Topological Framework for Part Families

Srinivas Raghothama
EDS PLM Solutions,
10824 Hope Street,
Cypress, CA 90630
e-mail: raghotha@ugs.com

Vadim Shapiro
University of Wisconsin,
1513 University Avenue,
Madison, WI 53706
e-mail: vshapiro@engr.wisc.edu

One of the fundamental unsolved problems in geometric design of mechanical solids has been the lack of a proper notion of family or class. Numerous heuristic and often incompatible definitions are used throughout the CAD industry, and it is usually not clear how to generate members of a family or, to decide if a given object belongs to an assumed family. Until these difficulties are resolved, no guarantees or standards for parametric modeling are possible, and all efforts to allow exchange of parametric representations between different CAD systems are likely to remain futile. Standardizing on a particular definition may be difficult, because parametric families depend intrinsically not only on shape but also on its representation. We classify families into parameter-space and representation-space, and show that both types are representation-induced families. We propose a formal framework for families based on the notion of topological categories. Every parametric family is defined by the representation-induced topological space of solids that are closed under the continuous maps in the assumed topology. We illustrate several well defined families and formally define a special but important case of CSG-induced family that generalizes to the more general case of feature-induced families.
[DOI: 10.1115/1.1558073]

1 Introduction

1.1 Difficulties with Families. Mechanical parts are modeled as solids that are defined precisely in terms of their mathematical models and representations. The mathematical models assure the validity and other desirable properties of the solids and their representations. The more recent paradigm of modeling solid families has gained importance,[1] but it also introduced numerous difficulties that are well documented in literature [2,3]. Broadly, these difficulties may summarized as: (1) unpredictability (because of ambiguity), (2) inconsistency (within the same system and different systems), (3) dependence on representation scheme, parameterizations, and implementation, (4) lack of common standards.

Consequently, the tasks of generating the members (solids) in a family and testing if a given solid belongs to an assumed family do not appear to be well defined. The solid families are often categorized as 'parametric' or 'variational,' depending on a particular representation scheme [2]. This distinction is not always clear but is suggestive that, in contrast to the notion of a rigid solid, the notion of family is *not* representation free. The differences between the two types of families, as well as semantic difficulties, are easy to illustrate in the context of the two classical solid representations: CSG (Constructive Solid Geometry) and b-reps (boundary representations).

CSG representations are implicit, high-level, globally parameterized representations of solids that naturally define parametric families. Figure 1(*a*) shows a CSG representation $F(\mathbf{P})$ of a 3-D solid that was created as a union of three cylinders *A*, *B* and *C*. The cylinders are parameterized by their radii, heights and relative positions that act as parameters **P** of the family. When values of the parameters $P_i \in \mathbf{P}$ are restricted to be in a certain range, $F(\mathbf{P})$ defines a family. Assignment of distinct values to the parameters produces distinct members in the family. Figure 1(*a*) shows a parameter P_1 which controls the relative position of cylinder *A* with respect to the cylinder *C* (by constraining its axis). All the solids shown in Fig. 1 could be considered as members of the same CSG family, since all of them were parameterized using the same parameter set **P** *induced* from the same CSG representation *F*.

This concept of a CSG family places few restrictions on the shape topology of the member solids. For example, we can see that the solids shown in Fig. 1(*b*) and 1(*c*) are very different topologically. Even though such families are not intuitive in E^3, they are accepted in many proposals (see for example [4]), and are useful for a variety of design applications: visualization, shape-space exploration, and others. However these families are not acceptable in applications such as tolerancing [5] and pattern recognition [6], where the solids in a family need to obey certain topological properties.

In contrast, b-reps explicitly store the shape of the boundary of the solids they represent, thus proving the natural means for constructing a variational family through direct modification of the solid's boundary. Figure 2(*a*) shows a b-rep *K* representing the boundary (2-D set) of the solid shown in Fig. 1(*a*). B-rep *K* is a union of five faces (three cylindrical and two planar). It is well known that b-reps are harder to parameterize globally [2]. But b-reps can be parameterized locally using the angles, distances or other well defined local geometric properties among its cells. This gives us a direct control of the shape of a solid by transforming its boundary either by varying the local parameter values or, by 'tweaking' (transforming) the cells and thus defining b-rep families. In this case, the set of parameters **P** is *added* to the representation scheme in order to control and constrain a larger family.

Unfortunately these operations (tweaks) are only beginning to be understood formally [7]. Consider what happens when we tweak (or move) the cylindrical face f_1 of *K* by the transformation shown in Fig. 2(*a*). Assuming that the operation results in a valid solid, we expect to see minor geometric changes to the face f_1, and perhaps to the other two cylindrical faces. Figure 2(*b*) shows the result of this tweak computed by a popular commercial system. Clearly, the produced change in geometry of f_1 and adjacent cylindrical faces is inconsistent with the intended tweak, but the precise cause of the problem may not be obvious. We shall see in section 2.3 that this update *might* in fact be considered correct, even if not intuitive; we will also explain how to prevent such updates from consideration in section 3.3.

Modern systems typically combine constructive parametric rep-

Contributed by the Computer Aided Product Development (CAPD) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received September 2002; revised November 2002. Guest Editor: N. Patrikalakis and K. Lee.

[1]The concept of families of parts is not new in engineering and has been used informally in many applications, e.g. standard part catalogues in design and group technology in process planning [1].

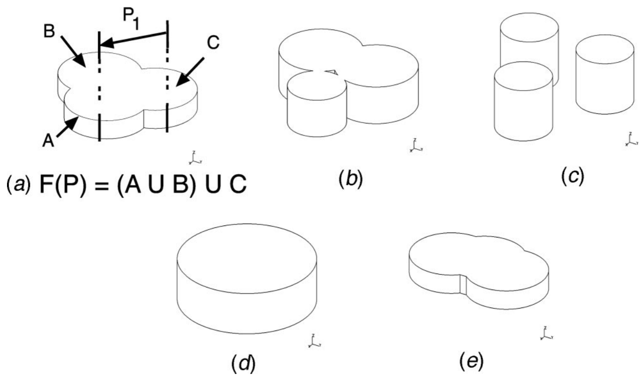

Fig. 1 Family defined using a CSG representation

resentation with its b-rep, resulting in a hybrid *feature* representation and hence attempt to model hybrid parametric-variational families. This combination of families in turn leads to incompatibilities and raises more complex issues than the individual families [8]. For instance, the CSG and b-rep families shown in Fig. 1 and 2 are not contained within each other: we cannot continuously tweak the b-rep K in Fig. 2(*a*) into either of the two solids shown in Fig. 1(*b*) or 1(*c*); and the b-rep L in Fig. 2(*b*) does not correspond to the union of three cylinders any longer. The membership of a solid in a CSG family may be checked using techniques described in [8], but the more general question of what is an appropriate family in this case clearly has more than one answer.

Consider the simple solid S shown in Fig. 3(*a*) with its parametric and boundary representations (derived from the parametric representation) super-imposed. The constructive representation uses a number of parameters, including parameters t and d that constrain the locations of primitives—big cylinder (C_2) and small cylinder (C_3), with respect to the solid's edges as shown in the figure. The value of t determines the location of edge e_1, which in turn determines the location of C_3, as it is constrained by parameter d. Observe what happens during a parametric edit of t. When the value of t becomes equal to the radius of C_2, the face f_1 should continuously collapse and the edge e_1 should coincide with e_3, and we would expect the resulting solid shown in Fig. 3(*b*). Surprisingly, even such a simple update is not handled consistently in most commercial systems. The result from a commercial system could either be the solid shown in Fig. 3(*c*) or the solid shown in 3(*d*), depending on the type of features used in the parametric representation for creating the two cylinders C_2 and C_3 in the original solid shown in Fig. 3(*a*). When C_2 was created as a 'hole' feature and C_3 was created as a 'cut' feature, we obtained the result shown in Fig. 3(*c*). On the other hand, for the same parametric update the solid in Fig. 3(*d*) resulted when C_2 was created as a 'cut' and C_3 was created as a 'hole' in the original parametric representation. For the same parametric update but another construction technique, the system simply signaled an error indicating that it could not locate the proper edge. Yet for another method of constructing the solid, the system might delete the constraint (and the corresponding feature C_3) with an appropriate warning message for the *same* parametric update. Thus we see that for the same parametric update within a system, we get different and inconsistent results, depending on the internal representations and algorithms used for establishing a correspondence between the two b-reps. (The latter problem is known as the persistent naming problem that was formally characterized and partially solved in our earlier work [3].)

