AN ALTERNATIVE GOAL-ORIENTED HIERARCHICAL REPRESENTATION OF SOLID OBJECTS FOR COMPUTER-INTEGRATED MANUFACTURING

Leila De Floriani
Istituto per la Matematica Applicata del CNR
16132 Genova, Italy

George Nagy
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

ABSTRACT

The Face-to-Face Composition (FFC) graph is the formal representation of a family of models of solid objects for advanced engineering applications. It is a multi-rooted hierarchical structure based on boundary representation and is capable of accommodating different conceptual views of the same object or assembly. A node of the FFC graph describes a volumetric object component consisting of a single shell, while its arcs correspond to connection faces between pairs of single components. Operators are available both for modifying the object and for changing the representation. Node groups define a correct order of evaluation of the FFC graph which produces a valid solid object at each step. The model also includes a formal definition of the notion "feature-of" as an open subgraph corresponding to constructs not necessarily realizable on their own, such as rivet holes with pads.

1. INTRODUCTION

Objects must be represented at many stages in the design and manufacturing cycle, and the manner in which the same object is considered at various stages depends on the goal associated with each step. For example, the designer is primarily interested in functional characteristics and groupings of components. The production engineer has to consider the stock from which components are generated by material removal and the most economical sequence of machining, assembly, fastening, and finishing operations. The view of the supporting data base also changes from stage to stage: for design, it needs to be organized according to functional characteristics, while for production planning, inventory and manufacturing resources play the most important role.

Our principal objective is to define methods, formalisms, operators and data structures to support the development of CAD/CAM systems sufficiently flexible to accommodate the above requirements. Our ideas are rooted in prior work on structured graphs and solid models at the Institute of Applied Mathematics in Genova, and are guided by the experience and requirements of the industry-supported Computer Integrated Manufacturing (CIM) project at Rensselaer Polytechnic Institute. Our current focus is to impose the following requirements on boundary representation:

(1) Hierarchical organization of the parts constituting the final object.

(2) User-specified or algorithmically-developed alternative hierarchical representations for the same object. Convenient means of transforming one valid representation into another.

(3) User-specified level of abstraction (granularity) of representation. The size of the elementary components considered as a single item should be variable.

(4) Composition and decomposition of objects both in terms of "real" faces (juxtaposed, welded, glued objects), and "virtual" faces (temporary divisions established for the convenience of the user). Extension of the real/virtual dichotomy to edges and vertices in order to provide greater flexibility in surface and edge treatment.

(5) Meaningful definition of the concept of "feature" as the modification of an existing structure. Provision for "features" and "feature libraries" consisting of physically non-realizable components such as holes that become meaningful only when applied to an appropriate support structure.

(6) Provision for locality of operations. Minor modifications should be relatively independent of the overall size of the model.

(7) Flexibility with regard to both the internal structure of the elementary components and to the external interfaces of the model (for example, to graphics display and numerical control modules).

(8) Computational and storage efficiency.

(9) Provision for imbedding the necessary operators in user-friendly application-specific languages without affecting the basic structure of the model.

In our approach, a single <u>object</u> is described by means of a <u>Face-to-Face Composition</u> (FFC) graph. Each node of the graph describes the topology and geometry of a physically realizable, single-shell <u>component</u> (a <u>shell</u> is any maximal connected set of faces on the bounding surface of an object).

CH2555-1/88/0000/1101$01.00 © 1988 IEEE

Each arc in the FFC graph corresponds to a connection face between two components. Two components may share more than one connection face, resulting in multiple arcs between the corresponding nodes.

Transformations of the FFC graph are changes in the representation of a given object. Transformation operations are MERGE and SPLIT: they combine two nodes into one or split one node into two. The results of these operations are also valid, single-shell components. Elementary components are constructed through Euler operators. Models of more complex objects may also be obtained by combining models of simpler objects: the COMPOSE operator combines two objects represented by distinct FFC graphs, while DECOMPOSE separates them. Although the FFC model is quite general, we are currently considering its application only to polyhedral objects.

