Virtual Prototypes and Product Models in Mechanical Engineering

Michael J. Pratt
National Institute of Standards and Technology
Manufacturing Systems Integration Division
100 Bureau Drive, Mail Stop 8262
Gaithersburg, MD 20899-8262, U.S.A.
Tel: +1 (301) 975-3951, Fax: +1 (301) 975-4482
E-mail pratt@nist.gov

Abstract

This paper gives an overview of some of the modelling and virtual prototyping techniques used in product realization, with emphasis on the mechanical engineering field. It is pointed out that virtual prototypes, in the commonly accepted sense of computer models permitting realistic graphical simulation, represent only one class amongst the many types of computer models used in design and planning for manufacture. Each such model is usually created for some comparatively narrow purpose, and one of the major problems faced by developers of integrated computer-aided product realization systems concerns the transmutation of one type of model into another. A related problem is that of interpretation by any model of information generated by interrogations of another model. These difficulties are compounded by the increasing presence in such models of semantic information concerning different aspects of the intended functionality or manufacturing requirements of the modelled artefact.

Keywords

Models, virtual prototypes, product realization, mechanical engineering, CAD/CAM.

1 INTRODUCTION

A model is an abstraction or representation of some real thing. It may take many different forms. For example, a mathematical model of the economy of a country may consist of a set of differential equations, while a model of the exterior shape of a new car may be sculpted in clay.

Engineers construct models throughout the product realization process to obtain answers to questions. Sometimes qualitative answers are required; in the car body case the clay model is used to assess the general appearance and attractiveness of the body shape. Other applications, a structural analysis of the car body for example, may require precise numerical results and demand the use of some other type of model.

Given the wide diversity of different types of query arising, for example, in designing and planning for the manufacture of a new airliner, it is inconceivable that any one model can serve for all purposes. Specialized queries demand specialized models; only the real thing – the airliner which has not yet been made – holds the answers to all possible queries.

This paper is concerned with computer models, which reside in a computer and provide support for the mechanical product realization process. In order to set the scene for the discussion of various types of computer models a brief summary will initially be given of the major activities making up that overall process.

2 THE PRODUCT REALIZATION CYCLE

The product realization process can be divided into three stages: design, manufacturing engineering, and production. The output of the design stage is a detailed specification of the product to be manufactured. This becomes the input to the manufacturing engineering stage, whose output gives detailed specifications of the intended manufacturing processes. These in turn are the input to the actual production process.

The three stages are separately described below, although in practice some of their activities may overlap. This is particularly so when modern *concurrent engineering* practices are used, in which case design and manufacturing engineering proceed to some extent in parallel, with frequent exchange of information (Nevins & Whitney 1989).

Much effort is currently being devoted to the use of computers in automating individual product realization activities, and in combining such automated processes into integrated product realization systems. Integration requires the smooth flow of appropriate information between activities, and progress in this area is hampered by the use of different models, each having its own informational requirements, for individual product realization activities. The focus of the present paper is the modeling aspect of automation, the intention being to highlight one of the major problems underlying the achievement of integrated systems for design and manufacture.

2.1 Design

Effective design is crucial to the success of any manufacturing organization, since a major fraction (up to 70%) of the total life cycle cost of a product is committed by decisions made in the early stages of design (Ullman 1992). The objectives of the design process are the attainment of a short development time with high product quality and low production cost. The use of computer models may help significantly in achieving these aims.

The product design function can be broken down into four phases (Pahl & Beitz 1984):

- Product planning
- Functional design
- Configuration design
- Detail design.

The activities actually undertaken in the design process vary considerably according to the nature of the product and the commitment of a company to the use of computer aids. Where families of essentially similar and comparatively simple products are concerned it is sometimes possible to encapsulate the basic design principles used in a few equations or design rules. These may then be used to drive the detail design process in such a way that the designer only has to enter values of a few key dimensions or other parameters to enable the design system to generate a complete specification of the product. The achievement of this situation requires considerable preliminary work in developing new software systems or configuring existing ones for the intended specialized applications.

On the other hand, the design of a more complex product such as a new passenger aircraft can require the individual design *ab initio* of many thousands of completely new components. The design activity can then extend over a period of several years, even with extensive use of computer aids. The overall process involves the extensive use of analysis and simulation in arriving at an optimal design solution meeting all the constraints imposed by conflicting requirements on payload, range, fuel economy, safety, noise generation, price and operating costs.