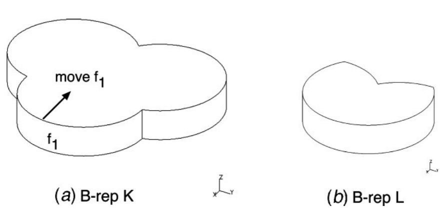

Fig. 2 Incorrect update for a b-rep "tweak"

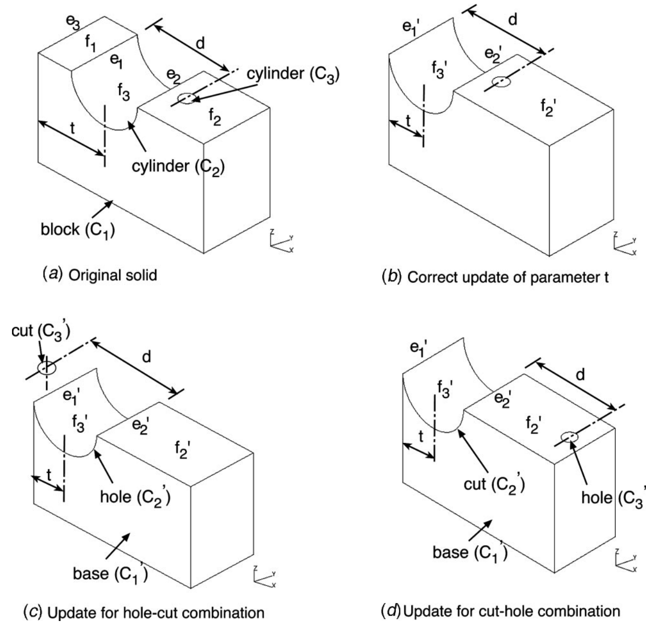

Fig. 3 Ambiguous updates within a parametric modeling system [3]

Such ambiguous and unpredictable updates are not restricted to any one system, and all CAD systems are victims of the lack of a standard notion of a family. Figure 4(*a*) shows a parametric representation of a solid with its b-rep super-imposed. This solid was constructed as a union of a big block C_1 ('base' feature) and a small block C_2 ('boss' feature). Then a 'hole' C_3 was made on C_2. The parameters P_1 and P_2 constrain C_2's position with respect to C_1 via the edges e_4 and e_5, and the parameters P_3 and P_4 constrain C_3's position with respect to C_2 through the edges e_1 and e_2. The values of P_1 and P_2 are set to zero. The resulting

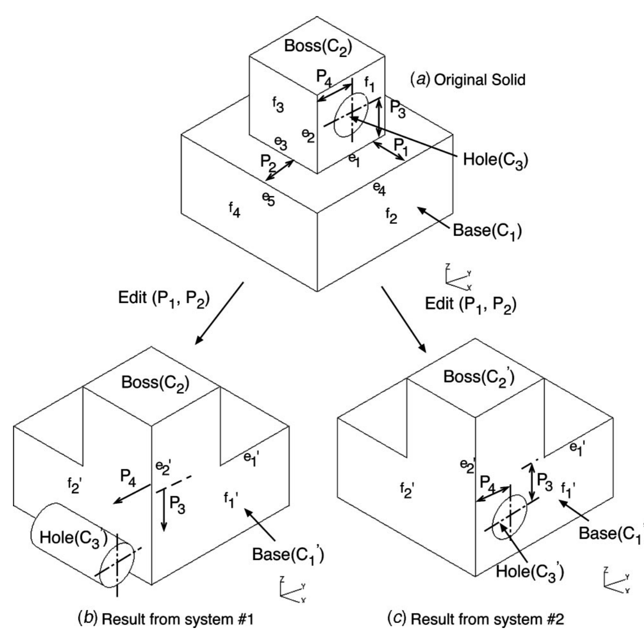

Fig. 4 Inconsistent parametric update in two different systems

solid in one system is shown in Fig. 4(*b*), where the hole C_3 jumps out of the rest of the solid with its parametric constraint P_3 inverted relative to its old position. When the same parametric edit was executed in another system, it resulted in the solid shown in Fig. 4(*c*). In this case the hole jumps onto the larger block from the smaller block with its parametric constraints P_3 and P_4 as shown.

It is noteworthy to mention that all the examples illustrated so far are very simple solids that can be represented with just three or less primitives/features and so it may appear that one can modify the parameterization of the solid's representation to avoid the above mentioned problems. But solids used to model real mechanical parts are usually much more complex requiring several hundreds of primitives/features and it may not be easier to induce a 'safe' parametrization that can avoid such problems during the update. The above examples and discussion clearly suggest that we are missing a set of postulated principles needed to formulate the concept of a family, before any of the above problems are solved. This paper proposes such a set of principles and illustrates how they can be used to design and implement parametric families in practical situations. The material is based largely on the doctoral thesis of the first author [9] that contains many additional details and examples, including a fully implemented set of algorithms to maintain an intuitive CSG-induced parametric family. These are not discussed in this paper due to space limitations.

1.2 Approach. We have no choice but to accept that a parametric family of a solid is not unique, but is determined largely by the solid's representation *and* a set of parameters **P** that are either induced or associated with the particular representation. Therefore, we must agree on a set of representation-free principles that can be applied to any solid representation and parameterization. This is our only hope to provide objective measures for eliminating unpredictability and ambiguities illustrated above. We need a precise notion of objects and transformations (or operations) on these objects which can *generate* other objects in a family, or *classify* two objects that are potentially in the same family. The transformations should ideally obey certain well-defined rules so that they *consistently* and *unambiguously* operate on the objects. We postulate that the two key principles on the objects and transformations respectively of *any* part family are:

1. *Nearness*: A notion of nearness among the elements (parameters, primitives, cells, etc.) of the part representations should be definable.
2. *Nearness preservation*: The operations in the part family should *preserve* or respect the notion of nearness.

In section 2, we will show that the nearness requirement is essentially satisfied by defining a **topology** on the part representations. This in turn transforms the requirement of nearness-preservation into a principle of **continuity**, and strongly suggests that every family should be viewed as a **category** of topological spaces.

We can distinguish two complementary classes of part families: *parameter-space* (defined in the space of parameters induced from or added to a representation scheme) and *representation-space* (implied by the properties of the representation scheme as a whole). Because every solid representation may be viewed as a spatial decomposition [10], we shall see that all representation-space families can also be defined using an appropriate type of decomposition of space. Composite families (defined in both parametric and representation space) can be constructed by restricting operations in the family.

Based on these fundamental principles, we seek a systematic framework: (1) to define part families using any of the existing well understood classes of solid representations; (2) to define the semantics of operations in families and control the properties of the families through an ability to specify additional desirable properties; and finally, (3) to compare the properties of various representation-space families with respect to each other.

The latter requirement suggests that we also need a notion of a representation-free family associated with a solid as a pointset. Such a *pointset family* should define a class of pointsets associated with some given solid, in the same spirit as mathematical models are associated with all solid representations [10,11]. Since the pointset families are not dependent on any solid representation scheme, we should be able to state their properties irrespective of how they may be represented.