Node groups are collections of FFC nodes which must be evaluated before the FFC graph to which they belong can be evaluated. Node groups allow a correct order of evaluation of the FFC graph representation of a given object. Features, as defined in the FFC model, need not correspond to a valid volumetric component. Within an FFC graph, one may consider a feature together with its support, which must be a valid object. Thus, a feature must be combined with a support in order to guarantee a valid configuration for the whole. Since geometrically similar instances of the same component may be used to construct an object, the nodes of the FFC graph correspond to instances of elementary or structured components stored in a component data base.

The classical survey of solid models, including both Constructive Solid Geometry (CSG) and Boundary representation (Brep), is [12]. As a preliminary to developing non-manifold topological representations, Weiler provides a useful taxonomy in terms of boundary or volume, object or spatially based, and evaluated or unevaluated representations [16]. Requirements for defining unique boundary representations are discussed in [15]. The winged-edge data structure for describing the boundary of polyhedra was first demonstrated in [2]. Alternative boundary data structures are proposed in [1,18]. Approaches to extracting manufacturing information from solid modelers are presented in [17,9,5,7], while the integration of process constraints in the design and description of mechanical parts is shown in [8]. The problem of classifying form features in design and manufacturing is discussed in [11]. The application of hierarchical graph-based models to solid modeling is discussed in [5]. The RPI Computer Integrated Manufacturing program, which provided the impetus to apply these techniques to automated manufacturing, is described in the proceedings of the Third Annual CIM Conference [13].

2. THE FFC MODEL FOR AN OBJECT

The Face-to-Face Composition (FFC) graph of an object is a directed acyclic multigraph. Each node represents a valid single-shell volumetric component. Arcs between the nodes correspond to pairs of perfectly abutting connection faces, as discussed below. If an object consists of disconnected, non-contiguous components, then these components correspond to different connected components of the FFC graph. However, a single connected component of the FFC graph can describe an object consisting of multiple shells, provided that such shells are physically juxtaposed.

Single nodes are internally described according to one of the accepted boundary models for single-manifold objects [1,2,16,18]. The definition of the FFC graph is independent of the particular model chosen to represent individual components. The FFC model is therefore modular in the sense that any geometric or topological modification of a single component that does not affect its connection entities is local to that particular component.

Each node has one or more parents, except for an (arbitrary) set of root-nodes called the base of the FFC graph. The base may be the baseplate of an assembly on which everything rests, or the largest component, or any other component chosen as the starting point. The resulting hierarchy defines a valid partial order for constructing the object starting at the base by successive addition or subtraction operations. Holes and depressions may be created by subtracting one component from another. Whether addition or subtraction is performed at a given step is indicated by the sign of the corresponding arc. The sequence of operations required to construct a mortise-and-tenon joint from stock is shown schematically in figure 1. The tenon is obtained by removing components C and D from stock E. The mortise is made by removing B from A. The mortise has five connection faces with the tenon, indicated by the five arcs with a positive sign.

2.1 Connection entities

At an abstract level, each component of the FFC graph can be viewed as the collection of its connection faces, which define the interface of such a component. Connection loops, edges and vertices of a single component of the FFC graph are attached to their connection faces. In general, connection entities occur wherever components are joined, juxtaposed, parted or separated. Each connection entity must be either real or virtual. Real connection entities are preserved when the entire object is reconstructed from its FFC graph representation. Virtual connection entities simply disappear in reconstruction: they serve as a conceptual aid to the user to define the reconstruction process.

An arc in the FFC graph which connects two components C1 and C2 defines a correspondence between a connection face f1 of C1 and a connection face f2 of C2. Faces f1 and f2 will have the same topological and geometric structure. Hence, such an arc is completely identified by the

4-tuple (C1, C2, f1, f2) if directed from C1 to C2, or by (C2, C1, f2, f1) if directed from C2 to C1. Each arc (Ci, Cj, f', f") has a sign which denotes whether Cj is added or subtracted from Ci, and an attribute (real or virtual) which indicates whether f' and f" are real or virtual. The manner in which connection faces are joined - welded, glued, soldered, or press fit - may be indicated also as an attribute attached to the corresponding arc of the FFC graph.