This very wide spectrum of possible approaches to design implies that any breakdown of the process into component tasks will almost certainly differ from the practice in any particular company. What follows is an 'averaged' breakdown, typical of the practice in companies manufacturing a diverse range of non-modular products.

Product planning: This first phase is essentially clarification of the design task to be addressed. Its initiation may be stimulated by the desire to improve upon an existing product, or by the identification of a new market niche. The latter may be stimulated in turn by new developments in technology. The questions arising at this stage are of a very broad nature – What is the purpose of the new product? What market sector is it aimed at, and what therefore should it cost? What will be the size of its market, and how many should be produced? The output of this phase is a set of constraints on the work of the next phase; in particular, the intended functionality of the product is defined and limits imposed on its development and production costs.

Functional design: This phase is concerned with the achievement of the desired functionality in the new product, subject to the constraints imposed at the product planning stage. There may be several solutions to this problem, possibly making use of different physical principles. An example of a design choice at this level is the decision whether a new aircraft will be powered by jet engines, turboprops, piston engines or some new and exotic form of propulsion. Initially, design choices are made at a high level, but each choice leads to a new set of design problems at a lower level which must be solved in turn. The process is therefore one of successive refinement; at each level, design possibilities are either rejected or followed down to lower levels of problem decomposition. Each new level poses a set of functional problems to which technical solutions must be found by the designers. What results eventually is a set of viable possibilities for achieving the desired functionality whilst satisfying the design constraints. The functional design phase is completed when the possibilities have been evaluated against each other and the one chosen which is optimal from the point of view of estimated cost, estimated performance, or some combination of these and other criteria.

Configuration design: Whereas the previous phase is concerned with a functional decomposition of the intended new product, the configuration phase deals with the mapping of the functional elements of the design onto mechanical systems and subsystems providing the required functionality. This phase therefore covers the specification and layout of assemblies and subassemblies. Once again the process is one of decomposition from higher to lower levels, and some iteration between levels may be necessary to obtain acceptable results. It is appropriate during configuration design to minimize the number of parts in assemblies, and to make preliminary decisions on part materials and manufacturing methods (Boothroyd 1994). As in the previous phase, the result is a multiple set of possibilities from which an optimal choice must be made. At this stage it is possible to make more accurate estimates of cost and performance.

Detail design: In the detail design phase the finally chosen configuration design is fully documented. Detailed drawings or product models are created for all components to be manufactured for the new product, and any standard components to be bought in from outside are specified. Once the detailed part designs are available, it is possible to generate detailed assembly models and to perform various computer-based analyses to determine whether the desired product functionality will be achieved. If not, a design iteration will be necessary.

2.2 Manufacturing engineering

The primary input for this activity is some representation of a product to be manufactured, and the output is a set of instructions for manufacturing it. Certain supporting resources are needed for the automation of manufacturing engineering. One is a database of available manufacturing resources, and another is a set of *process models*, i.e. computer models of the manufacturing processes which may be used in the production process. Most research to date has concentrated on the automatic generation of instructions for the production of machined metal parts (Alting & Zhang 1989, Eversheim & Schneewind 1993). However, there are many production methods other than machining. Some of the most important are stamping and other forming methods for sheet metal parts, die casting and injection moulding. Some attention has been given to process planning for these processes, but the technology is less advanced than for machining.

Another important production process, occurring after the individual parts of a product have been manufactured, is assembly. This activity also requires planning, and the development of automated assembly planning methods is a major current topic of research (Baldwin et al 1991, Sanderson et al 1990).

Various types of product models play an important role in the planning activities mentioned above.

2.3 Production

By the time the production stage is reached the product models have already played their major part. However, they still have some remaining roles, for example as specifications of 'nominal' parts against which measured data from inspection and testing processes can be compared.

3 COMPUTER MODELS USED IN PRODUCT REALIZATION

Traditionally, the output of the design process is a specification of the product to be manufactured in the form of manually generated 3-view drawings together with supporting documentation. The use of CAD systems allows such drawings to be generated by the computer, but other more sophisticated types of geometric product descriptions are now routinely created by such systems, as described later in this section. These are models or representations of the design, whose key purpose is to act as substitutes for the real thing, in particular to provide answers to queries about the real product. Different types of models are generated by various classes of CAD systems, including the 2D drawing, the 3D wireframe model, the solid model and its enhancements containing parametric, constraint and form feature information with their associated engineering semantics.

