HP PE/SolidDesigner: Dynamic Modeling for Three-Dimensional Computer-Aided Design

In most solid modeling CAD systems, knowledge of the history of the design is necessary to avoid unanticipated side-effects when making changes. With dynamic modeling, local geometry and topology changes can be made independently of the model creation at any time, using both direct and dimension-driven methods. The core components enabling dynamic modifications are the tool body and the relation solver.

by Klaus-Peter Fahlbusch and Thomas D. Roser

HP Precision Engineering SolidDesigner (PE/SolidDesigner) is a 3D solid modeling design system based on the ACIS® Kernel (see "About Kernels" on next page). It provides the geometric model needed by design workgroups in product development environments. The system's dynamic modeling technology gives the designer the freedom to incorporate changes at any time and at any stage of product development, without dependence on the history of the product design.

HP PE/SolidDesigner is a member of the HP Precision Engineering Systems (PE/Systems) product family. Today, HP PE/Systems consists of:

- HP PE/SolidDesigner for solid modeling
- HP PE/ME10 for 2D design, drafting, and documentation
- HP PE/ME30 for 3D design
- HP PE/SurfaceStyler, an engineering styling application integrated with HP PE/SolidDesigner
- HP PE/SheetAdvisor, a sheet-metal design-for-manufacturability application
- HP PE/WorkManager for product data and workflow management
- HP PE/DDS-C for electrical system design
- HP PE/Complementary Application Program (CAP), a joint research and development and marketing program that provides HP PE/Systems users with access to more than 200 leading applications from 70 companies.

HP PE/SolidDesigner

HP PE/SolidDesigner makes it easy for designers to move to 3D solid modeling. It supports the coexistence of surface data with solid data and provides the ability to import and modify surface and solid design data from a variety of CAD systems. It also offers new modeling functionality and enhanced ease of use.

Using improved IGES (Initial Graphics Exchange Standard) import capability, both surface and wireframe data can be imported. Surface data and solid data can also be imported and exported using the STEP (Standard for the Exchange of Product Model Data) format. Once imported, this data can coexist with HP PE/SolidDesigner solid data. It can be loaded, saved, positioned, caught to (see footnote on page 15), managed as part and assembly structures, deleted, and used to create solids. Attributes such as color can be modified. If the set of surfaces is closed, HP PE/SolidDesigner will create a solid from those surfaces automatically. Other solid modeling systems, which are history-based, are unable to import data and then modify it as if it had been created within the system itself.

HP PE/SolidDesigner allows solid parts and assemblies to be exported to ACIS-based systems using Version 1.5 of the ACIS SAT file format. This feature provides a direct link to other ACIS-based applications.

With HP PE/SolidDesigner, users can set part and layout accuracy. Because users can model with parts of different accuracy by forcing them to a common accuracy, they can import and work on models from other CAD systems regardless of their accuracy.

Dynamic modeling is the underlying methodology within HP PE/SolidDesigner. This flexible, nonhistory-based, intuitive design technique provides direct interaction with modeling tools and designs, allowing the engineer to focus effectively on the design task.

HP PE/SolidDesigner allows designers to work with user-defined features to capture design intent. Users can explicitly group a variety of 3D elements such as faces and edges of a part. These features then can be viewed, edited, renamed, deleted, or used to drive changes to a design.

HP PE/SolidDesigner has variable radius blending, which allows users to create, modify, and remove variable blends. They can now create constant and variable blends during one session. Another new feature, called shelling, provides a quick way for users to create thin-walled parts from solids, as in injection-molded parts, for example.

Also new in HP PE/SolidDesigner is mass property capability. The following properties can be calculated for parts and assemblies: face area, volume, mass, center of gravity, inertia tensor, and boundary area. Tolerances can be supplied and achieved accuracies are returned. HP PE/SolidDesigner also incorporates interference-checking capabilities, which allow detection of interference, face touching, and noninterference of assemblies and part combinations. The results can be shown as text reports or in graphic format with color coding for easy identification.

About Kernels. A kernel is the heart of a modeling system. Currently, three kernels are used in various CAD systems. These are Romulus from Shape Data, Parasolid, an extension of Romulus, and the ACIS Kernel from Spatial Technology. The ACIS Kernel is rapidly becoming a de facto standard, having been accepted to date by 25 other commercial licensees, 50 academic institutions, and 12 strategic developers. As of July 1995, companies that officially have committed to using ACIS as their underlying technology include MacNeal-Schwendler/Aries, Applicon, Autodesk, Bentley Systems, CADCentre, Hewlett-Packard, Hitachi-Zosen Information Systems, Camax Manufacturing Technologies, Intergraph, and Straessle.

About STEP. The STEP protocol for data exchange is the product of a group of international organizations including PDES/PDES Inc. USA, a joint venture with several member companies, ESPRIT (European Strategic Program for Research and Development in Information Technology), European data exchange technology centers such as CADDETC (CADCAM Data Exchange Technical Centre) and GOSET, and ProSTEP, the German industry project for establishing STEP in the automotive industry.

HP has been active in STEP technology since 1989 through projects such as CADEX (CAD Geometry Exchange), PRODEX (Product Data Exchange), and ProSTEP. HP provides STEP processors with its HP PE/SolidDesigner 3D solid modeling software.