1.3 Outline. The rest of the paper is organized as follows: in section 2 we will introduce a category as the formal notion of a family. Using categories we can precisely formulate the problems of generating objects in a family and classifying an arbitrary object against a given family. The proposed framework allows systematic comparison and exploration of possible definition of families and associated computational issues, as we will illustrate this in the context of CSG, b-rep, cell complex, and feature representation families induced from the corresponding representation schemes. In section 3 we will formally define a composite CSG-complex family and illustrate its properties through examples. The summary in section 4 includes a brief discussion of other composite families, some applications of the framework and open issues. Additional details may be found in [9].

2 Framework for part families

2.1 Categories. Past research in solid modeling has shown that topology is the proper setting for modeling physical parts [12], and hence we turn to topology to develop the appropriate framework for modeling families of parts. The most general idea of families[2] of objects is captured by the notion of *categories* in topology [13–16]. Categories are the mathematical tools of sufficient generality and universality, having applications in several diverse computational areas such as pattern recognition [6], programming languages [17], and databases [18], to name a few. Using the standard definition of a category from Munkers [16] and Jänich [13], we adapt this simplified definition of a category.

Definition 1 (Category of Objects) *A category $\mathcal{C}$ consists of the following:*

1. *A class of objects X called objects (Ob) of the category.*
2. *For every ordered pair X, Y, of objects, a set $hom(X,Y)$ of morphisms or operations f acting on the objects, where the operations obey the following three mathematical rules:*

- *obey a law of composition*
- *obey the law of associativity*
- *existence of identity morphism*

The set of morphisms $hom(X,Y)$ between objects X and Y includes the identity morphisms 1_X, 1_Y, the forward morphism $f{:}X \rightarrow Y$ and possibly its inverse morphism f^{-1}, if it exists. The axioms assure existence of non-trivial complex transformations of objects that preserve the essential properties of the category (for more detailed discussion on these three laws refer to [9]). Numerous examples of categories are known [16], but the two most important categories for our purposes are the **category of sets** and the **category of topological spaces**. The category of sets consists of sets as objects, and the maps between the sets as the morphisms. In the context of solid modeling, the notion of (family) category of sets is representation-free. A category of topological spaces consists of topological spaces as the objects and continuous maps acting on the topological spaces as the morphisms. In the context of solid modeling, topological spaces are often associated with different representations of the solid (point set). Two very practical and useful examples of categories of topological spaces are:

[2]Our concept of family is different from the mathematical notion of family which is often informally used to refer sets of sets, and suffers from some well known paradoxes [13].

1. the category of simplicial complexes, consisting of simplicial complexes as objects and simplicial maps as morphisms; and
2. the category of cell complexes, consisting of cell complexes as objects and cell maps as the morphisms.

Sub-categories can be defined by imposing restrictions on the objects and/or morphisms. For example, a cell complex family is a sub-category of the point set family and, as we shall soon see, a b-rep family is a sub-category of the corresponding cell complex family; the latter is defined using cells of all *n*-dimensions whereas a b-rep only contains cells up to dimension $(n-1)$ for a solid in E^n.

We can now define the most general notion of *nearness* in terms of neighborhoods of a topological space, and *nearness-preserving* operations as continuous maps (morphisms) in the corresponding category of topological spaces. A given point set X belongs to many distinct families (categories) determined by different methods for topologizing X and variety of morphisms.

From the definition of topology [13,19], we know that any topological space is collection of open sets and can be defined by either explicit specification of the open sets (neighborhoods) in it or, by specifying the basis/sub-basis of that space. Defining a topological space via the basis is easier than specifying all the open sets in a topology, since we require only finite sets to define a basis [13,19]. By definition, *any* topological space on a given set X (or its representation ϕ) should contain X (or ϕ) and $\emptyset$. Further, all the elements in the basis also belong to its defining topology [19]. A continuous map between two topological spaces $\mathcal{X}$ and $\mathcal{Y}$ takes open (closed) sets of $\mathcal{X}$ into open (closed) sets of $\mathcal{Y}$. When the topological spaces are defined using the basis, then continuity is applied basis-by-basis and should be extendable to the whole topological space.

The morphisms in a category are operations on the objects under which the category is closed. If a morphism f has an inverse, then f is called an *equivalence* in the category in question. Some examples of equivalences are [16]: bijective correspondences, homeomorphisms, simplicial homeomorphisms and cell homeomorphisms. Ideally we would like every family to be a class of *equivalent* objects, and thus all the morphisms to be equivalence relations. But it appears that engineering practice in general, and mechanical applications in particular, often imply some partial ordering between the objects in the family. For example, when part A is modified into a part B, it is normal to expect that the variant part B remains in the same family; but family of B may or may not contain the original object A depending on the representation and parameterization of B. Another manifestation of this asymmetry is the existence of one special object of the family known as *nominal* object that is particularly useful when dealing with dimensional variations and tolerances [5,20].

Given a representation ϕ of a solid X, let $\mathcal{T}_\phi$ denote a topology that is either assumed or induced from ϕ on X. Then the family of solids $\mathcal{R}$ is the topological category of pairs $(\phi,\mathcal{T}_\phi)$ under some suitable collection of morphisms. The family $\mathcal{R}$ containing solid X may be defined in one of two ways:

1. **Generation:** Given a nominal object $(\phi,\mathcal{T}_\phi)\in\mathcal{R}$ and a known continuous morphism g, generate another object $(\psi,\mathcal{T}_\psi)$ presumably represented and topologized in the same representation scheme as ϕ.
2. **Classification:** Given a nominal object $(\phi,\mathcal{T}_\phi)\in\mathcal{R}$ and another object $(\psi,\mathcal{T}_\psi)$ in the same representation scheme, classify ψ with respect to $\mathcal{R}$ by constructing a morphism g^{-1} from $(\phi,\mathcal{T}_\phi)$ to $(\psi,\mathcal{T}_\psi)$.

This classification problem is analogous to the well-known mathematical problem of classifying topological spaces in topology [16]. In general, the classification of topological spaces may involve identification of all the invariants of the spaces. Some well known invariants of 2-dimensional spaces (surfaces) include Euler characteristic and Betti numbers.

The notion of category is representation-free, but representations induce topology and also enable computations. Let us now compare the families induced from the common representation schemes by particular choices of topology and morphisms.

2.2 Parameter-Space Families. The simplest example of the family is a parameter-space family, where the family is defined in the parameter space only, without concern for the resulting representation *or* the point set. A solid that is parameterized by n real-valued independent parameters (induced by a nominal solid representation) is simply a point in the parameter space R^n, and the parameter values when constrained within a range define a sub-space of R^n. Every point $p\in R^n$ has a n-dimensional neighborhood and the set of all neighborhoods is the topology in parameter space. Alternately, every n-dimensional neighborhood can be considered as a product of all the 1-dimensional neighborhoods or open intervals defined on each parameter. The morphisms are continuous maps that change the values of the parameters and thus map parameter values (neighborhoods) into parameter values (neighborhoods). Every assignment of the values to the parameters corresponds to a representation of some solid part, but these representations (or point sets they represent) are not related or transformed directly. In other words, generation and classification of objects in the family may be straightforward in the parameter space, but difficult or impossible in the representation space or E^3. Such families are well defined but are not always desirable because they often result in non-intuitive point sets and notion of continuity, as demonstrated by the parameter-space CSG family in Fig. 1. Furthermore, depending on representation scheme, not every assignment of values may be valid; for example, feature representations are only valid for some range of parameter values. The problem of computing such ranges is well defined [2], but solutions appear to be available only in the simplest cases [21].