The complexity of the product realization cycle for mechanical products often makes it appropriate to generate different models of the product, for use in different activities contributing towards the overall process. These models may be crude in the early design stages, but sufficient to provide rough-and-ready answers to the broad questions arising at the time. Clearly the output of detail design should include a fully detailed geometric description of the product; it may also contain a great deal of non-geometric information of various types discussed in the following sections.

3.1 CAD systems and their models

Historically, the first interactive graphical CAD systems were 2D drafting systems. These provided a means for the generation of drawings of the traditional kind, their primary advantage being that this could be done more quickly. The major time-saving resulted from the use of automated techniques for generating drafting symbols and for copying other recurring combinations of geometric elements. Many smaller industrial companies are still using systems of this kind, often running on PCs.

The next major development came in the early 1970s, with the introduction of the 3D wireframe model. This is a representation of the shape of a designed object as a set of edges in three dimensions; its primary significance is that it provides a unified model of the object rather than several partial models, as in the case of the traditional three orthogonal views of the engineering drawing. One immediate advantage of the wireframe representation of an object is that the computer can automatically generate drawings of it from any point of view and in any projection chosen by the user. Wireframe systems have been extensively used by industry for several years, but are now being rapidly superseded by more modern systems.

Most wireframe CAD systems also allow the attachment of surfaces to the edge-based model, and this enables the use of realistic shaded surface renderings. The geometry available generally includes complex doubly curved surfaces such as NURBS (non-uniform rational B-splines), whose use was pioneered in non-graphical systems developed in the 1960s, mainly in the aircraft industry.

The next development was the solid modeler, which brings together the advantages of the wireframe and the surface modelers in an optimal way. Like the enhanced wireframe model, the solid model contains information concerning all the faces of the object, including the surfaces they lie on and the edge curves which bound them. It also stores *topological* information indicating how all these elements are connected together in the model. One significant advance is that most of this information is now generated automatically and verified internally by the system, which can also automatically compute the volume, mass, and moments of inertia of the object. Most major CAD systems now possess a solid modeling capability, though this technology has only recently become widely used in industry.

During the 1970s it was thought that the existence of a complete computer model of the geometry of an object would enable the automation of many activities downstream of design, such as process planning. Unfortunately, during the 1980s this proved not to be so, and further developments in CAD systems have been made and are still being made since that time. There are several different but related thrusts, which are beginning to converge in the CAD systems available today. The aim is to generate not merely a *solid model* (i.e. geometry alone) but a *product model*, containing additional engineering semantics.

Some of the major areas of new development in CAD modeling are briefly summarized below:

Parametric modeling: Here the intention is to allow the design of a product in which certain dimensions are not fixed, but can be varied for purposes of design modification or to generate different members of the same family of products. This capability has existed in a limited form for several years.

Constraint-based modeling: This is related to parametric modelling but is more powerful. It allows the specification of constraints on elements of the design, such as 'these two plane surfaces are parallel', or 'Circle A is concentric with Circle B'. Such constraints are usually driven by the intended functionality of the product, and once defined they are required to hold when any design modifications are made. The provision of this capability is giving rise to many technical problems, but most major CAD systems now offer at least limited 2D constraint modelling.

Feature-based modeling: In the mechanical engineering context a *feature* (or more fully a *form feature*) is a local geometric configuration on the surface of a manufactured part which has some engineering significance. Design features are related to the intended functionality of the product; examples include cooling fins, gear teeth and holes for bearing housings. Other product realization activities may have different feature-based views of the same part. For instance, features for machining processes are simply volumes of material which must be removed, such as holes, pockets or slots. Research has shown that form feature information provides the 'natural' input required for manufacturing engineering applications. It has proved difficult to generate this information automatically from the shape representations used by the purely geometric type of solid modeler mentioned earlier. For this reason, many CAD systems are now providing facilities for 'design-by-features', though few of them currently have any means of automatically generating manufacturing feature models from design feature models.

The most significant aspect of the historical progression of CAD system development is the increasing potential for interpretation of the model by the computer. The manually produced drawing was intended exclusively for human interpretation, whereas the design systems of the future will generate information that will directly drive automated processes downstream of design. In particular, these systems will be capable of creating models that not only provide geometric product descriptions, but also richly augment them with engineering semantics. One current research problem concerns the capture of 'design intent' or 'design rationale', i.e. the retention with the product model of the reasons why particular design decisions are made.