Dynamic Modeling

Currently, the most popular 3D CAD solutions are history-based. When designing with these systems, dimensions and parameters have to be specified at the outset. The model can only be manipulated indirectly by modifying these dimensions and parameters. The initial definitions have a major influence on the ease or difficulty of carrying out subsequent modifications, which can only be reliably implemented if all the previous steps in the design process are known. Laborious manipulation may be necessary to make changes that, intuitively, should be achievable in a single step.

Unless the history of the design is thoroughly understood, any change made to a model may have unanticipated side-effects. Relatively straightforward changes to the model involve many convoluted steps. Future interpretation becomes ever more difficult and the effects of further modifications are unpredictable. Even when a single designer takes a part from start to finish, the designer will usually recreate the model from scratch many times as decisions made earlier make further progress impossible.

Although history-based systems are appropriate for solving family-of-parts problems, and are ideal for companies who simply produce variations on a given design, they are inflexible when used during the conceptualization phase of a project.

Dynamic Modeling

Dynamic modeling has been developed by HP to overcome the many problems designers experience with history-based CAD systems. In particular, it aims to remove any dependencies on history and the need to anticipate future changes.

The concept underlying dynamic modeling is to make optimal use of technologies without constraining the designer's creativity and flexibility. In contrast to history-based systems, dynamic modeling allows direct manipulation of model elements in 3D space. With dynamic modeling, local geometry and topology changes can be made independently of the model creation at any time, using both direct and dimension-driven methods. In the latter case, dimensions can be specified at any stage in the design, not just at the outset.

The core components enabling dynamic modifications are the tool body and the relation solver. To make a model modification a tool body is created and then transformed to the appropriate position. A Boolean operation between the original model and the tool body results in the desired model modification.

HP PE/SolidDesigner is the only currently available CAD solution that uses dynamic modeling. The remainder of this article describes the underlying technology of dynamic modeling and compares it with other methods like parametric model modification techniques.

State of the Art

Currently, solid modelers use two different approaches to create the final geometrical model. CSG (constructive solid geometry) modelers are based on volume set operations with volume primitives such as cubes, cones, or cylinders. This approach is characterized by a Boolean engine, which implements the basic operators unite, subtract, and intersect. The sequence of all the Boolean operations, parameters, and positions of the primitives are kept in the *CSG tree*. Modification of the solid later in the design process can be done by using more primitives or by editing the CSG tree. Local modifications of the model are not possible, since no access to faces or edges is given. This cumbersome way to modify solids requires the user to analyze the design beforehand and dissect it into the necessary primitives and operations. While anticipating design modifications and building designs out of primitives is not typical in the mechanical engineering design process, pure Boolean modelers have proven useful when entering a final design for postprocessing, such as for finite-element analysis (FEM) or NC tool path programming.

B-Rep (boundary representation) modelers represent the solid by concatenating surfaces towards a closed volume. Model creation is similar to CSG modeling, but the user can work locally with surfaces, trim them against each other and "glue" them together. Local geometry modifications are very flexible and represent the way engineers think. For example, "I would like to blend this edge" is a natural way of specifying a model change for a mechanical engineer, while "I have to remove a volume that cuts away all material not needed" is a very unnatural way of specifying the same task during design.

As the development of B-Rep modelers continued, a new class of operations emerged in the early 1980s from the research institutes and appeared in commercial implementations. These operations are called *local operations*, or more commonly, *LOP*s, in contrast to global operations like Boolean set operations. Typical representatives of this kind of modeler are all Romulus-kernel-based systems like HP PE/ME30.

The difference between modifications with Boolean operations and modifications with LOPs lies in the amount of context analysis required. A Boolean operation always works on the complete volume of the operands (global operation). A LOP only analyzes the neighborhood of the operand and is usually not able to perform topological changes. To perform a model modification several constraints have to be met by the model, two of which are illustrated in Figs. 1 and 2.

The example shown in Fig. 1 is a block with edge E1 to be blended (rounded). If the radius chosen for the blend is larger than the distance between the two edges E1 and E2, the topology of the model would need to be changed or the model would be corrupted.

Fig. 2 shows a block with a pocket on its left side. To move or copy the pocket from the left top face to the right one cannot be done using LOPs, because both top faces would change their topology (i.e., add or remove faces or edges). The left top face would lose the inner loop resulting from the pocket while the right top face would add one.

These two restrictions are only examples of the complex set of constraints on the use of LOPs. Removing these restrictions one by one means evaluating more and more scenarios, thus adding to the complexity of the algorithms needed for the operations. A new approach was necessary.

The Tool Body

The limitations illustrated above led to the question, why can't Boolean set operations do the job? Boolean operations would be able to handle all special cases and at the same time would increase the stability of the algorithms. In the late 1980s a lot of research and development was done using this approach. Two directions were taken. The first was to further develop the old-style CSG modeling systems to make them easier to use. The second was to remove the limitations of LOPs in systems like HP PE/ME30 and all other Romulus-kernel-based systems. HP took the latter approach to develop the dynamic modeling capabilities of HP PE/-SolidDesigner.

To enable model modifications with topology changes, Boolean operations were added to the LOP modification capabilities. The system generates a *tool body* and positions it according to the specifications of the modification. A Boolean operation between the original model and the tool body results in the desired model modification.

