Proceedings of the 15th International Conference on Manufacturing Systems – ICMaS
Published by Editura Academiei Române, ISSN 1842-3183

“Politehnica” University of Bucharest, Machine and Manufacturing Systems Department
Bucharest, Romania, 26–27 October, 2006

MODELING TOOLS IN THE DESIGN PROCESS

Cristian PISARCIUC

Abstract: *This paper summarizes the evolution of modeling technology, and provides a status report on solid modeling. It discusses some issues that indicate how little we really know about design and about the interplay between design and manufacturing. Contemporary modeling systems are most useful for refining and documenting nearly finished designs and for driving a growing array of computer aided manufacturing modules but they provide little help in the early, conceptual stages of design.*

Key words: *design, modeling.*

1. INTRODUCTION

Design is the first major step in a product’s life cycle, and design is often the main determinant of a product’s manufacturability, saleability, serviceability, and longevity.

Fig. 1 shows an idealized product cycle. Design begins with an indication by a customer that a new product is needed. Sales and marketing define a new or revised product in terms of functional requirements, price or volume trade-offs, and other similar parameters. At conceptual design time, the desired performance parameters (*e.g.*, weight, size etc.) are only loosely defined. Producibility parameters (*e.g.*, testability, process stability) are even less well defined, and supportability parameters (reliability and repairability) are least defined. Design and engineering convert the set of perceived needs and market constraints into complete specifications, a design, for a deliverable product.

Manufacturing planners then produce specifications for the product’s manufacture (typically process and inspection plans, numerical control programs etc.), and these are executed to produce a product that is then marketed.

Design is often the pivotal operation in the product cycle because it establishes a match (a compromise) between the initial marketing goals and a product’s deliverable functionality, economic producibility, maintainability, and longevity. Obviously, the design potentials of individual companies are strong determinants of their long-term viability in a competitive world.

Although a multitude of commercial, cultural, and historical factors influences the companies design capabilities, the primary intrinsic determinants are the skills of the designers and the tools and methods that they use. The principal focus in this paper is on tools, specifically modeling tools, because these are understood well enough to admit technical assessment and forecasting. While design methods and designers’ skills are at least as important as tools, they are poorly understood and are covered only briefly and somewhat indirectly.

Computer-aided design (CAD) and computer-aided manufacturing (CAM) systems have proliferated in the mechanical industries over the past decades, and within each lies a modeling system of some kind. Although advances in computing and graphics technology paced the early progress in CAD and CAM, progress in the past decade has been paced mainly by advances in modeling and in understanding of how to use models.

2. MECHANICALLY ORIENTED MODELING SYSTEMS

Contemporary modeling systems are concerned primarily with geometry. They provide means for defining the shapes of components and sometimes allowable shape variations, for positioning component representations to define assemblies, for calculating properties (appearance, mass, etc.), and, when linked to CAM modules, for generating manufacturing process data such as NC programs.

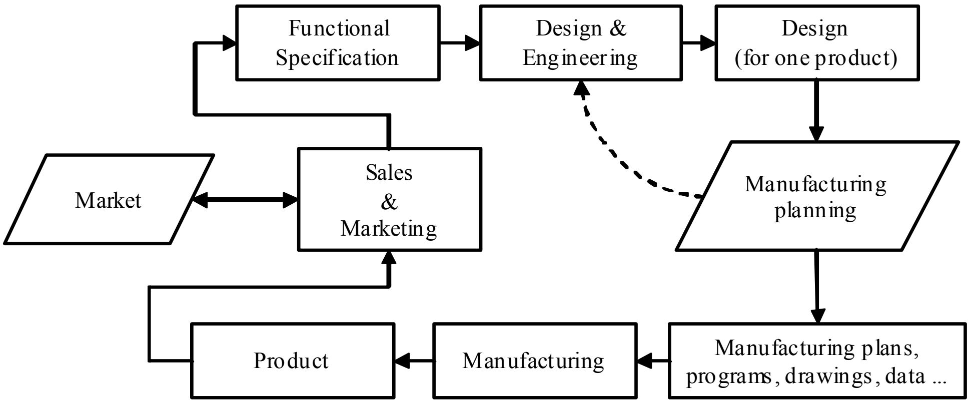

Fig. 1. An idealized product cycle.

In current industrial practice, four coupled bodies of information define a finished design for a product:

1. Ideal-form (shape) specifications for the component parts,
2. Associated variational specifications (tolerances) (these first two items taken together are equivalent to detail drawings),
3. Component combination specifications (assembly drawings),
4. Material and finish specifications.

