

# Geometric modeling architecture for manufacturing-process simulation software

*M. Gallaher and P. Athirajan*

*Concurrent Technologies Corporation,  
1450 Scalp Avenue, Johnstown, PA, USA*

*Phone: 814-269-2635, 814-269-2654*

*Fax: 814-269-2666, e-mail: gallaher@ctc.com, rajan@ctc.com*

## **Abstract**

Geometric modeling capabilities are an integral part of any computer-aided design system. Most of these capabilities focus on the final product but do not consider other information needed for supporting downstream applications. Additional product- and process- related information needs to be added to the geometric model. This includes information on functional requirements, process geometry, process planning, engineering analysis, and the specification of analysis attributes to the geometric entities. The National Applied Software Engineering Center (NASEC) is applying object-oriented technology to develop an extensible software architecture for manufacturing-process simulation software that encapsulates this information. This will ensure seamless integration of the geometry model with the numerical model and, as a consequence, will greatly reduce the time for setting up the process simulation. This paper addresses the geometry modeling constructs supported by this architecture, and specifically, the design of an object-oriented, feature-rich, geometric modeling architecture layered on top of the ACIS™ Geometric Modeler. The advantages of the design are also addressed.

## **Keywords**

CAD, object-oriented design, geometric modeling, process simulation, software architecture, object modeling

## 1 INTRODUCTION

A geometric model<sup>1</sup> is the computerized representation of a given spatial domain. Geometric modeling capabilities are an integral part of any computer-aided design (CAD) system. Most of these capabilities focus on the final product but do not consider other information needed for supporting downstream areas such as engineering analysis and manufacturing process simulations. Engineering-analysis and process-simulation software packages apply numerical analysis techniques to the pertinent physical equations and to an appropriate idealization of the geometry model (that is, the 'numerical' model). In order to directly use the CAD's geometry model to support a manufacturing-process simulation or general numerical analysis scheme, additional product- and process-related information must be added to the model. This information represents the manufacturing domain and completely defines the situation in mathematical physics. The information includes functional requirements, process geometry, process planning, and engineering analysis attributes.

This paper addresses software architecture areas associated with (i) the geometric representation of objects that are essential to simulating manufacturing processes, such as casting, forging, and powder metallurgy and (ii) the seamless integration of the geometry model with the numerical model.

In addition, the use of object-oriented technology in geometric modeling is discussed. The high-level design of an object-oriented, feature-rich, geometric modeling architecture layered on top of the ACIS™ Geometric Modeler is presented and the advantages of the design are also explored.

## 2 GEOMETRIC MODELING FOR PROCESS SIMULATION

Manufacturing involves complex processes that transform raw materials into useful products. Constant pressure to improve quality while at the same time reduce costs and lead times, coupled with recent scientific and technological developments, provide many challenges to U.S. manufacturing base. Process modeling and simulation offer the best approach to meeting these challenges and achieving manufacturing agility (CTC, 1994). In all commercial manufacturing-process simulation software, geometry plays an important role. In recent decades, a considerable amount of geometry-based software has been developed to support modeling and simulation of manufacturing processes.

Curiously though, integration of the primary models for driving the analysis-geometric model and numerical model-has not changed significantly in recent decades. This is due in part to the manufacturing industry's traditional sequential positioning of process planning after product design. This demarcation between product design, process design, and ultimately manufacturing, usually means that the specification of process-related information is included only in the 'numerical model' used in the analysis and not tied to the 'geometric model' passed on to manufacturing. The basic objective of the geometric model should be an unambiguous, unique, and complete representation of an object or part so that the model

---

<sup>1</sup> For the purpose of this paper, the use of geometric model and CAD model will be considered interchangeable.

data can be used in all engineering functions (Zeid, 1991). In addition, the adequacy of the geometric model for process simulation software's analysis scheme is determined by its associated attributes. The deficiencies with the models' underlying data structures and the separation of product design from process design are beginning to retard the geometric modeling objective and the progress of automating software in the manufacturing arena. Furthermore, the growing trend toward using automatic mesh generators makes it desirable to truly automate the pre-processing activities and drive the analysis solely from the geometric model. For this purpose, loads, boundary conditions, body fluxes, material properties, and other related physical parameters will have to be appended to the geometric model (Shepard, 1995).

