Issues and solutions in feature-based modelling: re-designing the shape kernel of CAD systems

A. Gomes, A. Middleditch and C. Reade
Brunel University,
Centre for Geometric Modelling and Design,
Uxbridge, Middlesex, West London UB8 3PH, England
Tel:+44 1895 274000 ext. 2699
Fax: +44 1895 812556
Email: abel.gomes@brunel.ac.uk

Abstract

Form features have been around since early 80's. However, until recently, no (mathematical) theory underpinning form feature-based modelling has been proposed. This is largely because of an incomplete understanding of the relationships between shape and function of engineering artifacts and the inadequacy of current geometric models to conform with the arbitrary space decompositions. This paper shows the essential shape incompatibilities between current geometric models and form feature models, and describes briefly an integrated and general-purpose shape model.

Keywords

Feature modelling, shape modelling, geometric modelling

1 INTRODUCTION

Integration is a concept that is not wholly achieved in current computer aided design and manufacturing (CAD/CAM) systems. Paraphrasing Warman (1990), in fact, a simple joining of systems is not the same as integration. Significant efforts have been carried out mainly in the last ten years to achieve a reference model for CAD systems in order to establish a general CAD framework capable of describing and formalising a generic CAD system architecture. A CAD system consists of a

G. Jacucci et al. (eds.), *Globalization of Manufacturing in the Digital Communications Era of the 21st Century*

network of processes or virtual machines cooperating with each other (Reference Model, 1990).

As specific virtual machines, geometric and feature-based modellers play a central role in design and manufacturing. They are distinct, although related, shape processing machines since they represent and manipulate distinct shape structures. Additionally, feature-based modellers focus on the function of the engineering artifacts shape, promoting the symbiosis between shape (usually geometry) and functionality. This enables, for example, the construction of complex models of matching mechanical components, called assemblies. But, more importantly, form features allow us to envisage a more effective integration of design and manufacturing languages and processes. However, to achieve an effective integration of design and manufacturing, we have first to integrate geometric and feature-based modellers. The sucess of such integration has been found to be extremely difficult.

Our methodology towards such an integration closely follows three guidelines, namely:

- Unlike the usual methodologies in feature-based modelling, we ***separate the shape and function aspects*** of form features; that is, we distinguish between functional entities, called form features, and their underlying shapes, called feature shapes. This is extremely important, because it enables us to distinguish and formulate a functional model and a shape model for form features, conforming with the principle that function determines the shape of engineering artifacts, but not vice-versa. The functional model carries engineering data, including form feature type (e.g. keyways, grooves, through holes, slots, bosses), assembly relationships (e.g. shaft-hole engagement F5-g6), type of material, surface finish, and even geometric shape (e.g. cylindrical shape or square shape), etc.
- The ***shapes underlying form features (feature shapes) do not depend on any functional or engineering requirements.*** This does not contradict the fact that function determines shape. We show that, the family of form features is potentially infinite, depending on the state of technology development, whereas the family of feature shapes is finite. Furthermore, this shape classification is general since it does not depend on the application, whether it is feature-based modelling or anything else. This shape classification follows from mathematical principles that are beyond the scope of this paper (see Gomes, 1998, for details). It is dimension-independent; for example, a circle and a torus in $\mathbf{R}^3$ both have a hole through them.
- ***Current geometric modellers (ACIS, Parasolid, etc.) cannot process directly the types of shape that underly form features***. Consequently, we say that current geometric modellers suffer from shape incompleteness. In fact, extra, external, application-oriented geometric data structures between the geometric modeller and the applications, feature modellers in particular, are absolutely necessary in order to find a reasonable degree of shape compatibility. This introduces not only significant geometric redundancy, but also forces the feature modeller to take control of all geometric interactions amongst form features. This promiscuity between the geometric and the feature modeller is

rather undesirable and encourages us to think of an alternative architecture of a shape modeller as a result from synthesising the geometric shape processed by geometric modellers and the shape associated with feature modellers. This releases feature modellers from any shape overheads, and leaves them responsible only for functional processing of engineering design and manufacture. Such shape integration should facilitate the integration of CAD/CAM systems.