In addition to the essentially geometry-based graphical systems of the kind discussed above, which are what generally come to mind when CAD is mentioned, there is a variety of other types of systems providing additional support for the design process. Some of these are briefly discussed below.

3.2 Modeling for engineering analysis

Analysis and simulation tools provide support for the design process. They aid designers by providing information about functional behavior, cost and other concerns pertinent to the design process. Many computational tools are currently available for structural, thermal and fluid flow analysis and associated simulations. Another widely available form of engineering analysis system provides a means for modeling kinematic assemblies and allowing dynamic simulations of their motion. Such a system often provides an additional capability for vibration analysis of mechanical systems.

Analysis and simulation tools are most frequently used in the detail design phase, after the part is fully described. However, as emphasis shifts toward concurrent engineering (Nevins & Whitney 1989), where decisions must be made earlier in the design cycle, these tools will need to be developed to support the design in its earlier phases as well, for example by providing approximate results on the basis of incomplete design information (Dabke 1994).

One of the most common types of analysis model is the finite element (FE) model, a specialized approximate representation of a part in terms of a mesh of simple geometric elements, used as the basis of structural and other types of analysis (Armstrong 1994). The elements are usually either triangles or quadrilaterals in 2D (e.g. cross-sectional) analysis, and tetrahedra or hexahedra in 3D analysis. In the structural analysis case, loads are specified at the nodes of the mesh (usually at the corners of elements where they connect to each other), and the resulting displacements of the mesh are calculated, again in terms of the nodes. Although FE models appear to be purely geometric in nature, there is also a partial differential equation or variational principle underlying the analysis which makes use of them, and this must also be regarded as an implicit component of any such model.

A major current problem with FE analysis is that, although the process is automatic once the mesh is set up and the loading conditions imposed, a 'good' finite element model cannot in general be created automatically from a detailed geometric product model. There are several reasons why this is difficult, especially in 3D. Some of them are concerned with problems of generating the preferred hexahedral meshes whilst satisfying certain criteria on mesh topology or connectivity. Others are concerned with the avoidance of long, thin element shapes, whose presence leads to inaccurate computed results. The fact is that the setting up of good FE models is an activity generally requiring the knowledge and experience of a highly trained human operative, and it has been found difficult so far to encapsulate the necessary knowledge in a rule-based system. Consequently, the interface between CAD and finite element analysis is at present far from fully automated, and the setting up of analysis models is a lengthy and painstaking task that sometimes creates bottlenecks in the design cycle.

A further aspect of the mesh generation problem is the desirability of idealizing regions of 3D models as thin shells, plates or beams. This allows simplification of the FE model through the use of 2D or 1D elements. The resulting reduction in size of the system of equations to be solved may lead to greatly reduced solution times and possibly also to improvements in accuracy. Advantage can additionally be taken of symmetry of geometry (provided it is associated with corresponding symmetry of loading conditions), since this often permits the results of a full analysis to be inferred from the analysis of only part of the model. This again reduces the size of the computational problem. Full automation of mesh generation therefore requires the automatic identification from a CAD model of symmetries and regions where idealizations can be used. These capabilities currently exist only in certain university research projects (Dabke et al 1994).

Another major problem at present relates to the reverse interface between FE and CAD. The results of FE analysis are in the main human-interpretable, the provision of automatic feedback into the design process being in the very early stages of development. The optimization of designs with respect to functionality and cost is essentially an iterative process, and this paucity of feedback puts the human very firmly in the loop. Optimization can therefore be quite a labor-intensive activity.

This particular type of model has been dealt with at some length because it provides good illustrations of some of the difficulties facing researchers trying to develop integrated product realization systems.

3.3 Virtual prototypes

Virtual or computational prototyping is generally understood to be the construction of computer models of products for the purpose of realistic graphical simulation, often in a 'virtual reality' (VR) environment. This provides the ability to test part behavior in a simulated functional context without the need to manufacture the part first. It is one of many strategies aimed at reducing design cycle time. However, a 'virtual prototype' in this sense is only one amongst many different types of model having value in the design process – the name given to it reflects the fact that this type of model originated in the computer graphics community whilst most of the others discussed above were developed by the engineering community. There is no clear-cut distinction; they are all models, and in the sense that they can be used to provide answers to engineering queries they are all virtual prototypes.