## 2.1 CAD Model Interface to Manufacturing Process Simulations

Clearly, the central part of an analysis model is the mesh, which is based on the geometry of the object. The geometry is usually represented initially by a CAD model. This model should be used directly for developing the mesh. This can be achieved with the following three approaches, as shown in Figure 1 (McMahon, 1993):

1. The CAD system interfaces to a process simulation pre-processor that is capable of receiving geometric data from the CAD system via an input file;
2. The CAD system may itself incorporate a mesh generator and directly generate numerical models from the object's geometry; or
3. A compromise of the previous two: a mesh is generated by a CAD system and the process simulation modeler adds the needed analysis attributes (material data, loads, body fluxes, and boundary conditions).

At first glance, all of these would seem to be viable alternatives and each has been implemented to some degree. In addition, compared to the first approach, the second and third approaches appear to be far superior in that they take full advantage of the existing geometric descriptions provided by the CAD model. In practice, however, there are a number of glaring limitations:

- Commercial and customized process-simulation software have more advanced pre-processing facilities than those provided by CAD applications.
- There is no universally accepted (used) format for supporting data exchange. The Initial Graphics Exchange Specification (IGES) is not sufficient and the Standard for the Exchange of Product Model Data (STEP) is not yet ready for broad use.
- Mesh generation in CAD applications is usually not sufficiently robust for analyzing a manufacturing process.
- Inconsistent and antiquated data structures between CAD systems and manufacturing-process simulation software systems does not permit interoperability of the numerical model. This results in unnecessary and complex integration steps to adapt one system's data structure to the other's.
- CAD applications do not give strong enough consideration to the process that is actually producing the product. While a limited set of product features (such as hole, slot, and boss) are generally found, process features (such as workpiece, mold, cavity, die, and punch) are rarely included.

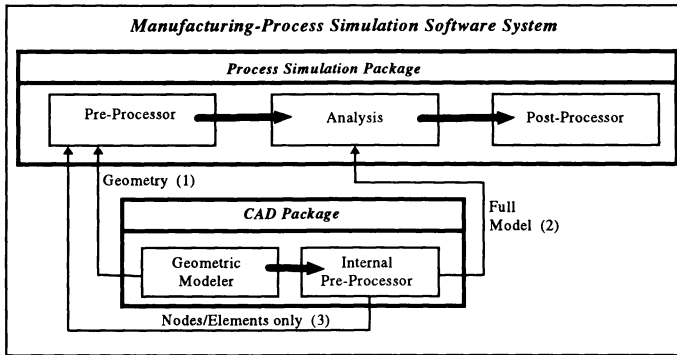


Figure 1. Approaches to Numerical Model Generation.

### 3 GEOMETRIC MODELING

Boundary representation (B-rep) and constructive solid geometry (CSG) are the two most common approaches for the geometric representation of solid objects. Until recently, geometric modeling systems normally operated with dual representations (B-rep/CSG) of the solid body being modeled. These systems were marred by the difficulty in converting from B-rep to CSG (the reverse, while still complex, is relatively straightforward). This, along with the improved performance in display generation and the flexibility in the forms that can be modeled, has led to the current B-rep domination of commercial solid modeling packages. In addition, the facilities for boolean operations of solids and CSG style input are generally provided by B-rep systems.

Our approach to providing geometric modeling capabilities for the manufacturing-process simulation software is based on:

- Augmenting traditional geometric building blocks with both product- and process-related features. This will enable next-generation pre-processors to totally represent the geometry from both product and process perspectives.
- Providing flexible, intelligent, abstract geometric-modeling data structures. The size of the set of features needed to support all manufacturing processes is staggering, therefore, generalized abstractions will be needed.
- Using a widely accepted commercial solid modeling kernel as the underlying infrastructure for data structures and features. This was dictated by a desire for an early implementation of the software and by the complexity of these kernels.
- Including non-manifold modeling capabilities, as opposed to the manifold modeling approach employed by many solid models.
- Driving the engineering analysis process from the CAD model, which is the same model used for manufacturing product and process procedures.