2 SHAPE THEORY ESSENTIALS

This Section provides a summary of the fundamentals of the shape theory introduced in Gomes and Teixeira (1994), Gomes and Middleditch (1996), and detailed in Gomes (1998).

Geometric equivalence, geometric shapes

Two geometric objects in $\mathbf{R}^n$ are geometrically equivalent if and only if they are congruent, or, equivalently, they can be exactly superimposed on each other by a rigid motion (translations, rotations, and reflections). That is, two objects are geometrically equivalent if and only if their point sets coincide by superimposing one on the other. These rigid transformations or mappings preserve lengths (or distances, areas, volumes and angles. So, for example, two cubes with the same volume belong to the same equivalence class of congruent objects; similarly, all circles of radius five form an equivalence class of congruent objects. Obviously, *the number of distinct geometric objects in* $\mathbf{R}^n$ *is infinite* and is equal to the number of all subsets of $\mathbf{R}^n$. Some examples of non-congruent geometric objects are depicted in Figure 1; (a) a solid sphere with two local geometric shapes, the surface $x^2+y^2+z^2=1$ and the solid $x^2+y^2+z^2<1$; (b) a parallelipipedic surface containing 6 planar surface patches, 12 straight line segments, and 8 corners; (c) 1-dimensional geometric object consisting of 1 straight line segment, 1 circle line, and 2 corners.

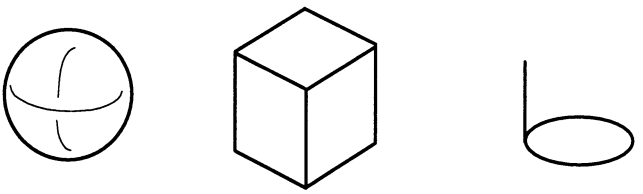

Figure 1. Objects with different geometric shapes.

Topological equivalence, topological shapes

The *topological shape* takes into account the global properties of polyhedra and neglects the local ones. Every topological shape is essentially *relaxed* in the sense that two objects are equivalent if only if they are homeomorphic. Thus, two objects are of the same topological shape if and only if they are homeomorphic. Intuitively, homeomorphisms behave like elastic transformations of subsets made

of perfectly elastic rubber. In other words, two subsets in $\mathbf{R}^n$ are topologically equivalent if and only if one subset can be made to coincide with the other by an elastic transformation. For example, a solid sphere can be deformed into a solid cube, or a cube with a protrusion, or even a cube with a depression. Thus, topological equivalence does not preserve geometric properties such as distances, angles, convexity, etc.

Equivalently, two objects possess the same topological shape if only if their first three Betti numbers are the same. These three Betti numbers stand for the number of components (or separated 'pieces') , holes through components, and voids in components, respectively. For example, the solid objects shown in Figure 2 are all topologically equivalent because they possess the same number of components (one), through holes (one), and internal cavities (zero).

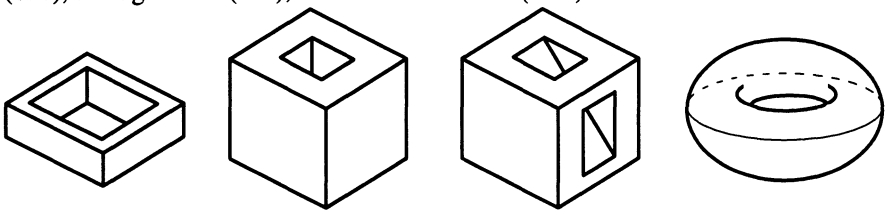

Figure 2. Objects with the same topological shape.

Vexitopy equivalence, vexitopic shapes

Roughly speaking, a vexitopy is a function which maps the convex and concave zones, called convexities and concavities, respectively, of an object to the convex and concave zones of another object, respectively. A vexitopy is a particular homeomorphism and a generalisation of an isometry. A vexitopy, unlike a homeomorphism, is an elastic mapping constrained by the fact that a convexity cannot be be transformed into a concavity, nor vice-versa. Gomes and Middleditch (1996) showed that every surface of a closed solid is decomposable into convex and concave zones, which are the shape representatives for surface form features in the feature modelling literature. Therefore, vexitopies are zonal shape mappings. Gomes and Middleditch also showed that these zones can be considered as 2-dimensional shape retracts of 3-dimensional *vexitopic (morphological) shapes*, called *protrusions and depressions*. Moreover, they have related the class of topological shapes with vexitopic shapes, using the following axioms:

- A protrusion is a convex point set;
- A depression is a convex point set;
- A component contains at least a protrusion; if the component is convex it is the single protrusion.
- A through hole contains at least one depression; if the through hole is convex it is the single depression.
- A void contains at least one depression; if the void is convex, it is the single depresion.