2.3 Cell Complex Families. In order to control the shape of the solids in the family, we must define the family as a category of topological subspaces of E^3. When a point set is represented by a cell complex, the open neighborhoods of individual cells in the complex define a finite basis naturally. An appropriate open neighborhood of each k-cell is defined by the *star* of the k-cell. A star (St) of a k-cell σ is the union of σ with all higher-dimensional cells incident on σ. The natural topology $\mathcal{T}_K$ of a cell complex K (where $|K|=\cup\sigma_i$) consists of all open sets formed by unions and intersections of sets $St(\sigma_i)$.

Thus, every cell-complex representation of a point set X provides a natural (but not unique) method for topologizing X by the unions of the cells in the complex. General cell complexes may be heterogeneous in dimension, contain cells of all dimensions, typically $k=0,1,2,3$ in solid modeling applications, and define spaces with unrestricted topological properties. Boundary representations in E^d are special homogeneously $(d-1)$-dimensional cell complexes [11,22] that are also required to be $(d-1)$-cycles or manifolds. A number of cell-complex families may be defined, depending on the type of cells *and* on the allowed transformations (morphisms) of the cell complexes. Assuming the most common type of geometric cell complex [3], there is a choice of at least three continuous transformations, listed here in the order of increasing natural and intuitive appeal: cell maps, orientation-preserving cell maps, and cell-complex deformations (homotopies). Cell maps result in the largest family of cell complexes characterized by allowable cell-by-cell morphisms; orientation preserving cell maps produce a proper restriction of this family, and homotopy produces the smallest and the most restricted cell-complex family of the three. These transformations are characterized precisely in [3], including specific incidence and relative orientation conditions that must be enforced by the corresponding transformations. It should not be surprising that additional restric-

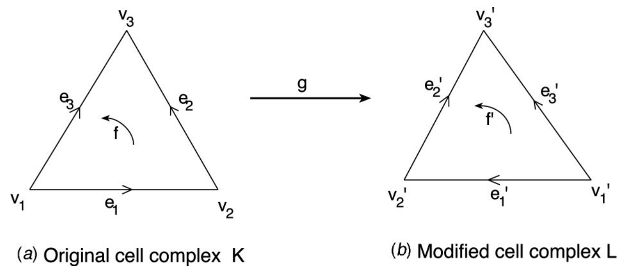

Fig. 5 B-rep family using cell maps

tions on the transformations are also progressively more expensive to maintain in data structures and more difficult to enforce.

For example, in Fig. 5 a topology on K can be defined using the star of each vertex (e.g: $St(v_1)=\{v_1,e_1,e_3\}$). With this topology the two 2-D solids in Fig. 5 belong to the same b-rep family defined by the shown cellular structure and continuous cell maps (the map g in this case is given by $g(v_i)=v_i'$). But the two triangles do not belong to the same family defined by the orientation-preserving cell maps because no such map exists in E^2. In contrast, the two solids in Fig. 3(*a*) and 3(*b*) are in the same b-rep families defined either by the cell maps or by the orientation-preserving cell maps. It is easy to check that the two b-reps in Fig. 2 also belong to the same b-rep family, despite the obviously unintended outcome. The two cell complexes in Fig. 6 belong to a cell complex family, where the cell complex in Fig. 6(*a*) is the nominal object and cell maps are the operations. Here the star of every vertex consists of the collection of 1-cells and 2-cells adjacent to the vertex, in addition to the vertex itself. Note how the continuous cell complex maps can reduce the dimension of a cell from 2 to 0 as shown in Fig. 6(*b*).

Other definitions of morphisms leading to distinct families are possible. For example, we could define two cell complexes to be in the same family whenever one of them can be deformed into another, or when both of them can be deformed into another. The latter family is an equivalence class defined by homeomorphic transformation of cell complexes and is very different from the family of all cell complexes that can be obtained by continuous (but not necessarily invertible) deformations of one 'master' cell complex K.

Among cell-complex families, b-rep families are particularly popular because they do not represent the cells of the highest dimension, resulting in data structures that are substantially smaller and easier to represent. However, b-rep families also suffer from serious limitations. For example, if continuity is defined in terms of a homotopy, the transformation must be applied not only to the boundary of the cell complex but to the whole space. Classification of objects in such a family is impossible when the objects are represented by their boundaries alone [3,8]. In this sense, the 3-dimensional cell-complex families permit a higher level of guarantee and sufficiency of the continuity than the respective b-rep families.

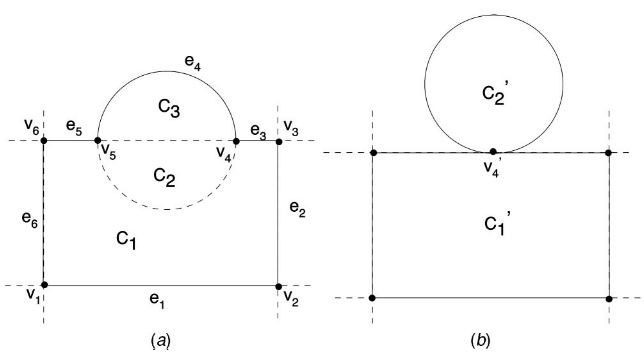

Fig. 6 Representation-space family defined using a cell complex

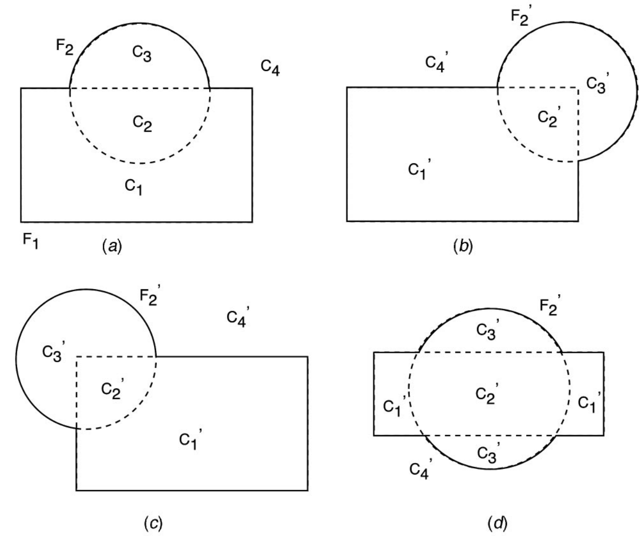

Fig. 7 Representation-space family defined using a feature representation

It is shown in [9] that other useful cell-complex families may be obtained by eliminating the restrictions on the type of cells and cell complexes (for example, allowing disconnected cells, eliminating cells of certain dimensions, and so on). In section 3 we shall see that such families can be both practical and useful.

2.4 Feature Representation Families. Feature representations can be thought of as a composition of parametric and cell complex representations. Combinatorially, features are often defined as *collections* of cells or sub-complexes of a three-dimensional complex [23–26]. Unlike the cells in a geometric cell complex, features need not be open, connected or smooth in E^d, but the corresponding collections of cells may be used *in place of* individual cells to topologize the feature representations. For example, suppose that for a given feature representation F, we have $|F|=\cup C_i$, where C_i denotes a collection of cells. The star of every such collection C_i serves as the neighborhood or an open set in a feature representation induced topology. In this case, the topology is defined in terms of collections and hence is more coarse than the natural cell complex topology on the same point-set. A variant of such a star neighborhood for collections of full-dimensional cells will be defined in section 3 and will used to topologize a CSG complex. The topology $\mathcal{T}_F$ of the feature representation F is a collection of all sets formed by unions and intersections of $St(C_i)$, including $|F|$ and $\varnothing$. The continuity in a feature representation family is applied feature (sub-complex) to feature (sub-complex), resulting in a very different family of objects.