In this article, the term *basic local operations* (B-LOP) will be used for the normal LOP, which cannot perform topology changes, while the process of using the Boolean operation, if necessary or more appropriate, will be referred to as an *intelligent local operation* (I-LOP). Although the Boolean operation does not need to be done in all cases, the term I-LOP will be used to indicate that there can be a Boolean-based part of the operation.

To use the Boolean set operations for I-LOPs the system needs to create a tool body first. Two major approaches can be distinguished:

- Analysis of the the geometry to be modified and generation of an appropriate topological primitive (i.e., a basic volume element such as a cube, prism, or other) whose faces will be forced (tweaked) to build up the geometry of the tool body.
- Topological and geometrical creation of the tool body in only one step by analyzing the geometrical and topological neighborhood of the face to be moved.

The first approach is easier to implement if a utility function (a set of B-LOPs) is available that performs the task of tweaking a topologically similar object onto the required geometry of the tool body. The tweaking function, however, is tied to the restrictions of this utility function. The second method is more flexible but requires more knowledge about the internal structure of the CAD system's kernel.

We chose the first approach because HP PE/SolidDesigner already provided a working internal utility function for tweaking. The tool body generation for moving and tapering

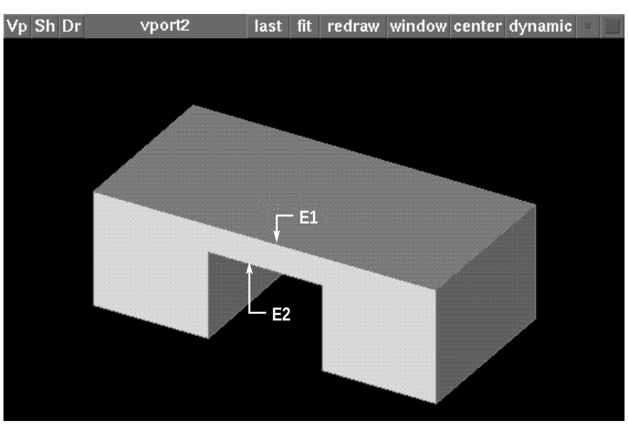

Fig. 1. An example of the limitations of local operations (LOPs). Edge 1 is to be blended (rounded). If the radius chosen is larger than the distance between E1 and E2, the topology of the model must be changed or the model will be corrupted.

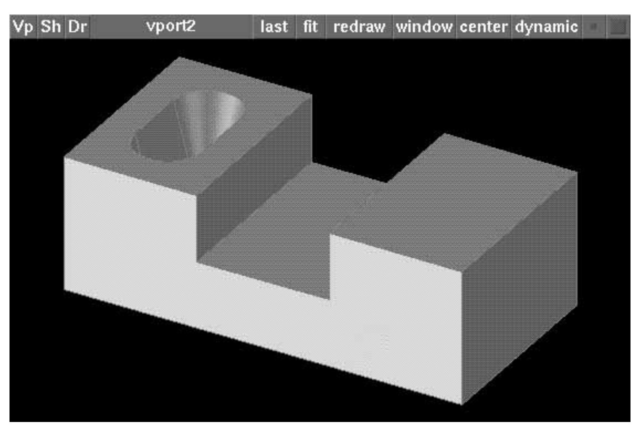

Fig. 2. Another example of the limitations of LOPs. The pocket in the left top face cannot be copied or moved to the right top face using LOPs because both top faces would change topology by adding or removing faces and edges.

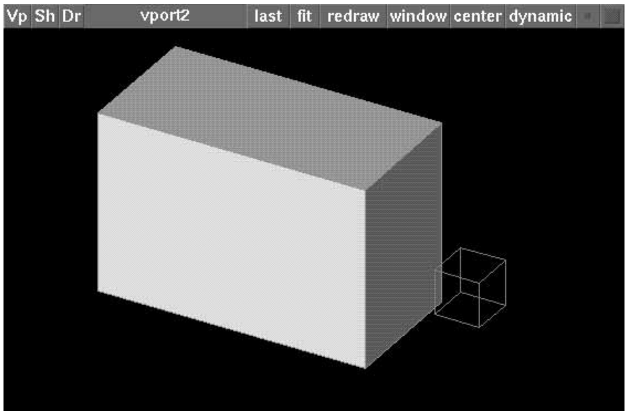

Fig. 3. The first step in the I-LOP (intelligent local operation) approach for a stretch (move face) operation in HP PE/Solid-Designer is the generation of the tool body, a four-sided prism in this case.

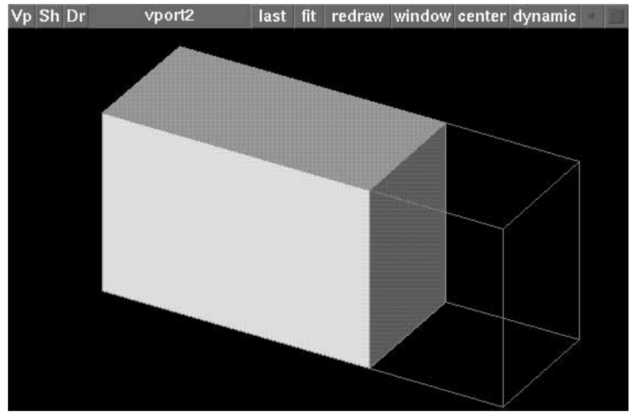

Fig. 4. The second step in stretching the box of Fig. 3 is to force the faces of the tool body to the final geometry, using a B-LOP (basic local operation).

faces (and for bosses and pockets) follows two steps, which are carried out by the system automatically without any user intervention. First, a 3D body is created that has the topology of the final tool body. The part to be modified is analyzed to determine the topology of the 3D body that has to be generated for the requested operation. Depending on the number of edges in the peripheral loops of the face to be modified this body is either a cylinder (one edge), a half cylinder (two edges), or an n-sided prism, where n is the number of peripheral loops. Second, the geometry of this body is modified using basic local modifications. The result is the final tool body to be used for the model modification.