Because of the importance our architecture places on features and the setup process, it is necessary to provide dual representation by incorporating both CSG and B-rep aspects. The architecture provides a feature-based paradigm for models constructed by combining 'features' characteristic of the manufacturing domain. The constructive representation of

complex features is compact and can be generated rapidly (given a robust set of primitive features). This is the primary factor favoring a feature-based modeling approach. While there is a performance penalty associated with generating the B-rep from a constructive specification, the benefits are substantial. The model's global topological and geometric features can be evaluated precisely and a visual display readily created from the B-rep. The architecture addresses both process-independent (abstract) and process-specific features. This combination will facilitate constructing complex models that would be time-consuming to produce using either CSG or B-rep techniques independently.

### 3.1 Rationale for Product and Process Geometry

Geometric descriptions of the product and of the manufacturing-process equipment and tooling are required as inputs in any manufacturing-process simulation. Process geometry consists of the geometry of the workpiece at various stages throughout the process and the geometries of the equipment and the tooling used in the manufacturing processes. The product geometry is a geometric representation of the product in its final form.

The workpiece represents the raw material in its pre-processed state (for example, billet or slug). The tooling for a process includes the components that interact with the workpiece. These interactions can be generalized into the following areas: shape control, material flow control; and thermal control. Surprisingly, most process simulation software deals with the workpiece only and does not give strong consideration to the process-specific tooling. As a consequence, most process-specific information is captured indirectly or lost.

#### *Simulation Setup Complexity*

Another issue in manufacturing-process simulation software deals with the complexity of the setup and expertise required to *run* simulations. A long setup time for complex parts is not uncommon, for example, six months for a truck differential casting (British Science and Engineering Research Council, 1993). Most existing simulation software requires an end-user to have deep knowledge in the underlying numerical techniques, input/output formats, process physics, and the peculiarities of the specific package being used.

This situation is best handled by adopting a feature-based approach to geometric modeling. The geometric features that form the basis for modeling are a mixture of industry-standard product features (such as through-holes, slots, and fillets) and process-based features (such as casting mold, cavity, riser, and gates).

#### *Process Features*

Introducing process features addresses the issue of setup complexity by providing a more natural way of geometrically representing the process. Through feature-based representation, process features associated with manufacturing processes can be specified, leading to:

- enhanced understanding of product geometry;
- reduced level of finite element expertise needed to run and set up a simulation;
- increased flexibility to produce custom simulations tailored to specific processes;
- supporting automatic generation of numerical models from the geometry model.

A feature is an aspect of an item that is meaningful to a domain expert and is usually dealt with as a unit. Feature-based representations of geometric data contain knowledge at a higher level of abstraction than an ordinary B-rep or CSG model. In this context, process features can be referred to as recurring patterns of information associated with the process-related geometry description.

A process feature is defined as a collection of related geometric or functional elements that correspond to a particular manufacturing-process simulation application. The process features are useful in understanding the function, behavior, or performance of the manufacturing-process simulation applications.<sup>2</sup> Examples of process features defined for sand casting include pouring cup, down sprue, runner, and riser.

Characteristics of process features include:

- representation of the features and corresponding link to underlying geometry;
- associated technological data of the features such as position and orientation, geometric tolerances, and material properties;
- semantic relationship to other features;
- viewpoint from which the feature is defined (such as design, analysis, assembly, and functional);
- link to a particular manufacturing process.

Generic process features will be characterized for reusability between specific, but similar, types of processes.

### 3.2 Object-Oriented Technology

Using object-oriented technology to develop a geometric modeling architecture is a logical means for expressing real-world models. The resulting design is easier to maintain and extend to many manufacturing-process simulations. Object-oriented technology is being used in the analysis, design, and implementation of the underlying data structures as well as in the organization of features in hierarchies. Using intelligent, abstract data structures allows less emphasis on the individual feature's data and procedural structures and more on the specification of the features themselves, the roles they perform, how they relate to each other, and how they automate aspects of the manufacturing setup process.

#### *Object-Oriented Frameworks*

Using object-oriented frameworks allow decomposing the geometric domain based on the underlying framework of the manufacturing process itself, rather than the ill-posed functional requirements of a single problem.