Performance specifications rank as collateral information or as part of the design process documentation [1]. They cannot be part of the design definition unless consistency with the four components can be guaranteed, in which case performance specifications are redundant (because they are derivable from the design definition). Manufacturing and assembly process specifications are not included in the design definition.

Current object modeling theory and technology can handle items 1, 3, and 4, at least in principle (*i.e.*, subject to the geometric coverage, complexity, etc., limits already noted.) but item 2, tolerances, remain a problem area.

These systems can be analysed in terms of the generic geometry system shown in Fig. 2 [2]. Representations (models) of objects are built from definitional data supplied by users, and procedures are evoked by user commands to compute properties and do other useful work. The users may be humans, as is almost universally the case in design, or programs, increasingly the norm in manufacturing applications, where modeling systems are used as utilities by programs that simulate the motion of robots, check the correctness of NC programs, and so forth. The effectiveness of systems of the type shown in Fig. 2 is set mainly by the intrinsic power of the internal representation schemes, what can be represented, and with what fidelity, and by the procedures that can be deployed to calculate useful results. Nearly all of the early systems carried ambiguous representations that required human interpretation to be useful, whereas the solid modeling systems carry unambiguous representations that permit many calculations to be automated, at least in theoretically.

Solid modeling is well known by the use of valid and unambiguous representations of solids and because of that will replace forever wireframe technology. Solid modeling has the potential to support the automation of almost all conventional technical tasks done in industry, from detailed strength analyses through graphic rendering to the automatic planning of machining and assembly operations and the programming of tools to do the work. The most two frequently used schemes for solids modelling are: boundary representations (b-reps), in which solids are represented by sets of faces that enclose them completely, and constructive solid geometry (CSG), in which solids are represented as Boolean combinations (unions, differences, and intersections) of simple primitive solids.

Four other unambiguous schemes for representing solids are known and used, often in conjunction with those presented above, for certain kinds of applications:

– *Spatial Enumeration.* A solid is represented (usually approximated) as a union of quasi-disjoint box- shaped cells filled with matter. The cells may be of uniform size or of varying sizes if generated by recursive binary spatial subdivision. Enumerations of the latter type may be organized as logical trees, called quadtrees in two dimensions and octrees in three dimensions.

– *Cell Decompositions.* A solid is again represented as a union of quasi-disjoint cells, but now each cell may have a distinctive shape, if it is homeomorphic to a sphere. Triangulations are the simplest form of cell decomposition, and finite element meshes are the most widely used engineering materialization.

– *Sweeping.* A solid is represented as the spatial region traversed (swept-out) by either an area or a solid moving on a spatial trajectory. Although sweeping is central to modeling motional processes such as machining and robotic assembly, there are many open mathematical and computational questions surrounding it.

– *Primitive Instancing.* This is a formalization of the family of parts concept. A solid is represented as a particular member of a family, for example the family of single-diameter round shafts with oil grooves, by supplying appropriate numerical parameters to a family-specific collection of formulas for displaying members of the family, calculating their mass properties, and so on.

There are many areas where 3D solids are useful but they are used mostly for automatic finite-element analysis (FEA) or NC machining. There are several approaches to the problem of FEA, with one of the most promising being a two-stage process using quadtree or octree enumeration [3] to mesh the interior of a solid, followed by boundary traversal to extend the interior mesh to the surface of the part based on diagram presented in Fig. 3 [4].

The solid modeler shown in Fig. 3 is delivering part geometry and, through an attribute facility, loading and boundary conditions. The modeler also generates the

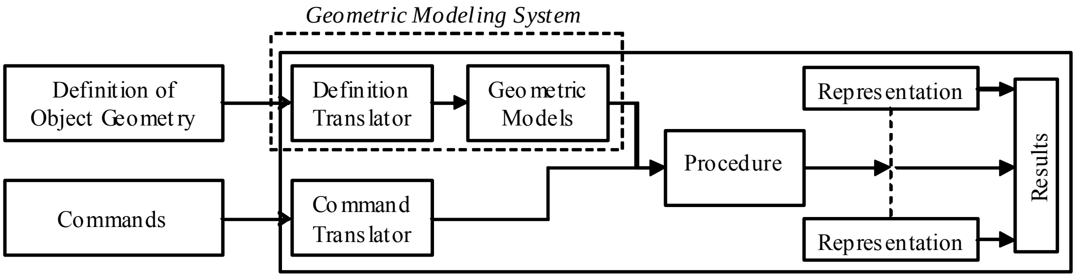

Fig. 2. A generic geometric modeling system.