These axioms relating zonal and global shapes of an object imply a space decomposition not available in conventional geometric modellers. It also provides

the first mathematical shape model for feature modelling. In fact, a graph-based model is easily derived from the containment relationships between zonal and global shapes. An alternative and stronger shape theory results if the distinction between zonal and global properties is relaxed. The idea is to consider every topological shape as a vexitopic one; components are protrusions, through holes and voids are depressions, but now we have to consider that protrusions and depressions may or may not be convex. Consequently, every protrusion (respectively, depression) is decomposable into a finite collection of simpler protrusions (respectively, depressions). For example, in Figure 3, the through hole (a) is a depression consisting of 3 convex depressions, and the depression (b) has two convex depressions.

Figure 3. (a) Through hole with three convex depressions, (b) a depression with two convex depressions.

3 SHAPE INCOMPABILITIES IN MODELLING

The primitives of geometric modellers are a finite number of families of pre-defined equivalence classes of geometric objects, from which more complex geometric objects are somehow constructed.

3.1 Set-theoretic shape modelling

Set-theoretic modellers, usually known as CSG (Constructive Solid Geometry) modellers, typically include five parametric families of primitive solids (or homogeneously 3-dimensional geometric objects): blocks, wedges, spheres, cylinders, and tori. Each primitive solid is an element of a geometric equivalence class of congruent solids. Each primitive solid is parametrically defined by a finite number of parameter values. For example, a block or rectangular parallelipiped is created or instantiated by assigning values to its lenght l, width w, and height h; symbolically, this is described by the function CREATE_BLOCK:$\mathbf{R}\times\mathbf{R}\times\mathbf{R}\rightarrow\boldsymbol{B}$, where $\boldsymbol{B}$ is the family of blocks.

Typical CSG modellers do not provide a compatible shape interface for feature-based modelling. In fact, they do not provide access to through holes, protrusions, etc. For example, a subtraction of one CSG solid from another may or may not create a through hole, but there is no way to know that directly from the CSG tree. Besides, topological and morphological shapes such as, for example, a through hole composed of three depressions (Figure 3a), requires some kind of shape grouping machinery which is absent from CSG trees. Consequently, automatic (algorithmic) detection of feature shape changes is not possible; for example, the transmutation of the through hole (Figure 3a) into a stepped depression (Figure 3b)

consisting of two convex depressions, after removing the bottom convex depression.

3.2 B-Rep-based shape modelling

B-Rep modellers have some advantages over CSG modellers because they possess a lower level of shape incompleteness. In fact, they are able in principle to represent and manipulate *n*-cells (e.g. points, lines, surfaces and solids), whether they possess holes and voids. The existence of these cells in B-Rep data strucutures is quite useful for many purposes; for example, it facilitates the direct, graphical interaction with the designer.

B-Reps have explicit representatives, called shells, for components and voids. Shells are particular cellular clustering entities. Unfortunately, through holes, depressions and protrusions do not have similar representatives. Even worse, there is no any representation for the relationships between global topological shapes (components, through holes and voids) and their constituent morphological ones (protrusions and depressions). Equivalently, there are no hierarchical shape clustering entities, and it is impossible to represent cellular interactions between topological and morphological shapes.

To overcome these shape incompatibilities between B-Reps and F-Reps, some authors have proposed external cellular data structures to emulate cellular clustering for topological and morphological shapes, their shape hierarchical relationships, and their cellular interaction relationships. However, this external cellular shape structure on the top of a B-Rep data structure clearly exposes the shape defficiencies of the B-Rep modellers. In fact, it can not be considered as an extension of B-Rep because the corresponding mathematical model has not been reformulated and extended. It is just an *ad-hoc* solution for a particular engineering application, such as feature-based modelling.