An object-oriented framework captures a reusable software design that supports common capabilities within a specific problem domain. It also provides a language-dependent implementation. When used in the design and implementation of complex geometric modeling modules<sup>3</sup>, object-oriented framework technology has advantages such as the following over many traditional approaches:

<sup>2</sup> This definition is derived from the manufacturing feature definition of Browne (McMahon, 1993).

<sup>3</sup> Module: A group of collaborative modeling constructs (classes, associations, aggregations, generalizations). We treat a module as a component that is analyzed, designed, implemented, and tested as a unit (Appley, 1996).

- enables the development of geometry-based modules that are driven from process abstractions that match the real world, simplifying setup and expanding the potential user base;
- makes it feasible to produce custom product and process features that are tailored to a specific process or user;
- allows for the straightforward implementation of an extendible abstract interface layered on top of a geometric modeling kernel (such as ACIS, Parasolid, SHAPES);
- reduces development time by ensuring that similar software does not have to be designed and implemented over and over again;
- improves the quality of the resulting modules, making maintenance and extensions easier. This capability of a module to accommodate its own incremental growth over time is not easily achieved by traditional approaches.

### 3.3 Geometric Modeling Architecture Development

A tailored version of object modeling technique (OMT) proposed by Rumbaugh, et al. (Rumbaugh, 1991) was used to develop the architecture for the geometric modeling aspects of a manufacturing-process simulation software system. To build complex software systems using this technique, the developer must abstract different views (static, dynamic, and functional) of the systems, build models using accepted notational constructs, ensure that the models satisfy the requirements of the system, and gradually add details to transform the models into implementation.

#### *Object Model Development*

In constructing the object model, it is essential to capture those concepts from the real world that are important to an application. Hence, while creating an object model for a geometry module, object classes have been conceived and their relationships have been established from the real-world geometry domain. The object model is represented graphically with object diagrams. Classes<sup>4</sup> are arranged into hierarchies that share common structure and behavior and are associated with other classes. Classes define the attribute types carried by each object and the operations each object undergoes. Figure 2 is the object diagram for a high-level description of the geometry module.

The geometry of a part can be modeled either in 2D profile or 3D solid. A 2D profile is internally composed of 2D entities (such as line, spline, circle, arc, ellipse, blend, and polygon) as shown in Figure 3. A 3D solid is an aggregated form of features. The features can be classified as primary features or secondary features. A primary feature defines the shape of a base geometry on which additional geometric features are built. A primary feature can exist as a stand-alone feature. For example, in an automotive piston, the primary feature would be a simple cylinder on which other features (such as holes and grooves) are added. A primary feature (Figure 4), can be either a primitive solid (such as block, cylinder, or prism) or it can be generated from sweeping operations (such as revolution, extrusion, skinning, and nonlinear sweeping). The sweeping operations require 2D profiles to create faces that are swept later to create solids.

---

<sup>4</sup> Classes and object classes are synonymous.

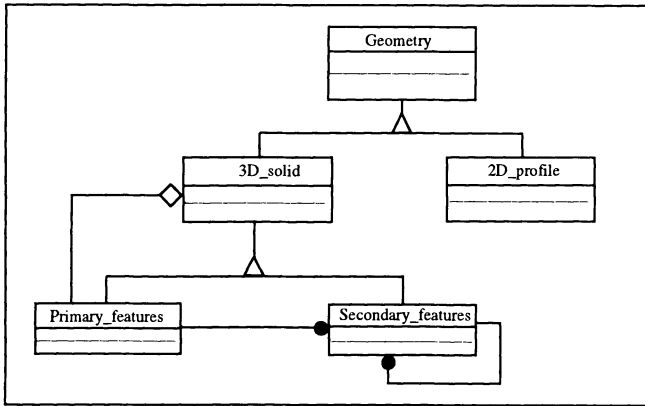


Figure 2. An Object Diagram for the Geometry Module.

The secondary features define the additive and subtractive volume that supplement the primary feature for a complete description of part geometry. Secondary features cannot exist independent of a primary feature. They can be categorized as positive volume features (such as boss, rib, fillet, flange, and fin) and negative volume features (such as hole, slot, cut, pocket, chamber, and blend).