Figure 7 shows a simple feature representation induced family defined over the given nominal feature representation shown in Figure 7(*a*). The feature representation is defined using only two 2-D features: rectangle and a circular disk. We can define a topology on this feature representation by considering it as a union of the three 2-dimensional collections C_1, C_2 and C_3, that classify as IN with respect to the solid. The fourth collection C_4 classifies as OUT with respect to the solid (i.e, corresponds to the complement of the solid). The star of every collection C_i contains all the collections $(St(C_1)=St(C_2)=St(C_3)=St(C_4)=\{C_1,C_2,C_3,C_4\})$ in all the four feature representations and hence all of them belong to the same family. Depending on the relative position and size of the two features, collections C_i could contain single cells (Figs. 7(*a*), (*b*), and (*c*)), or several cells, as is the case with collections C_1' and C_3' in Fig. 7(*d*).

On the one hand, this family allows many transformations that would not be considered continuous in terms of the finer topology generated by the individual cells in the complex; on the other hand this family may not be always intuitive because it is defined using much coarser topology. A more intuitive family that combines the usual cell-complex family and the feature family may be constructed by topologizing the solid with feature collections *in addition to* the individual cells in the cell complex. These are precisely the families that must be maintained in order to support the updates shown in Figs. 3 and 4.

The collections of cells in feature-based representations may be also characterized as connected components of the CSG representations [27]. In this sense, feature representation families are a generalization of CSG-complex families that are defined and examined closely in section 3.

2.5 Comparison of Families. The benefits of our formal proposal should be already clear, because we can now compare and assess many properties of the families that did not seem quantifiable before. Parameter-space families are clearly easier to generate than classify, because the latter requires constructing the inverse mapping from the Euclidean space to the parameter space. On the other hand, classification is easier in cell-complex families than generation when all cellular representations assume the same type of the topology on the given solid representation, because classification reduces to matching the cells, incidences, and orientations [3].

Different representation schemes correspond to different types of space decompositions and therefore different methods for topologizing the underlying space (solid in a family) [10]. To some extent, finer the topology induced by a representation, the closer it is to the usual Euclidean topology, but some continuous transformations, for example those corresponding to feature edits, may be enforced only on very coarse decompositions and topologies. Hence we may have to strike a balance between the fineness of the topological spaces and the ability to enforce the continuous morphisms between them.

Our analysis suggests that a reasonable proposal for a part family should include several desirable properties. The family should be defined both in parameter space and in the shape (Euclidean) space in order to support both generation and classification of family members. In the Euclidean space, the continuity should be defined in terms of a cellular decomposition that is rich enough to define a useful topology, but is also practical to compute and maintain. Assuming that the b-reps and 3-dimensional cell complexes employ the same type of decompositions and topology, the b-rep families are sub-categories of the more general cell complex families. The lower dimensional cells are needed to specify several important operations or features such as blends/fillets, specifying the boundary conditions for engineering analysis and often are used to define a local coordinate system for attaching other features or primitives (as illustrated by the feature representations shown in Figs. 3 and 4). Furthermore, certain classes of edits (for example, collapses such as those in Fig. 6) cannot be described without lower dimensional cells. But on the other hand, the 3-cells can be computed more robustly and a family defined using only the 3-cells can be maintained more easily than the cell complex family containing cells of all dimension.

In what follows, we will rely on these observations and the proposed theory to design a new part family that we call a *CSG-complex family*, because it combines the parameter space family implied by the CSG representation with the shape-space family defined by the decomposition of space associated with every CSG representation.

This space decomposition relies only on 3-dimensional and possibly disconnected cells that have unique names and are easy to compute. The importance of this family is two-fold: (1) it illustrates how to apply the proposed theory in practice in order to engineer families of parts with desired and predictable properties; and (2) the described family is a special case of the more general feature representation family described in section 2.4.

3 CSG-Complex Families

Viewing CSG representations as algebraic expressions that are decoupled from geometric embedding leads to the usual view of a CSG family as a parameter-space family. We already discussed the limitations of such families in sections 1.1 and 2.2. In this section, we show that CSG families may also be defined as categories of representations in E^3 with appropriately defined continuous transformations. This illustrates application of the proposed framework in a non-traditional setting, and suggests how the same approach may be applied to more complex families.

3.1 Canonical Cellular Form of CSG. As a Boolean function with finite number of variables (primitives), every CSG representation of a solid can be written in a unique disjunctive canonical form as a union of intersection terms [28]. The regularized intersection terms are homogeneously *d*-dimensional subsets of Euclidean space E^d that we will call *atoms*. Each *atom* is closed regular but possibly disconnected set that could contain holes or voids. For a fixed set of primitives, the *atoms* form a decomposition of the Euclidean space [10].

We define a *CSG complex* to be any collection of such *atoms*. A CSG complex does not satisfy the usual axioms of cell complexes, because the *d-atoms* are quasi-disjoint (either disjoint or intersect along their boundaries) sets of the same dimension *d*. The resulting cellular structure on the solids implies a 'natural' topology for CSG complexes that is quite different from the usual Euclidean sub-space topology when the same solids are considered as cell complexes.

3.2 Topology on CSG-Complexes. We define a topology on a CSG-complex *D* with the *d-atoms* in *D* and define the open sets or neighborhood of each *d-atom*. Following the usual practice, a neighborhood in the complex can be defined combinatorially in terms of neighboring cells in the complex. Specifically, since we are interested in defining topology on a pointset, we shall rely on the following set-theoretic notion of star from Lefschetz [29].

Definition 2 (Star of an *atom*) *The neighborhood or star (St) of a d-*atom C_i *in a CSG-complex D is a union of* C_i *with all the d-*atoms *in D whose (non-regularized) intersection with* C_i *is non-empty*.

Thus, in a CSG-complex *D*, the star of an *atom* $C_i \in D$ can also be defined in a computationally convenient manner as a collection of its neighboring *atoms*. In other words, the star of a 3-*atom* C_i in a 3-D CSG complex *D* is the union of C_i and all the 3-*atoms* that share a common boundary (vertex, edge or face) with C_i-even if these lower dimensional cells are not represented explicitly. We now can define a topology $\mathcal{D}$ on *D* using the sets $St(C_i)$ as the basis. In what follows we will interchangeably use both set-theoretic and combinatorial definitions of a star depending on the context.

Consider the CSG-complex *D* corresponding to the decomposition of the plane as shown in Figure 8. We can visualize the defined topology for this 2-dimensional CSG-complex by enumerating vertex and edge neighborhoods of each face (2-*atom*) with respect to the shown set of primitives. Each *atom* is defined by intersection of all five halfspaces h_i or their regularized complements. Neighborhoods of the 2-dimensional atoms are clearly defined by the adjacent atoms. Edges and vertices are not defined explicitly in a CSG complex, but their neighborhoods may be enumerated in terms of the 2-*atom* (face) neighborhoods. For example, Fig. 8 shows a vertex neighborhood N_3—four faces C_1, C_2, C_3 and C_4 share that vertex, and thus are in one another's neighborhood. The complete neighborhood of the face C_1 in-

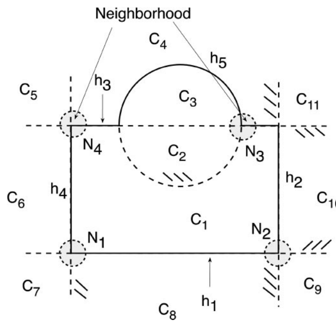

Fig. 8 Topologizing a CSG complex

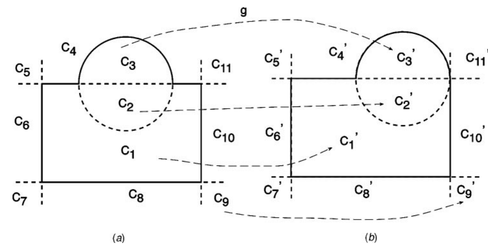

Fig. 9 Continuous CSG map from the CSG complex in (a) to the CSG complex in (b)

cludes the vertex neighborhood at all of its other component vertices, and hence it includes the neighborhoods N_1, N_2, N_4 and so on.