4 A GENERAL SHAPE MODELLING KERNEL

Feature-based modelling shows us that current geometric CSG and B-Rep families are shape-incomplete. This is largely due to deficiences in the mathematical models for CSG and B-Reps. Such CSG and B-Rep models were derived from the theory of semialgebraic sets (Requicha, 1977) and theory of closed surfaces (Braid, Hillyard and Stroud, 1978), respectively, but they were over-constrained by the notion of solidity. Consequently, they could not satisfy important *shape requirements* of a significant number of applications, even those related to design and manufacturing.

4.1 Shape requirements for applications

The most important foundation for the effective integration of computer aided design and manufacturing systems is an engineering environment with **integrated shape**. Otherwise, we can never say CADCAM integration has been reached. The most important shape requirements follow.

General geometric coverage

In the last twenty years, some efforts to integrate the geometries of solid modellers and free-form modellers have been attempted by integrating their implicit and parametric representations. However, as Gomes (1998) shows in his doctoral work, an integrated geometry has to do more with a general geometric coverage than its representations. Such a general geometric coverage has been recently proposed by Middleditch, Reade and Gomes (1998) to be the class of subanalytic sets. Subanalytic geometry is a generalisation of algebraic geometry, rational geometry and transcendental geometry; hence, it includes algebraic and semialgebraic sets (e.g. CSG objects) described by polynomials, rational sets (e.g. nonuniform rational B-splines, shortly NURBS) described ratios of polynomials, and transcendental sets (e.g. springs) described by transcendental functions, respectively. This is not only important to overcome geometric incompatiblities in design and manufacturing, but also because sub-analytic sets mathematically form a Boolean class, i.e., they can be combined through Boolean operators; for example, a cylindrical hole in a spring can created by subtracting a cylinder from it.

General shape coverage

This paper shows that geometry is only part of the business in shape modelling. By shape integration we mean not only geometry integration, but also the integration of geometry with other kinds of shape, namely topological and morphological shape. Other shape types can be defined, but they are of minor importance in feature-based modelling. Geometry integration is far from complete but subanalytic sets appear to be able to fill the gap between solid modelling and freeform modelling. Furthermore, the integration of topological and morphological shapes in a function-independent general shape modeller releases feature modellers from controlling shape representation and manipulation. This is advantageous in many respects, mainly because shape modelling is then application-independent. A feature modeller will be then just a (functional) modelling system taking advantage of the facilities provided by a general-purpose shape modeller.

Multi-dimensional cell structure

The geometric structure of an object should be piecewise manifold and multi-dimensional. Amongst several reasons, we mention the following. First, it provides the unified representation for drafting, wireframe, surface, and solid models (Weiler, 1986), essential in interactive design. Second, it is necessary to model objects and spaces of arbitrary dimension, in applications such as robot path planning which uses n-dimensional configuration spaces (Middleditch, 1992). Third, it is suited to the representation of finite element meshes, unlike solid models which do not possess internal membranes.

General cellular clustering

With the exception of the C-Rep (Cellular Representation) developed by Gomes and Teixeira (1994), current geometric cellular models do not provide general

dimensionally nonhomogeneous clusters for cells. This fact and the non-existence of a general shape theory have been the major obstacles in the way of integrating geometric and feature modellers. Such clusters are called subcomplexes in the theory of complexes (see, for example, Rotman, 1988). Basically, the classical hierarchical cellular structure of complex–cell is replaced by a hierachical cellular clustering structure of complex–subcomplex–cell. A subcomplex can represent the cell structure of a topological or morphological shape, or even general shapes necessarily non-homogeneous in dimension.

Unlike C-Rep, other cellular models use external, application-oriented cellular clustering as it is the case of those due to Floriani and Falcidieno (1988), Luo and Lukács (1991, Gomes, Bidarra and Teixeira (1992), Masuda (1993), Rossignac (1997), or Bidarra and Bronsvoort (1998). These external cell clusters were specifically designed to represent the cell structures of form features, that is, they are application-oriented.

4.2 Basic architecture of a general shape modeller

In generic terms, the architecture of a general shape modeller includes the following two layers (Figure 4).