Figs. 3 to 5 illustrate this approach in further detail, showing the I-LOP approach for a stretch (move face) operation in HP PE/SolidDesigner. The user wants to stretch the box in Fig. 3, which means that the right face of the box will be moved to the right. The only and outer loop of the face to be moved contains four edges. Thus, the system creates a four-sided prism in space at an arbitrary position.

As shown in Fig. 4, the system then forces the faces of the prism onto the surfaces underneath the front, top, back, and bottom faces of the box (B-LOP). The left face of the prism will be forced onto the right face of the box and the right face of the prism will be forced into its final position, specified by the user.

The last step, shown in Fig. 5, is the Boolean set operation (in this case a unite) of the tool body with the original 3D part, resulting in the modified 3D part. Although the modification in this example could have been achieved by employing a B-LOP operation, the use of the Boolean set operation will allow topological changes like interference of the stretched 3D part with some other section of the model.

The same approach works for faces with outer loops of n-sided polygons. The curves describing the polygons are not restricted to straight lines. All types of curves bounding the face are valid, as long as the boundary of the face is convex. In cases of convex/concave edges special care has to be taken in tweaking the faces of the prism onto the geometry of the adjacent elements of the original part. An approach similar to the one described applies for tapering faces.

There is a high risk of getting unpredictable results or self-intersecting tool bodies when dealing with several faces that are not related to each other. Although the example in Fig. 6 may look somewhat artificial, it is characteristic of many possible situations. The user wants to move the two vertical faces F1 and F2 farther to the right, and expects a result as represented by the right part in Fig. 6. However, depending on the sequence of selection, two different results can be obtained.

If F2 is selected before F1, the I-LOP performs as expected and the result is as shown at the right in Fig. 6. If F1 is selected first, however, F1 will be moved first. The tool body belonging to F2 will then be subtracted from the body and will interfere with the final position of F1. This leads to the unexpected result shown in the middle of Fig. 6.

The conclusion is that only single faces can be modified and change topology during the modification. For multiple faces the I-LOP is too risky. If multiple faces are to be modified at once, basic local operations (B-LOPs) instead of Boolean operations will be activated. No topology change is allowed, of course. One major exception to this rule is the case of bosses and pockets, which will be discussed later.

Although in most cases the I-LOP approach will be applied, there are situations where self-intersecting tool bodies would be created and therefore the B-LOP approach is preferred

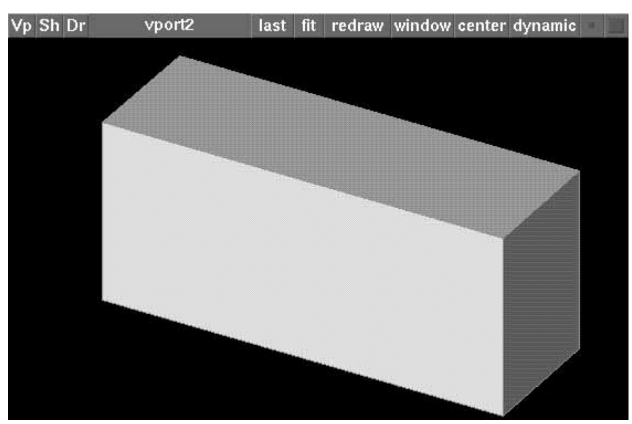

Fig. 5. The final step in stretching the box of Fig. 3 is to unite the tool body and the original part, using a Boolean operation.