Real connection edges occur whenever it is desired to preserve a transition in the surface characteristic of a face after reconstruction. Real connection vertices connect real connection edges. Virtual connection faces can be bounded by either real or virtual edges. In the planar-faced object environment that we consider here, virtual edges can occur only when the real faces bounded by the edges are coplanar. Virtual vertices connect a set of edges of which at least one is real.

A connection face f in a component C defines a downward (upward) connection if there exists in the FFC graph an arc incident from (to) node C corresponding to f in C. A connected subgraph of the FFC graph is called upward (downward) open if the set of the upward (downward) connection faces of the root (leaf) nodes of the subgraph is not empty. Otherwise, it is upward (downward) closed.

2.2 Node groups

Node groups are introduced in order to define a correct order in the evaluation of the FFC graph representation of an object. An FFC graph is evaluated by constructing an explicit boundary model of the object it represents.

A node group is a collection of nodes of the FFC graph G which define a connected subgraph of G describing an admissible three-dimensional object. A node group composed of a single FFC node is called an elementary node group. Node groups are created in the design process when an object S' is subtracted from another object S". The FFC graphs G' and G" of S' and S" define two node groups in the FFC graph G of the resulting object S. The creation of node groups is strictly dependent on the sequence of operations performed in object design. Thus, there may exist several admissible decompositions (into node groups) of the same FFC graph representing a given object.

A node group must be evaluated as a single component before the FFC graph to which it belongs can be evaluated. Two adjacent components can be merged together to form a valid volumetric component only if they belong to the same node group. Node groups can be either nested or disjoint. The evaluation of an FFC graph is thus a recursive process which takes place from the inside structure of node groups. Figure 2 shows a decomposition of the FFC graph G of figure 1 into five node groups. G has four elementary node groups.

3. A FAMILY OF FFC MODELS FOR AN OBJECT

Our scenario is that the designer will design an object by preparing a "Design FFC" which reflects a functional view of the object. The manufacturing or process engineer will then manipulate this graph (or the equivalent single-level Brep) to obtain the "Production FFC". The quality-control person may, instead, transform the graph in a manner suitable to emphasize visual inspection. Accordingly, we define operations that allow altering both the structure of the FFC graph and the nature of the entities considered as "elementary" components. These transformations are accomplished using two operators: MERGE and SPLIT, which are a generalization of the refinement and abstraction transformations defined in [5].

MERGE (G, C1, C2, C3) merges nodes C1 and C2 of the FFC graph G and forms a new node C3. In this process, the connection faces corresponding to the arcs of G joining C1 and C2 are merged pairwise and eliminated as a consequence. Their virtual bounding edges and vertices are also eliminated. Only components with virtual connection faces can be merged, since otherwise the resulting component would have more than one shell. Only pairs of nodes belonging to the same node group or forming two elementary node groups can be merged together. MERGE is an irreversible operation, since the connection entities are not retained.

SPLIT (G, C3, F, (C1, C2)) splits component C3 of the FFC graph G at the faces specified by the set F and generates two separate components C1 and C2 having virtual connection faces corresponding to the faces in F. A set of arcs from C1 to C2 are added to G, with each arc corresponding to a pair of the virtual connection faces in C1 and C2 defined by SPLIT. When the SPLIT corresponds to subtracting C2 from C1, then C1 and C2 form two separate (elementary) node groups. SPLIT is a reversible transformation, but requires the complete geometric specification of the connection faces of the components resulting from the operation.

MERGE and SPLIT are illustrated in figure 3. In order to consider the mortise as a single component, A and B must be merged to form a new component F. Only the node groups involved in the transfomation are indicated. Neither operation modifies the expanded boundary description of the object represented by the FFC graph, but affects only its hierarchical description.