- **Structure** layer. The two-level cellular structure (complex-cell) of the B-Reps should be replaced by a three-level cellular clustering structure (complex-subcomplex-cell). Cells are manifolds, that is, local topological shapes as used in classical B-Reps. Subcomplexes are arbitrary clusters of cells, and obviously, a complex is a cluster of subcomplexes.
- **Shape** layer. Here we have the geometric (points, lines, surfaces and solids described by sub-analytic functions), global topological (components, through holes, and voids), and morphological shapes (protrusions and depressions). The axiomatic relationships between global topological and morphological shapes are represented by an hierarchical graph.

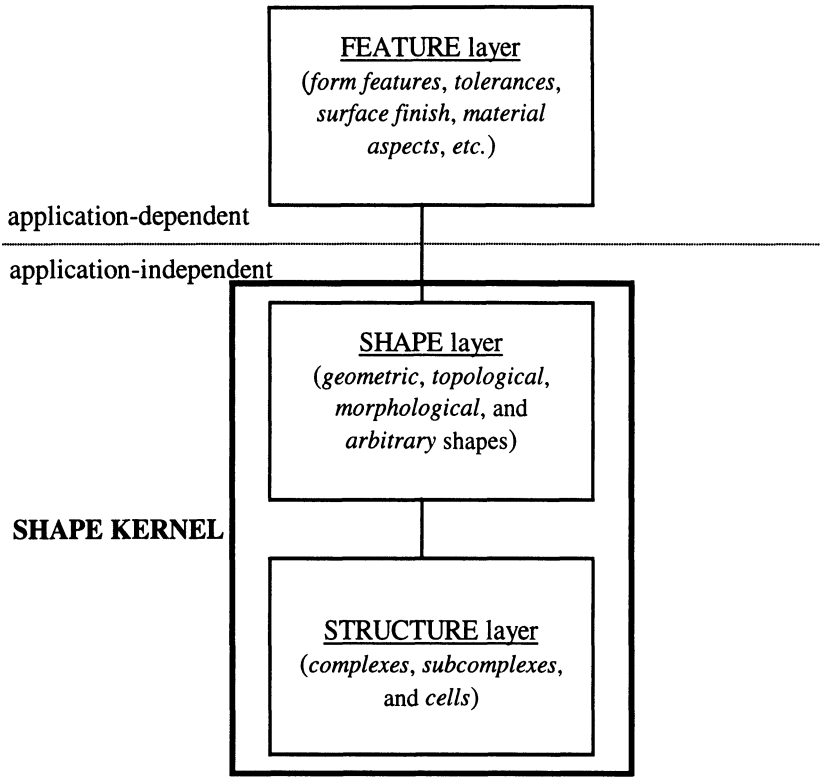

Figure 4. The diagram of a complete shape CAD kernel.

Note that the each shape is one-to-one mapped to a cell or a subcomplex, but not all subcomplexes correspond to a shape in the shape layer; for example, the subcomplexes of interaction between shapes have no representatives on the shape layer, but they may be useful for many purposes beyond the discussion in this paper.

This shape kernel is general-purpose since it is application-independent, whether or not it is included in a CAD system. It is necessary to bear in mind that some engineering terms used in feature modelling coincide with mathematical terms in shape theory; for example, a through hole is a form feature but also a topological shape. The distinction becomes apparent when we consider that there so many kinds of through holes (straight holes, stepped holes, etc.) in mechanical engineering design, but they all have the same topological shape, just a through hole.

5 CONCLUSIONS

The main conclusions to draw from the above discussion are:

- A shape theory capable of synthesising the shape of geometric and feature modellers has been outlined. It enables effective shape integration of CAD/CAM systems.
- Features combine function with shape, but the shape processing should be delegated to an application-independent shape kernel.
- In feature-based modelling, the separation of function from shape and structure provides a deeper insight of what should be a CAD kernel. In terms of object-oriented software engineering, a subtractive form feature class (e.g. SLOT, POCKET, etc.) is derived from a shape class (DEPRESSION, respectively), which in turn is derived from the SUBCOMPLEX class. Obviously this turns such a shape kernel into an extensible object-oriented system.
- Current geometric modellers are over-constrained since the relationships between shape and structure have not been completely understood and developed so far. Consequently, cellular clustering is usually relegated and controlled by applications externally. To avoid external, application-dependent cellular clustering, subcomplexes must be part of the structure.