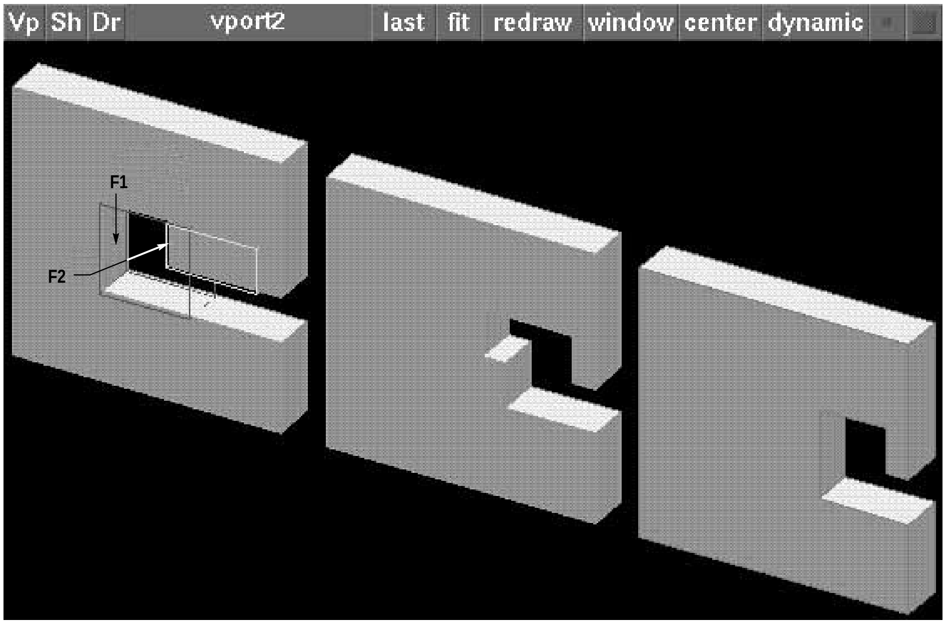

Fig. 6. Modification of several unrelated faces can lead to unanticipated results. Here the user wants to move faces F1 and F2 to change the part at the left into the part at the right. If F2 is selected before F1 the result is as expected, but if F1 is selected first the result is the part in the middle.

even in cases with only one face to be moved. Fig. 7 shows such a situation. The user wants to rotate the right face around an axis lying in the face itself. Another likely situation would be aligning the right face with another face of the model.

Using an I-LOP in the way described above, a self-intersecting tool body would be created without special care to dissect the tool body into two tool bodies, one to add to the part and one to subtract from the part. In Fig. 7, the volume to be added is colored green and the volume to be removed is red. If HP PE/SolidDesigner detects a situation like this, a B-LOP is used for the modification.

Geometry Selection and Automatic Feature Recognition

The next step in terms of increased complexity is the handling of groups of faces, which are known as bosses or pockets by mechanical engineers. These bosses and pockets need to be moved or copied, allowing topology changes. Of course the end user would appreciate it very much if these features could be selected as a unit as opposed to the cumbersome selection of faces sequentially.

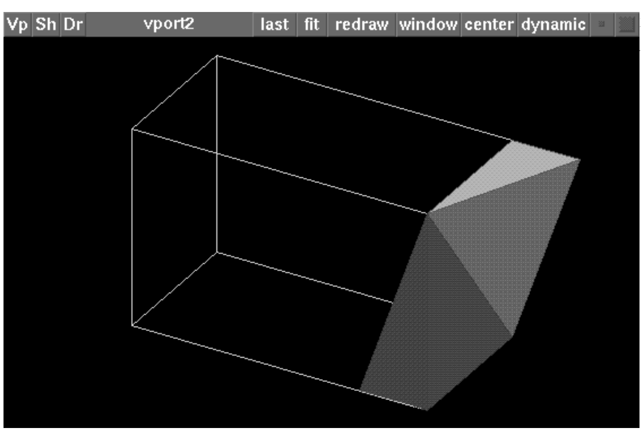

Fig. 7. Here the user wants to rotate the right face around an axis lying in the face itself. This would create a self-intersecting tool body if an I-LOP were used. HP PE/SolidDesigner detects such situations and uses a B-LOP instead.

First, the terms boss and pocket need to be further specified. Bosses and pockets can be defined as a number of connected faces whose exterior boundary loops (the edges describing the intersection of the tool body with the original 3D part) are internal loops of a face. This definition is not easily conceivable and can be replaced by the more understandable, yet not very exact definition, "a number of connected faces contained in one or two nonadjacent others." This is easily conceivable by the end user and fits a lot of cases. Figs. 8 and 9 illustrate the copying of a pocket to which this definition applies.

For moving or copying bosses or pockets the system dissects the part along the edges that connect the boss or pocket with the remaining part. Both the tool body (the former boss or pocket) and the part to be modified now have open volumes (missing faces, or "wounds"), which are "healed" by the algorithm before further processing with the tool body.

Figs. 8 and 9 show only simple pockets. The question remains of how to deal with more complicated situations like

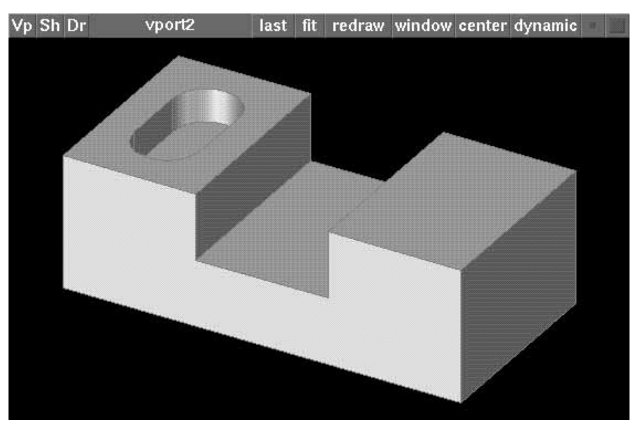

Fig. 8. A part with a pocket to be copied to the right top face.

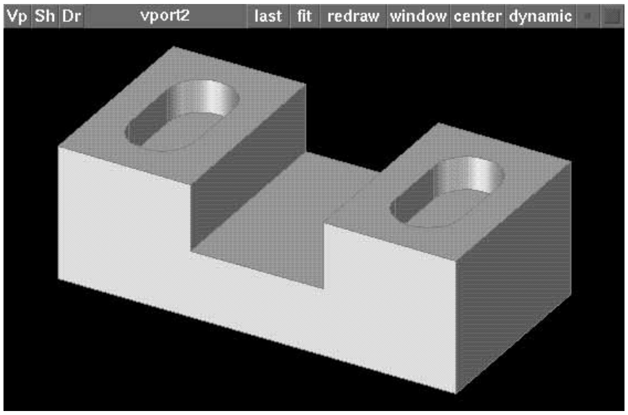

Fig. 9. The part of Fig. 8 with two pockets, one copied. The system recognizes simple and compound bosses and pockets.

countersunk holes or bosses inside pockets. Fig. 10 shows the extension of the simple bosses and pockets. A boss or pocket containing countersunk bosses or pockets will be referred to as a *compound boss or pocket*. Any number of nested bosses or pockets is allowed, as shown in Fig. 10.