4. CONSTRUCTION OF THE FFC MODEL OF AN OBJECT

An FFC graph of an object can be constructed by building its single components separately and then combining them by successive pairwise composition of distinct FFC graphs. A node in the FFC graph contains the "flat" boundary description of a component and thus can be constructed through application of so-called Euler operators which guarantee the topological validity of the resulting component. Sets of edge-oriented Euler operators which assume a winged-edge representation of the object boundary are described in [6,10]. A complete set of face-based Euler operators is presented in [1]. For specific applications it may also be desirable to define macro-operators from the basic Euler operators. With each primitive topological entity (face, edge, vertex) we must associate appropriate geometric descriptors. A basic choice is whether to restrict the model to polyhedral objects, in which case curved entitites must be represented by a piecewise-linear approximation, or to allow parametric representations of surfaces and curves [3]. In our model the adjacency topology and the geometry are well separated and in principle we could readily accomodate the parametric representation of sculptured surfaces. Geometric validity checks are, of course, considerably more complex for such surfaces.

The construction of complex objects from distinct models of simpler objects is accomplished through the COMPOSE operator. The inverse operation is performed through the DECOMPOSE operator.

COMPOSE (G1, G2, F, G) combines two objects S1 and S2 described by the FFC graphs G1 and G2 at the specified set F of matching faces and produces an FFC graph G describing the resulting object. G1 and G2 become subgraphs of G. F is a collection of 4-tuples of the form (Ci, f', Cj, f"), where Ci and Cj are components of G1 and G2 respectively and f' and f" define a pair of matching connection faces belonging to Ci and Cj respectively. When two separate objects are combined, their relative position must be specified. They need not abut: objects consisting of components that do not touch are perfectly acceptable. The connection faces are determined by geometric intersection before the composition. If S2 is subtracted from S1, then two node groups described by G1 and G2 are created in order to ensure a correct evaluation of the resulting object S. The operation is irreversible because the faces and nodes where the combination took place are not marked.

DECOMPOSE (G, G1, G2) decomposes the object represented by an FFC graph G into two objects represented by two disjoint and connected subgraphs G1 and G2 of G respectively. The decomposition is defined by the (real) connection entities common to G1 and G2. G1 and G2 must describe two valid objects. The arcs connecting nodes of G1 and of G2 must be deleted and the corresponding connection faces eventually transformed into single faces. DECOMPOSE does not alter the position of the resulting objects. There cannot exist any node group in G which contains both G1 and G2. The operation is irreversible.

COMPOSE and DECOMPOSE are illustrated in figure 4: two distinct FFC graphs are combined into the FFC graph G that represents the object in figure 1. Since, in this case, the composition corresponds to an addition, node groups are not indicated for simplicity.

5. FEATURES AND INSTANCING

The word "feature" has been used in many different ways in CAD/CAM. One definition reported in [16] is:

A feature is a geometric or non-geometric form or entity:
1) whose presence or dimensions are relevant for one or more computer-integrated manufacturing functions;
2) whose availability to the designer facilitates the design process;
and
3) feature primitives are often specified by the needs of a given process domain.

Another definition reported in [11] states that a form feature is a "region of interest on the surface of a part."

We have chosen a more restrictive model-dependent definition, which incorporates the semantic notion ("feature-of") that a feature has no existence on its own and is meaningful only as it is applied to some other entity. In our model, a feature is any construct (not necessarily a valid volumetric component) defined by an upward-open FFC graph. Figure 5 shows an L-shaped bracket feature defined by the subgraph formed by nodes C and D and the arc connecting them.

An upward-open connected subgraph of an FFC graph defines a feature with respect to any downward-open subgraph G' of G to which it is connected and which describes a valid object with a minimal number of FFC nodes; the object described by G' is called support of the feature. The support of the feature shown in figure 5 is the subgraph formed by nodes A and B and the arc connecting them.

A feature library would contain only upward-open subgraphs, together with a specification of the family of supports to which the feature can be attached. Features can be specific, containing a complete geometric description, or generic, where one or more dimensions are left unspecified.