It should be clear that CSG-complex topology is much coarser that the usual topology defined on a geometric cell complex, because every CSG-complex neighborhood corresponds to a union of neighborhoods in the latter. We also note that this is not the only way to define a topology on a CSG complex. For example, recall that every *d-atom* is a union of possibly one or more connected sets, and the neighborhood of a *d-atom* can also be defined as the union of its connected component neighborhoods. As we discussed earlier, the resulting topology would be much more appropriate for the feature representation families, but it would also possess very different computational properties, because connected components of some *atoms* may not admit any CSG representation with a given set of primitives [30].

3.3 Continuity in CSG Topology. We are now ready to define the notion of continuity for CSG representations in a manner that captures the notion of nearness and nearness-preservation for the *atoms* in a CSG complex. For any CSG representation, we need to choose an appropriate CSG complex and define the continuity between any two CSG representations in the corresponding topological spaces. If we want to assure continuity on the point set only, then it is sufficient to include in CSG complex only those *d-atoms* that classify IN with respect to the solid. A continuity on the whole space requires CSG complex to include all cells in the decomposition of space. We chose the latter, because it gives a stronger and more intuitive notion of continuity for CSG representations.

Consider a typical editing operation. Given a (pre-edit) CSG-complex D and another (post-edit) CSG-complex D', let us denote the corresponding topologized complexes as $\mathcal{D}$ and $\mathcal{D}'$ respectively. If *atom* $C_i \in \mathcal{D}$ is given by the canonical intersection term of primitives $\mathbf{P}$ in the pre-edit CSG representation, then *atom* $C_i' \in \mathcal{D}'$ is symbolically the *same* intersection term of primitives $\mathbf{P}'$ in the post-edit CSG representation. In other words, the correspondence between the atoms C_i and C_i' is well defined by the Boolean expression, even if C_i' is empty. This assures that both CSG representations belong to the same parameter-space family. Recall the problematic update shown in Fig. 2: it is easy to show that the post-edit solid is no longer represented by the original CSG representation of $A\cup B\cup C$, with primitive cylinders A, B, C shown in Fig. 1. This implies the pre-edit atoms C_i do *not* correspond to the post-edit atoms C_i'. The update is incorrect because existence of the continuous cell map between these two b-reps is not sufficient to establish the continuity in the parameter space. The latter is assured by the correspondence between the atoms and can be easily checked using techniques and algorithms detailed in [8].

The concept of continuity can now be extended to the topologized CSG-complexes. We have defined the topology of a CSG-complex D using the open sets (star of an *atom*) in D. This implies that continuity between two CSG-complexes is defined *atom*-by-*atom* [13,16], and has to be extendable to *all* the elements of $\mathcal{D}$ and $\mathcal{D}'$. Applying Cauchy's definition of continuity [19] for CSG-complexes leads to the following definition.

Definition 3 (Continuous CSG-complex map) *A map g from a topologized CSG-complex $\mathcal{D}$ to a topologized CSG-complex $\mathcal{D}'$ is called a continuous CSG-complex map if it maps every* atom $C_i \in \mathcal{D}$ *into a corresponding* atom $C_i' \in \mathcal{D}'$ *such that*:

$$g(St(C_i)) \subseteq St(g(C_i)). \tag{1}$$

The condition (1) in the above definition is also known as the *neighborhood-preserving* or star condition that must be true for every continuous CSG-complex map g. When a CSG-complex map g is also one-to-one and invertible, then g will be called CSG-complex homeomorphism.

Let us illustrate the properties of CSG-complex maps through the CSG complexes in Fig. 9. The one-to-one map g takes 2-*atom* C_1 to C_1', 2-*atom* C_2 to C_2' and C_n to C_n' (for all other *atoms* C_n, where $n=3, \dots 11$). Further, the star of *atoms* C_1 and C_2 are:

$$St(C_1)=\{C_1, C_2, \dots C_{11}\},$$

$$St(C_2)=St(C_3)=\{C_1, C_2, C_3, C_4\},$$

$$St(C_1')=\{C_1', C_2', \dots C_{11}'\},$$

$$St(C_2')=St(C_3')=\{C_1', C_2', C_3', C_4', C_{10}', C_{11}'\},$$

$$g(St(C_1))=g(C_1, C_2, \dots C_{11})=\{C_1', C_2', \dots, C_{11}'\},$$

$$St(g(C_1))=St(C_1')=\{C_1', C_2', \dots C_{11}'\}$$

$$\Rightarrow g(St(C_1))=St(g(C_1)),$$

$$g(St(C_2))=g(C_1, C_2, C_3, C_4)=\{C_1', C_2', C_3', C_4'\},$$

$$St(g(C_2))=St(C_2')=\{C_1', C_2', C_3', C_4', C_{10}', C_{11}'\}$$

$$\Rightarrow g(St(C_2)) \subset St(g(C_2)).$$

In a similar fashion we can show that the star or neighborhood-preserving condition holds for all other 2-*atoms* in the CSG-complexes. In other words, g is a continuous CSG-complex map.

On the other hand, consider the mapping g between $\mathcal{D}$ and $\mathcal{D}'$ in Fig. 10. The complete mapping g is given by:

$$g(C_1)=C_1', \quad g(C_2)=\emptyset, \quad g(C_3)=C_2',$$

$$g(C_4)=C_3', \quad g(C_5)=C_4', \quad g(C_6)=C_5',$$

$$g(C_7)=C_6', \quad g(C_8)=C_7', \quad g(C_9)=C_8',$$

$$g(C_{10})=C_9', \quad g(C_{11})=C_{10}'.$$

Now we need to check whether g also takes the star of each *atom* into its image *atom*'s star. For instance:

$$St(C_2)=St(C_3)=\{C_1, C_2, C_3, C_4\},$$

$$g(St(C_2))=g(C_1, C_2, C_3, C_4)=\{C_1', C_2', C_3'\}$$

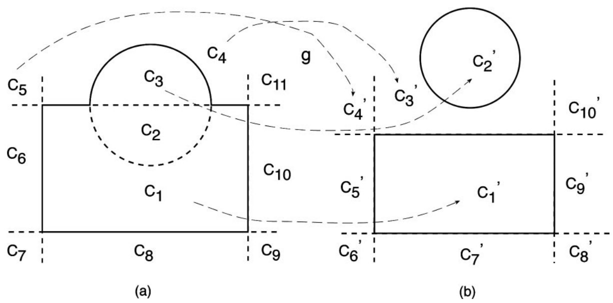

Fig. 10 Correspondence between CSG-complexes (a) $\mathcal{D}$ and (b) $\mathcal{D}'$ is not a CSG map

and $St(g(C_2))=St(\varnothing)=\varnothing$. Thus $g(St(C_2))\nsubseteq St(g(C_2))$. It can also be shown that $g(St(C_3))\nsubseteq St(g(C_3))$.

From the above examples we can see that CSG maps have the intuitive flavor similar to the continuous maps in Euclidean topology. We might even be tempted to conclude that this is always the case. But the example in Fig. 11 shows that the two definitions are not equivalent. The original solid in Fig. 11(*a*) was created by subtracting a torus from a rectangular block. Clearly, there is a bijective map $g(C_i)=C_i'$, between the 3-*atoms* of $\mathcal{D}$ and $\mathcal{D}'$. It is also not very difficult to check that the star condition holds for this map *g*, implying that *g* is a CSG map. Note that the original solid, shown in Fig. 11(*a*), has no 'through hole' (its surface is homeomorphic to a 2-sphere), but the updated solid shown in Fig. 11(*b*) has a 'through hole' (its surface is homeomorphic to a toroidal surface). This implies that the existence of a continuous CSG-complex map between two solids (as CSG-complexes) in the CSG topology does not guarantee a continuous map between the same two solids (as cell complexes or even point sets) in the familiar Euclidean topology.