Virtual prototyping also lends itself to realistic process modeling. The availability of a graphical model of a part or product in course of manufacture allows simulation of the effects of manufacturing processes. For example, it is possible to generate animated simulations of material removal during machining processes.

The advantages of using virtual prototypes in an 'immersive design' virtual reality environment are currently being studied by a few large manufacturing companies. Boeing uses it for 'fly-throughs' of complex structures in visual checks for interference of parts, and Caterpillar as an aid for the design of cabs for earth-moving equipment. Other reported users of VR in vehicle design are the Daimler-Benz group and PACCAR.

Such simulations rely on the ability to generate realistic graphical representations at real-time speeds, and to this end the true 3D shape of artefacts is usually approximated for rendering purposes in terms of a large number of planar tiles or facets. Interestingly, this type of model is also routinely generated for quite another purpose – it forms the input to a range of processes variously referred to as *solid free-form fabrication (SFF), layered manufacturing, rapid prototyping* or (more recently) *holoforming*. Stereolithography is an example of such a process, whose intention is the rapid generation, directly from CAD data, of a non-functional physical prototype of a part or assembly. This can be used to judge appearance or to test assemblability of a designed part into an assembly, for example. Many CAD systems generate a faceted representation of a part in an industry standard format known as a .STL file, to provide input to SFF systems. Workers in VR have also found that .STL files provide suitable models for generating animated visualizations.

3.4 Knowledge-based analysis

Knowledge-based systems use expert knowledge bases and inference engines. Their automated use in design requires the provision of interfaces to design systems that convert certain design data to 'facts' comprehensible to the inference engine. The inference engine then uses these facts or assertions in the knowledge base to deduce other facts, a process which may ultimately lead to important deductions about the characteristics, quality, and functionality of the design. In a system of this kind the design model is reduced to a set of assertions in the knowledge base, and depending on the particular application concerned these may be either quantitative or qualitative. The automated use of systems of this kind before the detail design stage is problematical, since design information may still be largely on paper or in the designer's head. However, the importance of advisory design systems is highlighted by the significant advantage to be gained from their use in early design with manual entry of product data (Boothroyd 1994).

A few cases exist where feedback from knowledge-based systems into geometry-based systems occurs automatically, but there is currently no standard allowing the automation of such interfaces in a general way.

3.5 Other examples of non-geometric models

Other kinds of non-geometric models also have a role to play in the product realization process. A model used for estimating production cost, for example, is likely to have the form of an algorithm or set of formulae, taking into account the time needed for manufacturing operations, the operational and depreciation costs of the equipment used, costs related to tool wear and so on.

4 PRODUCT MODELS IN MANUFACTURING ENGINEERING

The type of model required for manufacturing engineering depends upon the nature of the manufacturing process to be employed. There is an immediate difficulty here, in that the process may not be known at the time the product is designed. A subsequent decision on process may necessitate changes to the design to make it more suitable for manufacture by the chosen means. This is just one of many examples of feedback between the various stages of the overall product realization cycle.

For purposes of illustration it will here be assumed that a designed part is to be machined from solid material. Experience has shown that the most suitable type of model for planning this process is one based on form features. For this application the features will be material removal features such as pockets, slots and blind or through holes. The machining strategies available for generating each such feature type are relatively few in number, and they differ primarily in the accuracy and surface finish they are capable of achieving. The choice of strategy for any particular feature may then be made on the basis of the feature type and the required engineering tolerances and surface finish associated with it in the part model. Normally, the cheapest operation meeting the desired criteria will be chosen. If this procedure is repeated for all the machining features exhibited by the part, the resulting set of machining strategies forms the basis of a process plan for its manufacture. They must be sequenced in some logical manner to give the final plan; this requires complex reasoning, but much of the required information is of the same kind as is needed for the earlier stage of the process.

Other manufacturing processes, including assembly, may also be decomposed into feature-based sub-processes, but it is important to realize that different processes will require different feature models of the same part. For example, in machining, the features are all subtractive, but if the part is to be built up by (for example) welding together several originally separate components then the features of the final part are additive. It is possible to arrive at the same final geometry by either method in some cases.

An equally important point is that, if the part is designed in a feature-based design system the designer's feature model will almost certainly not be the most appropriate model for manufacturing planning. The design features are created to provide functionality in the part; they may be either additive or subtractive features, as in the case of a locating pin and the hole into which it fits. However, as we have seen, some manufacturing processes require features which are either all additive or all subtractive. There are also more subtle differences between the feature models appropriate for different applications (Pratt 1991).