Simple and compound bosses and pockets are recognized by the system automatically, depending on the selection of the user. If one face within the boss or pocket is selected, the feature recognition algorithm identifies all other faces belonging to the selected boss or pocket.

Fig. 11 shows a part with a countersunk pocket. If the user selects one of the red faces in Fig. 11, the whole pocket is selected. If the user selects one of the yellow faces a smaller pocket will be recognized.

Feature recognition very much simplifies geometry selection. Instead of many picks to sample the list of faces for a move or copy operation, one single pick is enough. HP PE/Solid-Designer recognizes the list of faces as a boss or pocket and the subsequent modification can include topological changes.

Once the bosses or pockets are selected, various I-LOPs are applied:

- The "wound" in the top face of the part to be modified is healed, resulting in a simple block and a tool body consisting of the two nested pockets (the colored faces).

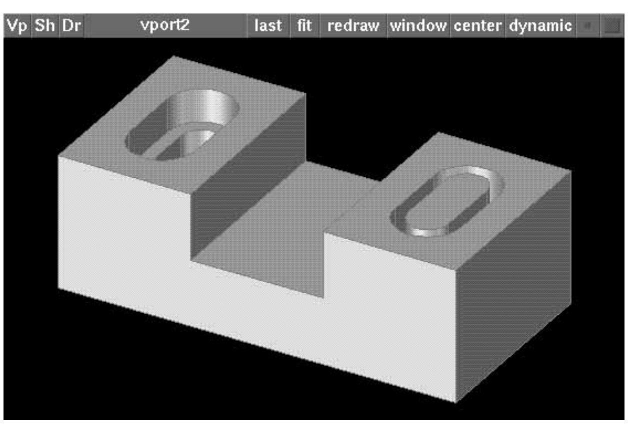

Fig. 10. Part with one compound pocket and a boss inside a pocket.

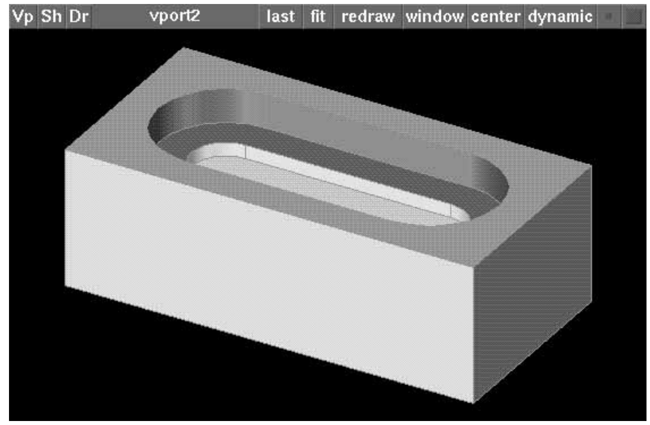

Fig. 11. Part with a countersunk pocket.

- This compound tool body is split into the larger pocket (colored red, nesting level 1) and a smaller pocket (yellow, level 2).
- Both tool bodies are transferred to their final positions.
- The larger tool body is subtracted from the block.
- The smaller tool body is subtracted from the result of the preceding, leading to the desired modification of the part.

The additional complexity of working with compound pockets or bosses is mainly handled by the Boolean engine of HP PE/SolidDesigner. Only a small part—the detection and subdivision of compound bosses or pockets—is needed in the I-LOP code itself.

Fig. 12 shows the result of tapering a compound pocket with HP PE/SolidDesigner. (The front corner of the block has been cut away to show the tapered pocket.) If there were a need to change the topology by this operation, the Boolean operation inside the I-LOP would take care of it.

These features in PE/SolidDesigner don't have anything to do with the generation method of the model, as is the case in history and feature-based modelers. The features are defined temporarily for specific purposes; they are not part of the model. The flexibility of defining features at any stage in the design process is very much appreciated by most mechanical engineers.

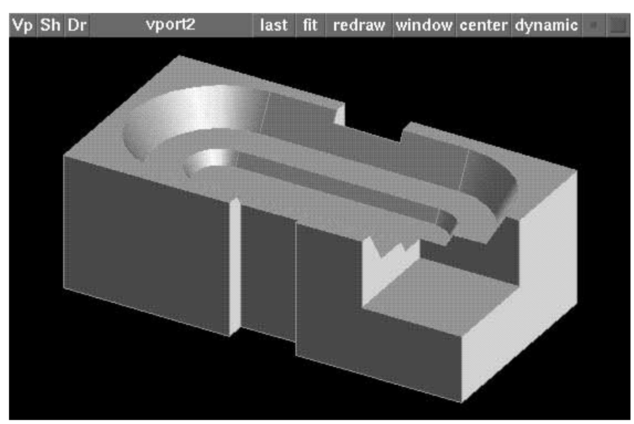

Fig. 12. Part with a tapered, countersunk pocket. The front corner of the block has been cut away to show the tapered pocket.