3.4 Topological Constraints. The non-intuitive nature of the CSG-complex family, illustrated in Fig. 11, is due to the coarser topology defined on a CSG-complex than the topology defined on the same CSG-complex as a sub-space of Euclidean topology. Consequently the solids in a CSG family do not have the same properties when relying on these two topologies. Specifically, the connectivity properties (connectedness, type, etc) of the solids in Euclidean space need not be preserved in this new topology assigned on CSG-complexes, as illustrated in Fig. 11. However we can further modify the definition of a CSG family using many of the tools we have introduced throughout the paper. Not only we can change the type of CSG-complex and topology, but we can introduce additional restrictions on the type of morphisms operating on CSG-complexes.

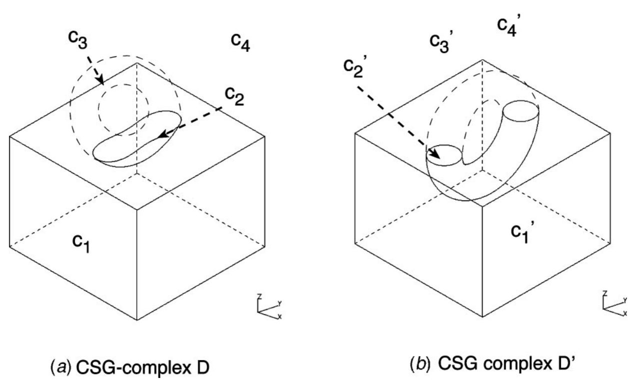

Fig. 11 Non-intuitive update (from (a) to (b)) in a CSG-complex family

From the various examples in section 1 it is clear that for basic engineering requirements, any shape-space family should satisfy certain essential properties (connectivity properties) of Euclidean continuity. We know from elementary topology [13,16] that the connectivity diminishing condition is a necessary condition for continuity in Euclidean topology and is also used for the classification of manifolds [16,31]. For example, recall that the connectivity properties are preserved under homeomorphisms [13]. Hence we can postulate a requirement that morphisms in every representation-space family should at least satisfy the *Euclidean connectivity diminishing* constraint. In particular, in order to make CSG-complex family more intuitive, we could require that

- no CSG-complex map can increase the number of connected components, 'holes' and 'voids' in the solids being represented.

The connectivity diminishing condition allows collapses of the connected components, elimination of holes and voids in the solid, but not the inverse (i.e., the number of connected components, holes or voids can never increase). This is consistent with observation that continuous maps in E^3 cannot transform a solid without holes (homeomorphic to a sphere) into another solid with holes. Similarly Euclidean continuity dictates that a connected solid remains connected. If disconnected solids are allowed, then the number of connected components in the solid before applying a morphism should be greater than or equal to the number of connected components in the solid after applying the morphism. The condition on holes and voids is also important in maintaining the application specific semantics of features [26], and is also present in the definition of tolerance classes [5], pattern recognition [6] and other applications [24].

The Euclidean connectivity diminishing condition on the solids in a CSG-complex family can be also expressed through the Betti numbers B_n of the solids. Each Betti number is related to a corresponding connectivity property of an *n*-dimensional object embedded in *n*-dimensional Euclidean space. Specifically, B_0 corresponds to the number of connected components in a set *X*, B_1 is equal to twice the number of holes/handles in *X*, and B_2 gives the number of connected components in the complement space $\overline{X}$. Consider the example in Fig. 11 again. The solid corresponding to the CSG-complex in Fig. 11(a) has a Betti number $B_1=0$ (same as a sphere) and the solid corresponding to the CSG-complex in Fig. 11(b) has $B_1=2$ (same as a torus). Since this particular update does not obey the Betti number diminishing condition, we disallow it; this in turn implies that the update does not belong to the family. Note that the two solids however *can* belong to the same family, since the *inverse* update from Fig. 11(*b*) to 11(*a*) is a valid morphism that satisfies the Betti number condition. This example clearly demonstrates the lack of symmetry in the definition of a family when we rely on continuous maps. A more restricted symmetric family can be constructed using equivalence morphisms, for example by requiring that the CSG-complex homeomorphisms preserve all Betti numbers.

4 Conclusions

4.1 Summary. We proposed a new framework for part families by identifying the proper mathematical models and connecting them with the common solid representations. This framework is centered around the notion of a category. A category of topological spaces seems to be the most general and appropriate model for representation-induced part families, as they formalize the postulated principles of nearness (*topology*) and nearness preservation (*continuity*) between the elements of part representations in a family. Every family is defined using at least one special object called *nominal* object, and the objects in a family are actually part representations carrying a parametric and/or Euclidean topological structure. The parametric and Euclidean structure of part representations let us define two different types of representation induced families having complementary properties:

- **Parameter-space family:** objects are *particular* representation-induced parameter spaces topologized as subspaces of the parameter space, and the operations are continuous transformations in this parameter topology.
- **Representation-space family:** objects are representation-induced cell decompositions topologized as a subspace of the Euclidean space, and the operations are continuous transformations in this cellular topology.

We demonstrated the applicability of the framework to many different families: parameter-space families, CSG-complex families, b-rep families, cell-complex families and feature representation families. The framework is general enough to include and characterize most reasonable proposals, because every representation scheme corresponds to some decomposition of space [10], and the latter may be used to define the key topological notion of continuity and continuous maps. For instance, the recent proposal by Bidarra and Bronsvoort [4] advocates a semantic constraint-oriented approach to defining part families that is consistent with our framework and appears to be a special case of the feature-representation family. When the decompositions are not topologized as sub-spaces of E^n and coarser topologies are defined on them, additional topological constraints, for example based on Euclidean continuity, may be introduced to make a family more intuitive. In all cases the family is defined formally and without ambiguity.

We argued that any family can be defined either by a generative scheme or by a classification scheme, but observed that different families have different computational properties. We did not discuss any algorithms in this paper, but our framework supports systematic development of algorithms based on the topological properties of assumed morphisms and the properties of the representation schemes. For example, all algorithms for enforcing the defined CSG-complex family, including the Euclidean connectivity diminishing condition, are described in the first author's doctoral thesis [9] and have been fully implemented.

4.2 Engineering Applications. The current problems in parametric geometric design have provided the primary motivation and focus for our work. But a number of other engineering modeling applications stand to benefit from the proposed framework as well. These include:

- **Automatic shape optimization:** Unlike geometric design, shape optimization usually takes place iteratively and without user's interaction, based on the programmed formal criteria. Any such iterative procedure requires an unambiguous definition of the implied parametric family and robust parametric updates that preserve the associated notions of nearness.
- **NC machining:** In NC machining simulation applications for automatic tool path verification, it is necessary that the tool does not introduce any unwanted 'through' holes in the part (i.e, the machined part should have the 'same topology' as the original unmachined part) [5]. This requirement implies that the two parts (pre and post machined) should belong to the same family of parts identified in this case by the Euclidean connectivity diminishing condition (and equal Betti numbers) as described in section 3.
- **Geometric tolerancing:** A notion of a part family is also prominent in the areas of geometric tolerancing and robustness. For example, the tolerance zone approach to defining families proposed by Requicha [20] supports parametric tolerances but appears to be characterized only geometrically; the more recent point set topological approach of Stewart [5] gives a representation-free characterization of the pointset family and hence does not directly support parametric tolerances. Our framework encompasses both parametric and set-theoretic tolerances, at least in principle, and thus is consistent with both proposals.
- **Parametric data exchange and standardization:** The well known standards for geometric data exchange (such as IGES) between CAD systems allow only transfer of CSG and boundary representations. Currently efforts are underway to enable data transfer between more general formats, including parametric and constraint-based representations [32]. But the transfer of parametric representations between CAD systems makes sense only if we agree upon some standards for representation of the solids and their families. Our framework suggests such a standard approach based on common sense and well known mathematical constructs.