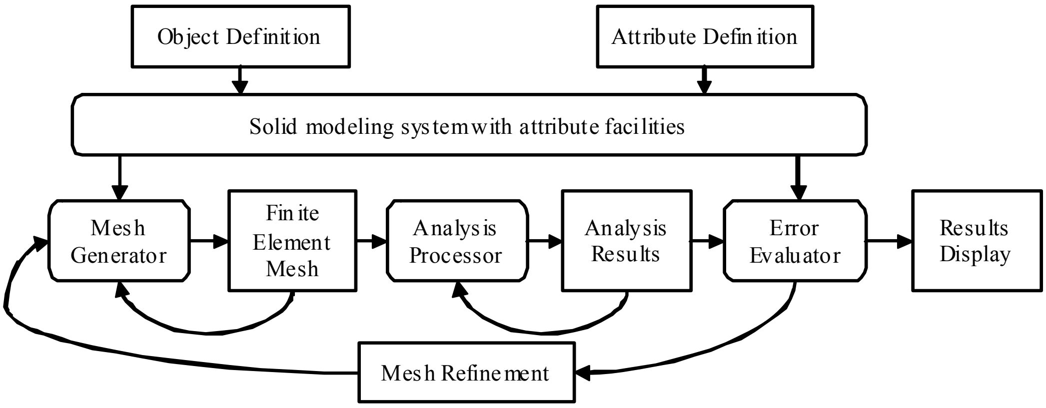

Fig. 3. A 2D automatic finite-element analysis system.

quadtree or octree approximations used in the meshing procedure, plus other aids for managing the process.

NC machining simulation or NC program generation can be easily done for a given a solid. The driving relation is:

$$W_i = W_{i-1} - V_i, \tag{1}$$

where: W_i is the workpiece after (simulated) execution of the *i* NC command; V_i is the spatial region swept by the cutter on the *i* command.

Consequently, a simple simulator reads an NC program block by block and displays the workpiece after each command, and user watches the displays and tries to spot problems (collisions, invasive machining, etc.).

Automatic NC program verification seeks to do two things: detect problems without recourse to human observers, and determine automatically whether the final machined part W is identical to the desired part P. The latter goal, attainment test (W = P ?), is easy to do in a solid modeler in principle, but there are computational subtleties. Automatic problem detection is done by applying two different kinds of tests at each stage of a simulation. Spatial problems are detected by various intersection tests, $P \cap V_i$ being the relation for testing invasive machining.

The detection of technological problems, such as cutter breakage or violation of tolerance constraints, mainly requires force calculations that are done indirectly. For example, $R_i = W_{i-1} \cap W_i$ is the solid actually removed (made into chips) by the i command, and the volume of R_i can be calculated automatically by a modeler's mass-property module. From this and the known cutting conditions, such as path length, and feed rate, an average material removal rate can be calculated. From the removal rate and other data, it is possible to estimate the average forces on the cutter and hence.

3. USER ENVIRONMENT AND INTERFACES

Early CAD/CAM systems were designed to be electronic drafting boards. T square, compass, and triangle were replaced with pointing devices (cursor, mouse, etc.) and command menus whereby users could create lines, circles, arcs, free-form curves, and text. Users could establish relations between elements of a drawing, for example, making one element parallel, perpendicular, or tangent to another and could copy, rotate, translate, save, and delete entities. These drafting interfaces came to be highly engineered, convenient, and fast as computer graphic technology advanced, but they enforced almost no model-based discipline on the user. These systems could be used to draw anything, because there were no underlying mathematical models of any object of higher order than curves.

When wireframe systems appeared, drafting interfaces, generally, were extended rather than redesigned, to exploit the mathematical rules governing wireframe structures.

The advent of solid modeling forced serious thought to be given to the design of user interfaces for several reasons. First, many solid modelers emerged from the research laboratories with command language interfaces rather than graphic interfaces. As a result, there was an interface design to do, since engineers often resist programming and insist on graphics. Second, solid geometry is usually created in chunks or whole blocks and cylinders, rather than through lower order lines and arcs. Therefore, the highly engineered drafting interfaces became largely irrelevant. Finally, solid modeling requires three-dimensional thinking and visualization skills; thus, 3D displays (perspective line drawings and shaded images) are almost essential, because defining entities in three dimensions is more difficult than in two dimensions, and working through two-dimensional views is not often the best approach. Contemporary solid modelers have solid-oriented graphic interfaces. These are, in essence, graphic versions of simple command languages that permit primitive solids to be instantiated from menus, positioned through rigid motions and coordinate system declarations, and combined through Boolean operations. Many systems also provide means for extruding and swinging closed planar contours into translational or rotational symmetric solids. The newest interfaces offer other features such as countersunk holes and various kinds of slots and pockets as definitional primitives. They also offer relational facilities that would, for example, allow a user to put face A of solid B against face C of solid D (matting) and supports constrained design, wherein critical parameters of parts are found automatically by solving systems of equations.