A further possibility is that the part is designed in a pure geometry-based system, so that the design model contains no feature information at all. Since a model based on manufacturing features is the prerequisite for the automated generation of a manufacturing plan, the essential problem in both this and the previous case is, how is the manufacturing feature model generated? Some partial answers are provided in Section 5 below.

5 TRANSMUTATION OF MODELS

The creation of feature models for processes downstream of design is one of the major problems impeding the building of integrated product realization systems for industrial use. The automated generation of a manufacturing feature model is discussed in some detail below, since this is currently a major emerging area of research. However, this is just one of many feature model transmutation problems, and some other cases are also given some attention at the end of the section.

5.1 Feature recognition

The initial motivation for working with features came from a growing realization that part models of purely geometric types do not readily provide the kind of information most immediately useful to a process planning system. At one time it was thought that the solid model would be able to do this, but experience proved otherwise. There are two main approaches to solid modeling: a boundary representation (B-rep) system represents a part as a connected collection of faces with specified geometry, while a set-theoretic or constructive solid geometry (CSG) system represents it as a set of points in 3D space, expressed in terms of combinations of simple volumetric primitives such as blocks and cylinders. It was found that B-rep and CSG modelers provided information respectively at too low and too high a level for easy interpretation by a process planning system. The appropriate median level proved to be that of the form feature, expressed as a (usually connected) set of faces in a B-rep model, or as interactions between two or more primitive volumes in a CSG model. It should be mentioned in passing that despite the popularity of the CSG approach some years ago all existing commercial CAD modeling systems are now based primarily on the B-rep methodology.

Much attention has been given to the problem of automatically recognizing form features for manufacturing processes (machining in particular) from a model of a part, usually in the form of a solid model of one of the types discussed above (Shah 1991). In a B-rep context this involves identifying a set of part faces which match some predefined sets of rules characteristic of each recognizable feature type. For example, a rectangular pocket consists of five faces: a rectangular floor, perpendicular to four walls connected at right angles to each other at the corners (and therefore forming two mutually perpendicular parallel pairs). This has proved to be an easy configuration to recognize in isolation, but a much more difficult one where features overlap and their characteristic face patterns are modified as a result. The first commercial generative process planning systems for machined parts based on the automatic recognition of manufacturing features from a solid model are now available. However, they are only successful for a limited part domain, and their capability needs to be extended to cover other types of manufacturing processes.

5.2 Feature model transmutation

Many modern CAD systems allow the designer to design in terms of form features. These systems provide a range of frequently occurring functional features, and also offer the facility for extending this range with user-defined features to meet the specialized requirements of any particular product range. The design process with such a system results in a product model containing design feature information; the problem for process planning is that design features and manufacturing feature are in general not the same. It is only necessary to think of a rib of material created by the designer as a strengthening element. If the rib exists on a machined part then it defines two machining features, one to remove material on either side of it. Whereas feature recognition takes as its input a pure geometric model, the corresponding process when the input is a design feature model is known as *feature model transmutation* (also *feature mapping, feature conversion, feature transformation* – there is no agreement yet on the terminology). Here the problem is to input a design feature model and output the corresponding feature-based model for some other activity such as process planning or inspection.

Although not much has yet been demonstrated in this area (Bronsvoort & Jansen 1994, Falcidieno & Giannini 1990, Shah et al 1994, Wozny et al 1994), feature model transmutation should ultimately prove to be easier than feature recognition, since the input model contains more information. An essential preliminary will be to check each design feature present to see whether it is also a manufacturing feature; if it is, the scale of the remaining problem is reduced. No commercial systems yet provide a capability of this kind. Those having the capacity for automatic feature recognition simply ignore any feature information present in the input model, and use methods based on geometry and topology alone, as described in the previous section.

5.3 Other examples of model transmutation

Other examples have in fact been given earlier in the paper. In all cases quoted, the CAD model has provided the primary or canonical representation, and the other model has been generated from it, generally on the basis of geometric and topological information alone. The generation of an FE model from a CAD model is one example, and in this case human intervention is still generally necessary to achieve the process. The generation of a faceted model for SFF or VR purposes is another example, though it has proved relatively easy to automate this process using an original CAD solid model with exact geometry. Despite this, ‘bad’ faceted representations with missing or unconnected facets are often encountered by organizations using SFF (Barequet & Sharir 1995). Knowledge-based models can sometimes be generated automatically, but other types of non-geometric models generally require human input.

