa. and School of Architecture and Urban Planning, University of California, Los Angeles.