As noted earlier, programs for automatic finite-element analysis, machining simulation, and others also use modeling systems. As a result, programs are likely to be the major users within a decade [5]. Formal languages are the appropriate interfaces for such programs also for humans who wish to design parametrically. Languages are becoming highly developed for modelers with CSG input facilities (because representational validity is easy to guarantee in CSG).

4. CONCLUSION

Modern CAD/CAM systems are best suited to the final tuning and detailing of parts and products and as sources of data for increasingly automated manufacturing processes; they provide little help in the early, conceptual phases of design.

Solid modeling is the best technology for defining mechanical components and products unambiguously if certain theoretical gaps (especially tolerancing) and technological limitations (geometric coverage, speed, complexity limits) can be overcome. Contemporary solid modeling systems provide good support for analytical procedures that can be used to verify final designs and to optimize parametric (nearly final) designs.

However, current systems do not provide much support for the conceptual and preparametric phases of design, which are wholly unautomated at present. Human designers may find a future generation of systems that admit incompletely specified solids, implied solids, and solids defined through constraints to be more friendly, but difficult research problems must be solved before such systems appear.

Automation of the manufacture and assembly of mechanical goods is progressing systematically, with two kinds of modeling playing key roles. Until several years ago, the automation of manufacturing process was concentrated at the effectors (*e.g.*, at the machine tools). The requisite upstream support in the form of manual process planning, machine tool and robot programming, etc., was expensive unless production runs were long. The key to this automation seems to lie in finding effective computational models for processes (machining, forging, dextrous assembly, etc.).

Solid modeling provides unambiguous definitions of what is to be made and also provides directly or through coupled analytical procedures models of the effects of processes on solids.

Lower-level (feature, process) models provide primitives for planning automation that eventually should produce complete sets of plans and programs for making, inspecting, and assembling parts automatically.

Mechanical design automation and, more fundamentally, the understanding of mechanical design in a scientific sense are progressing slowly. Consequently, can be noticed a growing technological inequality, with manufacturing striding ahead of design in terms of both scientific understanding and automation. One of the major gaps in the understanding of design is the lack of means for modeling mechanical function in a manner that links functions to form.

In the view of design and manufacturing, form is central. It defines a part or product as a spatial entity and, when a material specification is added, as a physical entity. Designers, using processes we understand poorly, bring on form from functions. Manufacturing planners, using processes we understand better, but still not well enough, induce fabrication from form. Broadly speaking, the backward mappings from fabrication to form through process simulation and from form to function through analysis are better understood than the forward mappings.

In current industrial practice, form specifications – designs – carry no explicit representations of function and no explicit specifications for manufacturing and assembly. Thus, modern part prints and assembly drawings or their solid modeling equivalents include no descriptions of what parts are supposed to do and how they interact functionally (as opposed to spatially) with other parts. Similarly, there are no form specifications such as mill slot A 10 mm wide or mill slot A of part B to mate with slider C of part D. In current practice, holes, slots, and almost all aspects of form are defined wholly geometrically through toleranced parameters of surface subsets (threads and a few other process-defined features are exceptions).

REFERENCES

[1] Pisarciuc, C. (2004). *New Approaches on Workflow Process Design*, Revista Recent, vol. 5, no. 2 (07.2004), pp. 46–48, ISSN 1582-0246.

[2] Compton, W., D. (1988). *Design and Analysis of Integrated Manufacturing Systems*, National Academy Press, ISBN 0-309-53544-1, Washington, D.C., USA.

[3] Kela, A. (1987). *Automatic Finite Element Mesh Generation And Self-Adaptive Incremental Analysis Through Solid Modeling*, Ph.D dissertation, University of Rochester, USA.

[4] Kela, A., Perucchio, R. L., Voelcker, H. B. (1986). *Toward Automatic Finite Element Analysis*. ASME, Computers in Mechanical Engineering, Vol. 5(1), pp. 57–71.

[5] Buckley, M. J. (1991). *Enabling Technologies for Unified Life-Cycle Engineering of Structural Component*, National Academy Press, ISBN: 0-309-58339-X, Washington, D.C., USA.

Author:

Ph.D. Cristian PISARCIUC, Associate Professor, “Transilvania” University of Braşov, Economic Engineering and Production Systems Department, E-mail: `pisarciuc.c@unitbv.ro`