6 FEEDBACK OF INTERROGATION RESULTS BETWEEN MODELS

As stated earlier, models are created for purposes of interrogation. The interrogation results are usually readily interpretable in the context of the model used to obtain them, but for most purposes it would be much more useful to have them interpreted in the context of the original, primary or canonical model, i.e. the CAD model. This was mentioned previously in connection with FE analysis. If this detects an unacceptably high level of stress at a certain node in the FE model, what is the implication on the CAD model? It may be that simply moving that particular node, and some of its neighbours, will lower the stress; the corresponding interpretation in the CAD model might be a thickening of material in a certain region. But in most cases the automatic generation of solutions in the CAD model to problems detected in the FE analysis is far from reality.

Similar problems exist in other cases. VR models, like FE models, are based on rather crude geometric approximations. Thus the accuracy of processes such as collision detection in simulated assembly may not be very high. This makes it desirable to check that a collision detected in the VR environment really exists in the more accurate CAD model environment. However, the links between the elements of the VR model and those of the CAD model are usually non-existent (or at best indirect), which makes automatic feedback of VR results into the CAD environment far from straightforward.

As a final example, a CAD/process planning dialog will be considered. Suppose the CAD model of a part to exist, and suppose also that no decision has yet been made on how it will be manufactured. Possibly there are several alternatives, such as sheet metal stamping, injection molding and die-casting. The original design is probably not ideal for any one of these processes. Ideally, a flexible planning system should be able to evaluate the cost of making the part as designed, using any one of the processes, but also to recommend design changes which will not change the functionality of the part but will make it cheaper to manufacture. In some cases we are currently fairly good at estimating manufacturing costs, but feedback of recommended design changes from the planning environment into the CAD environment is still some way in the future. One of the major barriers appears to be the requirement for the planning system to have some understanding of the design concept of functionality, which does not exist in the current conception of a planning model.

7 CONCLUSIONS

The paper has attempted to make and to illustrate three main points:

1. Multiple different types of product model are generated and used for different purposes in the course of the product realization process. Most of them are generated from a primary CAD model, which usually has a higher level of detail and geometric accuracy than the other types of model, some of which are in any case not geometric in nature.
2. The process of generating the secondary models is in most cases not completely automated, and in many cases is not even well understood. Nevertheless, strenuous efforts are being made to automate the interrogations and processes making use of those secondary models.
3. The information generated by interrogating the secondary models is readily interpretable in the context of those models, but it is often desirable to interpret it in the context of the primary model. We are currently in the very early stages of tackling this problem of information feedback between models.

Taken together, these points lead to an important conclusion regarding the development of integrated product realization systems. Significant advances have been achieved (ISO 1994) in developing standard means for importing, exporting and sharing the data required and generated by individual modules of such a system. However, the problem remains that each module functions in terms of its own internal model. Thus the data exported by one module is often not immediately comprehensible to another, since it is generated in a different context and has different semantics. Full communication between any pair of modules requires not only the *representation* and *transmission* of product data (the problems addressed by current standards), but also its *interpretation* by the receiving module, based on knowledge of both the old and the new context and semantics. The requirement is analogous to that of computer translation between different natural languages such as English and Japanese, a notoriously difficult problem. Much work remains to be done in this area.

The models discussed in the paper may actually be implemented in various ways. At one extreme is the case where all models are completely separate from each other, and communication is through the medium of file transfer or via calls to application program interfaces (APIs). At the other extreme, all the models are in some sense constructed on top of the original CAD model, with built-in associative links between related entities in the various models. The second option appears to make life easier in some ways; for example, it is possible to arrange for a change in one model to lead automatically to consistent changes in all the other models. This is certainly not easy if the first option is adopted. On the other hand, the second option effectively requires the overall system to be integrated through the use of a shared database, with all software modules provided by the same supplier and consequently 'speaking the same language'. This makes it difficult to link other systems which may be needed for specialized applications not supported by that supplier. In practice, most major manufacturing organizations who set out to build integrated systems start with a set of modules performing different functions, chosen for the effectiveness of their performance of those functions, and usually from *different* suppliers. Each module will then generate its own internal models, and the problems described earlier will have to be overcome. There is clearly at present no ideal solution to the integration problem.