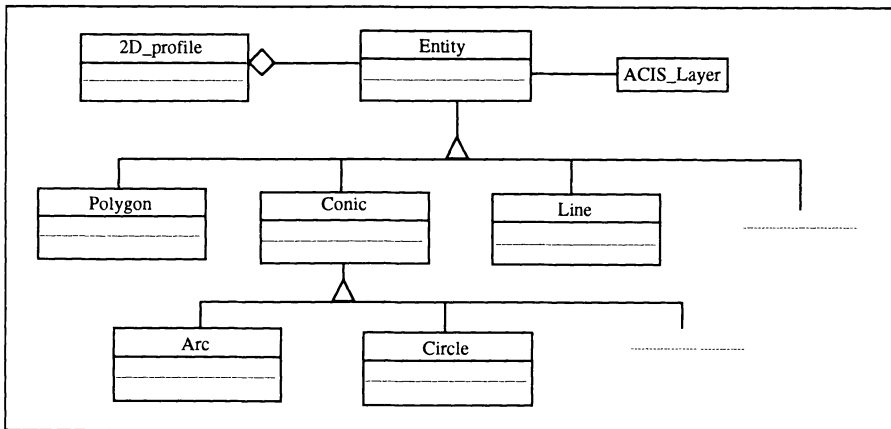


Figure 3. An Object Diagram for 2D Profile.

Object classes have been created for the process features related to manufacturing processes such as forging, casting, and powder metallurgy. An object model has been developed describing the various classes, their relationship, attributes, and methods.



Figure 5 depicts one such object model for the sand casting process. The software modules developed using this architecture will enable the user to create process-related geometry quickly.

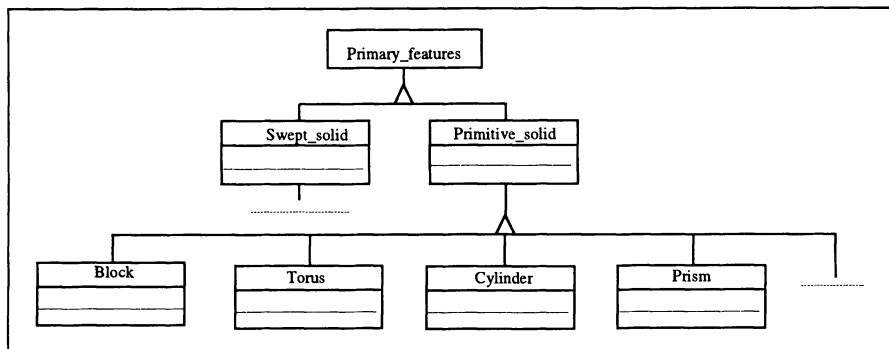


Figure 4. An Object Diagram for Primary Features.

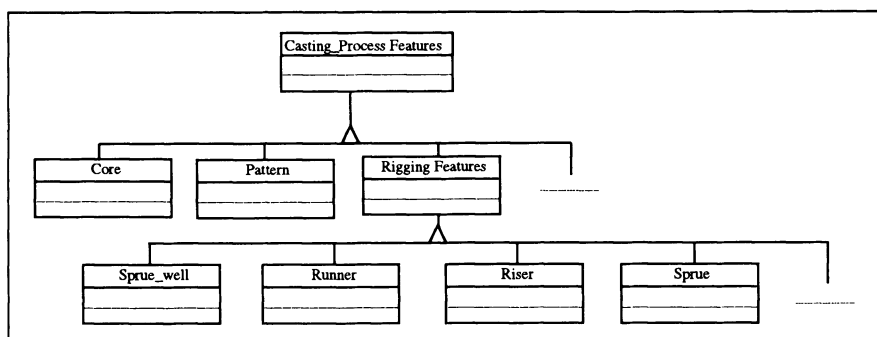


Figure 5. An Object Diagram for Casting Process Features.

### *Dynamic Model Development*

The dynamic model describes those aspects of a software system concerned with the time and sequencing of operations (Rumbaugh, 1991). It is represented graphically with state diagrams. The pattern of events, states, and state transitions for a given class can be abstracted and represented in a state diagram. Each state diagram shows the state and event sequences permitted in a system for one class of object; events represent the external stimuli and states represent the values of the object. A state is represented in a rounded box with an optional name; a transition is drawn as an arrow from the receiving state to the target state; and the name on the arrow is the name of the event causing the transition. An object model describes the possible patterns of objects, attributes, and links that can exist in a system. The attribute values and links held by an object are called its state. The objects stimulate

each other, resulting in a series of changes to their states. An individual stimulus from one object to another is an event. The dynamic model consists of multiple state diagrams (one state diagram for each class with important dynamic behavior) and shows the pattern of activity for an entire system.