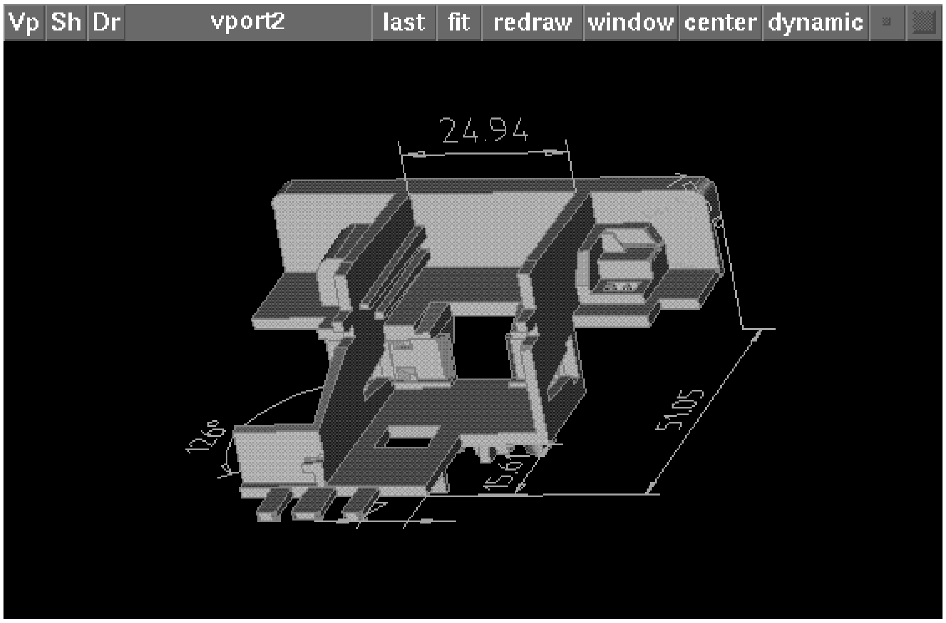

Fig. 13. Part of an HP DeskJet printer printhead.

3D Labels for Dimension-Driven Modifications

In the past, if a mechanical engineer or draftsman had to adapt an existing design to new dimensions, the easiest way was to mark the dimensions as "not true in scale," erase the original value and put in the new value. The rest was left to the people on the shop floor.

This concept of modifying labels was adapted by CAD systems that use variational or parametric approaches in either 2D or 3D. The difference between the parametric and variational approaches is minor in this respect. Both systems require a completely constrained drawing or 3D model which is generated with the help of user constraints and system assumptions. New values of the dimensions cause a recomputation of the whole model. Any dependencies that the user might have specified are maintained even when the model becomes modified later in the design process. The design intent is captured in the model. While this approach is most efficient for family-of-parts designs, it does not support flexible modifications, which are needed in the typical iterative design process.

HP PE/SolidDesigner's dynamic modeling capabilities support the concept of 3D labels that can be attached to the model at any time during the design process and can be used as *driving values*. Tapering of the selected geometry can be driven by angled labels, while the transformation of the selected geometry can be defined by employing distance labels. The user adds one or several 3D labels to the part, selects the geometry to be modified, and specifies new dimension values. Using the new values the system then performs the modification employing B-LOPs or I-LOPs. After the modification all values of the labels are updated to the current values of the geometry.

Fig. 13 shows the HP PE/SolidDesigner model of a part of the printhead of an HP DeskJet printer. Figs. 14 through 18 illustrate the concept of 3D labels.

As indicated in Fig. 14, the first draft of the design contained a 30-degree ramp that was to be used to aid manufacturing. All edges of the area are blended to meet casting requirements. Assume that later in the design process it turned out that the ramp was not needed at all or a different angle was needed. There are several ways to define the transformation in space for the ramp to disappear (e.g., aligning the original ramp face and the adjacent face below the ramp). If the user is trying to define the axis of rotation for the ramp face,

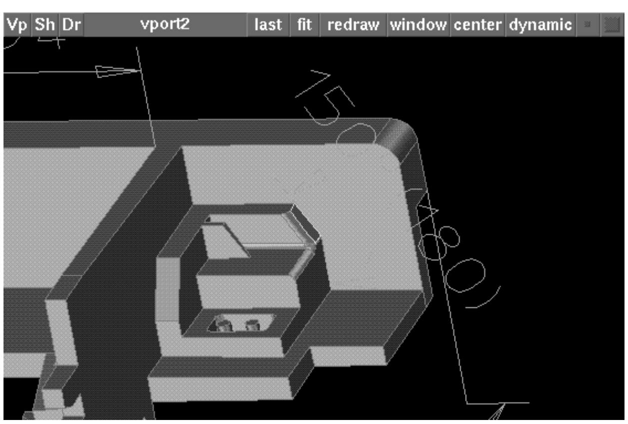

Fig. 14. Changing a dimension (the angle of the ramp) of the part of Fig. 13.