As a closing note, the author would like to reiterate the conclusion (generally agreed by the participants of the Providence Workshop) that almost any form of computer model will serve for some purpose as a virtual prototype. The use of this terminology should therefore not be restricted to the domain of virtual reality; the VR community is undeniably doing exciting things, but there are many parallel fields of endeavour in product realization which make use of essentially the same principles; modeling and interrogation are common to all of them.

REFERENCES

Alting L. and Zhang H. (1989) Computer Aided Process Planning: The State-of-the-Art Survey. *Int. J. Production Research* **27**, 4, 553 – 585.

Armstrong C.G. (1994) Modelling Requirements for Finite-element Analysis. *Computer Aided Design* **26**, 7, 573 – 578.

Baldwin D.F., Abell T.E., Lui M.-C., De Fazio T.L. and Whitney D.E. (1991) An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products. *IEEE Trans. Robotics & Automation* **7**, 1, 78 – 94.

Barequet G. and Sharir M. (1995) Filling Gaps in the Boundary of a Polyhedron. *Computer Aided Geometric Design* **12**, 2, 207 – 229.

Boothroyd G. (1994) Product Design for Manufacture and Assembly. *Computer Aided Design* **26**, 7, 505 – 520.

Bronsvoort W. and Jansen F. (1994) Multi-view Feature Modelling for Design and Assembly, in *Advances in Feature Based Manufacturing* (eds. J.J. Shah, M. Mäntylä and D.S. Nau), pp. 315 – 330. Elsevier.

Dabke P. (1994) Developing a Finite Element Analysis Agent. Report, Center for Design Research, Stanford University, Stanford, CA 94305, February 1994.

Dabke P., Prabhakar V. and Sheppard S. (1994) Using Features to Support Finite Element Idealizations. Report, Center for Design Research, Stanford University, Stanford, CA 94305, February 1994.

Eversheim W. and Schneewind J. (1993) Computer-Aided Process Planning – State of the Art and Future Development. *Robotics & Computer-Integrated Manufacturing* **10**, 1&2, 65 – 70.

Falcidieno B. and Giannini F. (1990) A System for Extracting and Representing Feature Information driven by the Application Context. In *Proc. IEEE International Conf. on Robotics and Automation, May 1990, Cincinnati, OH.* IEEE Computer Society Press.

ISO (1994) *Product Data Representation and Exchange*, International Standard ISO 10303. International Organisation for Standardisation.

Nevins J. and Whitney D.E. (1989) *Concurrent Design of Products and Processes.* McGraw-Hill.

Pahl G. and Beitz W. (1984) *Engineering Design.* The Design Council (London, England) and Springer-Verlag.

Pratt M.J. (1991) Aspects of Form Feature Modelling, in *Geometric Modelling: Methods and Applications* (eds. H. Hagen and D. Roller), pp. 227 – 250. Springer-Verlag.

Sanderson A.C., Homem de Mello L. and Zhang H. (1990) Assembly Sequencing Planning. *AI Magazine* **11**, 1, 62 – 81.

Shah J.J. (1991) Assessment of Features Technology. *Computer Aided Design* **23**, 5, 331 – 343.

Shah J.J., Hsiao D. and Leonard J. (1993) A Systematic Approach for Design-Manufacturing Feature Mapping, in *Geometric Modeling for Product Realization* (eds. P.R. Wilson, M.J. Wozny and M.J. Pratt). North-Holland.

Ullman D.G. (1992) *The Mechanical Design Process.* McGraw-Hill.

Wozny M.J., Pratt M.J. and Poli C. (1994) Topics in Feature-based Design and Manufacturing, in *Advances in Feature Based Manufacturing*, (eds. J.J. Shah, M. Mäntylä and D.S. Nau), pp. 481 – 510. Elsevier.

8 BIOGRAPHY

Dr. Mike Pratt is a Senior Research Associate in the Center for Advanced Technology at Rensselaer Polytechnic Institute, Troy, NY, but is currently visiting the National Institute of Standards and Technology, Gaithersburg, MD. Until 1991 he was Professor of Computer Aided Engineering at Cranfield Institute of Technology in England. He holds degrees from Oxford University (physics) and Cranfield (aeronautical engineering, mechanical engineering). He has worked in CAD/CAM for more than 20 years, mainly on applications of geometry in product realization, with particular emphasis on the use of form features in automating the interfaces between its various processes.