State diagrams were developed for each class with meaningful dynamic behavior. All the state diagrams combined constitute the dynamic model for the geometry module.

## 4 FUTURE CHALLENGES

### 4.1 Analysis Attributes

Analysis attributes are physical quantities that are added to the geometric model to fully qualify the physical problem. Analysis attributes include loads, boundary conditions, body fluxes, and material properties. Because all the attributes are physical quantities, these attributes are specified and controlled by dealing with them as instances of a parameterized class differing only in their dimensional formulae and numeric type (scalar, vector, tensor). This definition scheme gives the desired attribute independent definition but does little for associating the attributes to the geometry model. Applying attributes to the geometric entities within a CAD model is currently being addressed.

### 4.2 Data Exchange Standards Between Geometric and Numeric Modeling

The benefits of standardizing data exchange have led to major standardization efforts backed by the government and industry. The two major data exchange efforts, IGES (early effort) and STEP (major effort under way now), are still very much in progress. IGES focuses on basic geometric model information and, despite its limitations, remains the dominant standard for geometry data exchange. STEP promises to provide complete support for all aspects of product modeling. Part 104 of STEP is an application model concerned with finite element analysis. This part is mainly a linear elastic FEM-based specification. In addition, STEP application protocols (APs) related to manufacturing processes (casting AP 223 and forging AP 229) are being developed. The STEP geometric modeling workgroup is also working with the modeling workgroup for finite element analysis in the specification of a geometry-based generic analysis model. The generic analysis model provides a unique (discretization-independent) specification of the association of analysis attributes with geometric entities. This model is also immature but the fact that it's a geometry-based specification promises to provide an improved basis for standards activities between geometric and numeric modeling. The extent to which these standards are realized remains a challenge.

### 4.3 Coverage of Manufacturing Features

As mentioned previously, the sheer number of unique manufacturing process-related features is staggering. Feature abstractions are being built that directly parallel product and

process entities in manufacturing. Choosing the right set of specific product and process features as well as the abstractions that represent these features will be a challenging design problem. These feature-based abstractions are the key architectural element of the geometry module.

#### 4.4 Acceptance from the Community

Object-oriented technology provides a fundamentally different way to develop a geometric modeling architecture. The differences stem not only from a pure software development viewpoint, but also because of the potential to drive downstream engineering analysis from the original CAD model. Many developers of simulation software have a significant investment in the way they presently develop software, and they will need to be convinced of the value of this approach. Performance is also important, notably in simulations that offer high-fidelity solutions where run-times of days or weeks are possible.

These issues are being addressed by teaming developers and users of manufacturing-process simulation software, as well as by building real demonstration applications. These demonstrations emphasize solving unique problems that cannot be solved with existing geometric models.

As for performance, the ability to use both product and process features will shorten setup times, which has a much greater impact on the overall problem-solution life cycle than actual run-time. Performance is also kept in mind throughout the module development process, and a highly modular/modifiable design will allow less efficient components to be swapped for more efficient ones (Appley, 1996).

## 5 REFERENCES

- Appley, G. and Gallaher, M. (May 1996) An Object-Oriented Framework for Manufacturing-Process Simulation Software, *Object Magazine*, 33-41.
- British Science and Engineering Research Council (Oct. 1993), Modeling Materials Processing, Report produced for the Department of Engineering, Cambridge University, CUED/C-MATS/TR.206.
- CTC Technical Report (1994) Manufacturing Software Architecture and Application System Concept Document.
- McMahon, C. and Browne, J. (1993) *CAD/CAM From Principles to Practice*, Addison Wesley, MA.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenzen, W. (1991) *Object-Oriented Modeling and Design*, Prentice Hall, NJ.
- Shepard, M. (1995) Integration of Finite Element Analysis into Design Practice, short course presented by MALLETT Technologies, University of Maryland, MD.
- Zeid, I. (1991) *CAD/CAM Theory and Practice*, McGraw-Hill, Inc.