The above list is not meant to be exhaustive. Part matching and comparison, search in databases of parts and engineering functional families, redesign, consistency verification, and many other modeling tasks, rely on notions of families that may be derived and constructed systematically using the proposed framework.

4.3 Open Issues. We demonstrated the usefulness of the proposed topological framework for formalizing and standardizing the solid families using numerous examples. To reap full benefits of the approach, our proposal needs be accepted by the solid modeling and the standards communities, and applied to develop a hierarchy of useful families in the context of specific representations and applications.

Due to the complementary properties of the parametric and representation space families, each relying on a single topological structure, part families are commonly designed to rely on two or more part representations simultaneously. This implies that nearness and continuity must be defined on both parametric and Euclidean topological structures, leading to the concept of *composite* families. In the case of CSG-complex family, the chosen cellular structure assures that every *atom* is identified uniquely and unambiguously from the given CSG representation. Such unique naming establishes the correspondence between cells in topologized solids as is required for classification of solids with respect to a family. The price for this convenience was the exceedingly coarse topology of CSG-complex families. Should we have chosen a finer topology, say using connected components of *atoms*, we would be faced with the problem of *persistent naming* of the individual cells and consequently face difficulties in classifying the objects.

More generally, the widespread use of multiple representations requires understanding and formalizing the relationship between distinct families of solids, so that the respective objects and morphisms can be compared and associated as needed. In the first author's thesis [9] it was shown that the formal relationship between any two categories is captured by the notion of *functors* [13,16]. The precise role and application of functors in modeling parametric families remain to be studied, but their formal properties appear to hold the key to understanding many of the above challenges.

Acknowledgments

This research is supported in part by the National Science Foundation grants DMI-9522806, DMI-9900171, DMI-0115133, and CCR-0112758, General Motors Corporation, and EDS PLM Solutions. Vadim Shapiro gratefully acknowledges the support from the Department of Computer Science and Automation of the ''Roma Tre'' University, Italy that allowed him to continue research on this and other problems during his sabbatical visit.

References

[1] Pahl, G., and Beitz, W., 1984, *Engineering Design*, The Design Council, London.
[2] Shapiro, V., and Vossler, D. L., 1995, ''What is a Parametric Family of Sol-

ids?'' In 3rd ACM Symposium on Solid Modeling and Applications, Salt Lake City, Utah, May.
[3] Raghothama, S., and Shapiro, V., 1998, ''Boundary Representation Deformation in Parametric Solid Modeling,'' ACM Trans. Graphics, **17**(4), 259–286, October.
[4] Bidarra R., and Bronsvoort, W., 2000, ''On Families of Objects and their Semantics,'' In Proceedings of Geometric Modeling and Processing, Hong Kong. IEEE Computer Society Press, Los Alamitos, April.
[5] Stewart, N. F., 1992, ''Sufficient Condition for Correct Topological Form in Tolerance Specification,'' Comput.-Aided Des., **12**, 39–48.
[6] Pavel, M., 1989, *Fundamentals of Pattern Recognition*, Marcel Dekker Inc, New York.
[7] Sabin, M., and Shapiro, V., 1998, ''Modifications in Cellular Models,'' Technical report, NA-09, DAMTP, University of Cambridge, UK, September.
[8] Raghothama, S., and Shapiro, V., 2000, ''Consistent Updates in Dual Representation Systems,'' Comput.-Aided Des., **32**(8-9), 463–477, August.
[9] Raghothama, S., 2000, ''Models and Representations for Parametric Family of Parts,'' PhD thesis, Spatial Automation Laboratory, University of Wisconsin-Madison.
[10] Shapiro, V., 1997, ''Maintenance of Geometric Representations through Space Decompositions,'' International Journal of Computational Geometry and Applications, **7**(4), pp. 383–418.
[11] Requicha, A. A. G., 1980, ''Representations for Rigid Solids: Theory, Methods and Systems,'' ACM Comput. Surv., **12**, pp. 437–464, December.
[12] Requicha, A. A. G., 1977, ''Mathematical Models of Rigid Solid Objects,'' Technical report, TM-28, PAP, University of Rochester, Rochester, NY, November.
[13] Jänich, K., 1983, *Topology*, Springer-Verlag, New York.
[14] Borisovich, Y., Bliznyakov, N. Izrailevich, Y., and Fomenko, T., 1985, *Introduction to Topology*, Mir Publishers, Moscow.
[15] Maunder, C. R. F., 1996, *Algebraic Topology*, Dover Publications, New York.
[16] Munkres, James R., 1984, *Elements of Algebraic Topology*, Addison-Wesley, Reading, Massachusetts.
[17] Pierce, B., 1990, ''A Taste of Category Theory for Computer Scientists,'' Technical report, CMU-CS-90, Carnegie Mellon University, Pittsburgh, PA, March.
[18] ter Hofstede, A. H. M., Lippe, E., and Frederiks, P. J. M., 1996. ''Conceptual Data Modeling from a Categorical Perspective,'' Comput. J. (Switzerland), **39**(3), pp. 215–231.
[19] Munkres, James R., 1975, *Topology A First Course*, Prentice Hall, Englewood Cliffs, New Jersey.
[20] Requicha, A. A. G., 1977, ''Part and Assembly Description Languages: I—Dimensioning and Tolerancing,'' Technical report, TM-19, PAP, University of Rochester, Rochester, NY, May.
[21] Hoffmann, C. M., and Kim, K.-J., 2001, ''Towards valid parametric CAD models,'' Comput.-Aided Des., **33**, pp. 81–90.
[22] O'Connor, M. A., and Rossignac, J. R., 1990, SGC: A Dimension Independent Model for Pointsets with Internal Structures and Incomplete Boundaries, In IFIP/NSF Workshop on Geometric Modeling, Rensselaerville, NY, 1988, North-Holland, September.
[23] Rossignac, J. R., 1996, ''MAGISET: Architecture and Programming Interface for a Universal Modeler,'' In Proceedings of the Blaubeuren Workshop on Graphics and Modeling, Germany, June.
[24] Gomes A., and Teixeira, J., 1996, Modeling shape through a cellular representation scheme, In Modeling and Graphics in Science, J. Teixeira and J. Rix, eds., Springer-Verlag.
[25] Armstrong, C., Bowyer, A. A., Cameron, S., Corney, J., Jared, G., Martin, R., Middleditch, A., Sabin, M., Salmon, J., and Woodwark, J., 1997, ''Djinn: A Geometric Interface for Solid Modeling,'' Technical report, Information Geometers.
[26] Bidarra, R., and Bronsvoort, W., 2000, ''Semantic Feature Modeling,'' Comput.-Aided Des., **32**(3), pp. 201–225.
[27] Chen, X., and Hoffmann, C. M., 1995, ''Towards Feature Attachment,'' Comput.-Aided Des., **27**, pp. 675–702.
[28] Shapiro, V., and Vossler, D. L., 1991, ''Construction and Optimization of CSG Representations,'' Comput.-Aided Des., **23**(1), pp. 4–20, January.
[29] Lefchetz, S., 1942, ''Algebraic Topology,'' Am. Math. Soc. Trans., American Mathematical Society, Providence, RI.
[30] Shapiro, V., and Vossler, D. L., 1993, ''Separation for Boundary to CSG Conversion,'' ACM Trans. Graphics, **12**(1), pp. 35–55, January.
[31] Kinsey, Christine, 1993, *Topology of Surfaces*, Springer-Verlag, New York.
[32] ISO-STEP Part 42. 1998, ''Product Data Representation and Exchange,'' Technical report, ISO.