Consequently, a feature is defined not only by its shape, but also by the connection faces through which it can be applied to an object. For example, an L-shaped bracket feature would have connection faces corresponding only to its two external mounting surfaces, as shown in figure 5.

A related aspect of the model under development is "instancing". When several geometrically similar instances of a component are used to construct an object, only the elementary description of the component is stored in a component data base. The nodes of the FFC graph now contain pointers to the data base as well as transformation matrices that specify the position, orientation and scale of the component for each occurrence. Furthermore, the data base may contain also components, called structured components, which are themselves represented as FFC graphs.

6. SUMMARY

We have described the principles for a proposed family of boundary models of solid objects based on current theoretical developments and experimental CIM programs. The simplicity, hierarchical nature and enhanced representational power of the model are intended to facilitate the integration of design, manufacturing, and quality control operations in a highly automated environment. Novel aspects of the model include:

1. The imposition, either by the user or by an algorithm, of an arbitrary but valid partial order on object components. We believe that hierarchical boundary models combine many of the advantages of constructive solid geometry and of boundary representation.

2. The definition of real and virtual connection entitites, which facilitates the representation of joining and cutting operations (including material removal from stock), assembly sequences, and conceptual decomposition of objects.

3. Flexibility in the representation of single-shell components with a well-defined interface with respect to the FFC graph.

4. A small set of primitive operations defined independently of the application and the user interface.

5. A workable model-dependent definition of form features which allows creation of data bases containing useful, but not necessarily independently physically realizable constructs.

We now propose to experiment with the model by interfacing a data structure that allows manipulation of the FFC graph with (1) an existing computer-aided drafting system for interactive input, (2) an existing 3-D computer graphics package for perspective or isometric output, (3) an existing 2-D computer graphics package for interactive manipulation and display of the underlying graphs; (4) an existing computer-vision inspection system that provides the final step in an actual application.

ACKNOWLEDGMENT

The authors are indebted to Mr. A. Maulik, RPI-ECSE, for suggestions on drafts of this paper and for preparing the figures. Professor Les Gerhardt, Dr. Mark Steiner and Mr. Charles Malloch of the RPI Computer Integrated Manufacturing Program contributed valuable comments.

REFERENCES

[1] Ansaldi, S. De Floriani, L., Falcidieno, B., Geometric modeling of solid objects by using a face adjacency graph representation, Computer Graphics 19, 3, 1985, pp. 131-139.

[2] Baumgardt, B., Winged-edge polyhedron representation, Stanford AI Project CS 320, October 1970.

[3] Braid, I.C., Geometric modeling, in: Advances in Computer Graphics I (G. Enderle, M. Grave and F.Lillehagen, eds.), 1986.

[4] De Floriani, L., A graph-based approach to object feature recognition, Proc. Third ACM Symposium on Computational Geometry, Waterloo, June 1987, pp.100-109.

[5] De Floriani, L., Falcidieno, B., A hierarchical boundary model for solid object representation, ACM Trans. on Graphics 7, 1, 1988 (in print).

[6] Eastman, C., Weiler, K., Geometric modeling using Euler operators, Proc. First Conference on Computer Graphics in CAD/CAM systems, MIT, April 1979, pp. 248-254.

[7] Falcidieno, B., Giannini, F., Extraction and organization of form features into a structured boundary model, Proc. Eurographics '87, Amsterdam, September 1987, pp.249-269.

[8] Henderson, M.R., Extraction of feature information from three dimensional CAD data, Purdue University PhD Dissertation, Lafayette, IN, 1984.

[9] Jared, G.E., Shape features in geometric modeling,in: Solid Modeling by Computers: From Theory to Applications, (M.S. Pickett and J.W. Boyse, eds.), Plenum Press, New York 1984.

[10] Mantyla, M., Sulonen, R., GWB: A solid modeler with Euler operators, IEEE Computer Graphics and Applications 2, 1982., pp. 17-32.