6 REFERENCES

Bidarra, R. and Bronsvoort, W. (1998) Validity maintenance of models with interacting features. In Proceedings of the 8th Portuguese Meeting on Computer Graphics (ed. A. Sousa and J. Teixeira), Coimbra, Portugal.

Braid, I., Hillyard, R. and Stroud, I (1978) Stepwise construction of polyhedra in geometric modelling, CAD Group Document 100, University of Cambridge, Computer Laboratory, England.

Floriani, L. and Falcidieno, B. (1988) A hierarchical boundary model for solid object representation. *ACM Transactions on Graphics*, **7**(1).

Gomes, A. (1998) A shape model through combinatorial sub-analytic Whitney stratifications, PhD Thesis (in preparation), Brunel University, England.

Gomes, A. and Middleditch, A. (1996) Synthesis of a unified approach to shape modelling, in *Geometry Modeling: Theory and Practice* (ed. W. Strasser, R. Klein and R. Rau), Springer-Verlag.

Gomes, A., Bidarra, R. and Teixeira, J. (1992) A cellular approach for feature-based modelling, in *Graphics Modeling and Visualization in Science and Technology* (ed. M. Göbel and J. Teixeira), Contributions to Computer Graphics Series, Springer-Verlag.

Gomes, A. and Teixeira, J. (1994) Modelling shape through a cellular representation scheme. In *Modelling and Graphics in Science and Technology* (ed. J. Teixeira and J. Rix), Contributions to Computer Graphics Series, Springer-Verlag.

Luo, Y. and Lukács, G. (1991) A boundary representation for form features and non-manifold solid objects. In Proceedings of the Symposium *on Solid*

Modeling Foundations and CAD/CAM Applications (ed. J. Rossignac and J. Turner), ACM Press.

Masuda, H. (1993) Topological operators and boolean operations for complex-based nonmanifold geometric models. *Computer Aided Design*, **25**(2).

Middleditch, A. (1992) Cellular models of mixed dimension. Technical Report BRU/CAE/92:3, Brunel University, England.

Middleditch, A., Reade, C. and Gomes, A. (1998) A formal basis for the objects of the Djinn API to a geometric modelling kernel (submitted).

Reference model for CAD systems (English version), Gesellschaft für Informatik e.V. (1991).

Requicha, A. (1977) Mathematical models of rigid solid objects. Production Automation Project, Technical Memorandum 29, University of Rochester, USA.

Rossignac, J. (1997) Structured topological complexes: a feature-based API for non-manifold topologies. In Proceedings of the Symposium on *Solid Modeling and Applications* (ed. C. Hoffmann and W. Bronsvoort), ACM Press.

Rotman, J. (1988) *An introduction to algebraic topology*. Graduate Texts in Mathematics, Springer-Verlag.

Warman, E. (1990) Integration revisited: an appraisal of the state of the integration of CAD. *Computers in Industry*, **14**(1-3).

Weiler, K. (1986) Topological structures for geometric modeling. PhD Thesis, Rensselaer Polytechnic Institute, New York, USA.

7 BIOGRAPHY

Abel Gomes took a *licenciatura* (five-years degree) in electronics engineering and a MSc (PAPCC) in computer graphics at Coimbra University, Portugal. He is now a PhD student at Brunel University, working on a unified shape theory for solid modelling, free-form modelling and form feature-based modelling.

Alan Middleditch obatined a BSc (Tech) in electrical engineering and a Msc in digital systems at Manchester University, England. After that, he obtained MS and PhD degrees at Carnegie-Mellon University, USA. In the early 70s, he became a research fellow at the University of Rochester where he was involved with the design of the first set-theoretic solid modelling system. He is now a professor of geometric computing at Brunel University, England, continuing research into the application of solid modelling to computer aided engineering problems. His current research includes the development of a representation-independent API for geometric modelling, constraint based modelling, algebraic free-form surfaces and mesh generation.

Chris Reade has a degree in mathematics, and MSc and PhD in computer science all from London University, England. He is a lecturer at Brunel University, England. His research interests include functional programming, programming language theory, as well as geometric modelling. He is the author of a book published by Addison-Wesley entitled *Elements of Functional Programming*.