[11] Pratt, M., Wilson, P.R., Requirements for support of form features in a solid modeling system, Tech. Rep. CAM-I, R-85-ASPP-01, 1985.

[12] Requicha, A.A.G., Representation of rigid solids: theory, methods and systems, Computing Surveys 12, 1980, pp. 437-464.

[13] RPI, Proc. Annual CIM Conference, Troy, NY, December 1986.

[14] RPI, Center for Manufacturing Productivity, CIM Program Progress Report, First Quarter 1987.

[15] Silva, C., Alternative definitions of faces in boundary representation of solid objects, Tech. Memo no. 36, Production Automation Process, University of Rochester, 1981.

[16] Weiler, K., Topological structures for geometrical modeling, Rensselaer Polytechnic Institute PhD Dissertation, Troy, NY, 1986.

[17] Woo, T.C., Feature extraction by volume decomposition, Proc. Conf. on CAD/CAM Technology in Mechanical Engineering, MIT, 1982.

[18] Woo, T.C., A combinatorial analysis of boundary data structure schemata, IEEE Computer Graphics and Applications 5, 3, 1985, pp. 19-27.

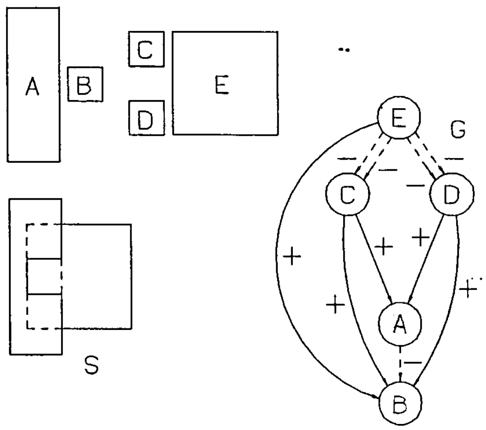

Figure 1. FFC graph representation of a mortise-and-tenon joint.

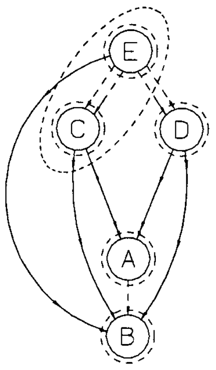

Figure 2. Node Groups.

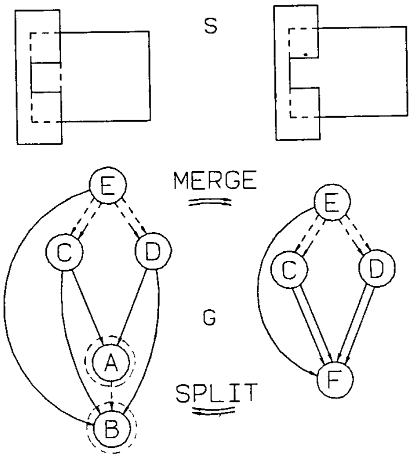

Figure 3. MERGE and SPLIT.

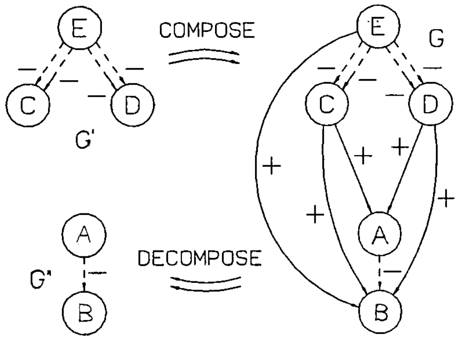

Figure 4. COMPOSE and DECOMPOSE.

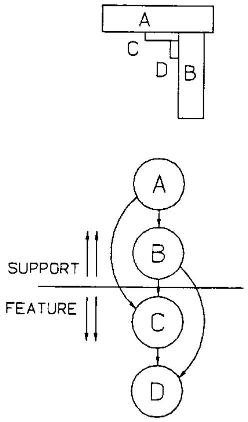

Figure 5. An L-shaped bracket feature.