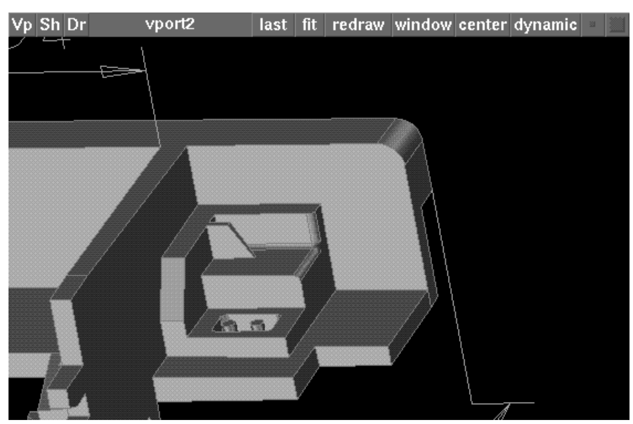

Fig. 15. The part of Fig. 13 with the new ramp angle (the ramp has been removed).

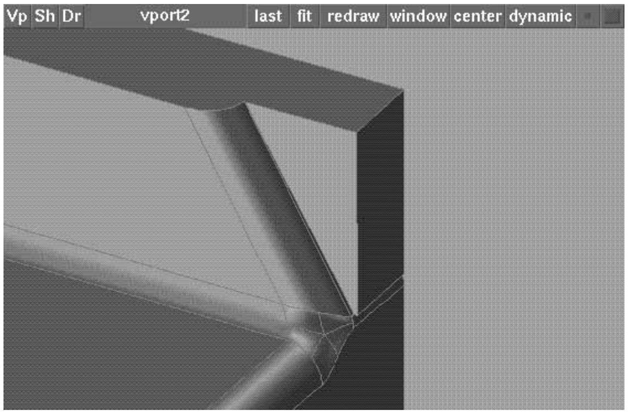

Fig. 16. The part of Fig. 13 changed by I-LOPs without the knowledge that there is a blended edge.

problems arise because the axis is a virtual one and cannot be found in the model. Either a special method for axis definition is needed or the user has to do the calculation by hand.

A third possibility is employing 3D labels. Using the 3D label already defined to show the functional angle enables the system to do all the necessary computation. A new value (in this example 180 degrees) needs to be entered by the user. The system derives the transformation that has to be applied to the ramp face and the model becomes updated. (Fig. 15).

If the label had not been ready for use, it could have been created to drive the modification. The labels are independent from the model creation and can be used temporarily. If the model has been changed, the values of the dimensions update automatically to their new values.

Relation Solver

Once the geometry to be modified is selected and new values of the labels are entered, the system will start with the unspecified transformation and six degrees of freedom (three translational and three rotational). The solver will derive the relations from the labels and reduce the number of degrees of freedom sequentially one after the other until all specified relationships are satisfied or an impossible configuration is encountered.

The system is only designed to solve relationships that can be described by equations solvable by algebraic means. No iterative solution is attempted.

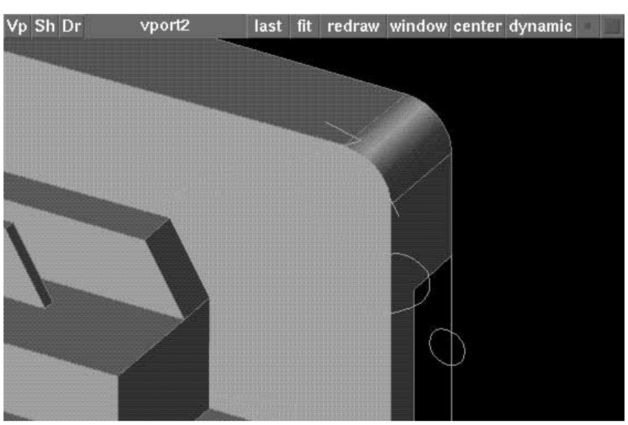

Fig. 17. HP PE/SolidDesigner avoids the behavior of Fig. 16 by first suppressing the blend as shown here.

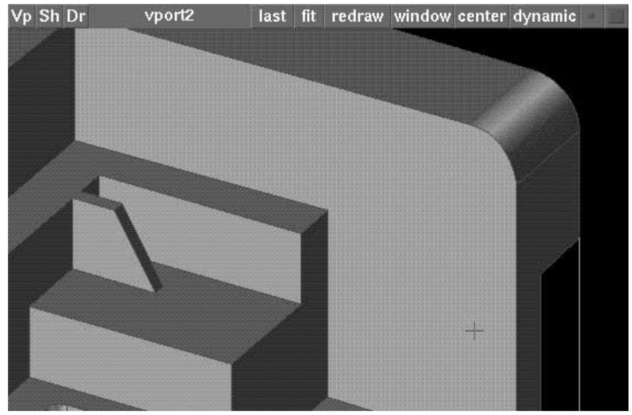

Fig. 18. After suppressing the blend, the system makes the change as shown here. The final step is to readd the blend as shown in Fig. 15.

The resulting transformation is dependent on the order in which the user has selected the modification-driving labels. Thus, the result of the modification is order dependent, especially if rotational and translational transformations are specified for the same modification.

Modifying Blended Faces

In Fig. 14, there are blends adjacent to the face to be moved. If the system didn't know that there were blends in the neighborhood of this face and how to handle them, moving the face might create a strange object like the one shown in Fig. 16.

To avoid this behavior, the system suppresses the blends in a preprocessing step before doing the main operation (rotate the ramp face) and recreates them after performing the main operation in a postprocessing step. Figs. 17 and 18 show the steps used by the system internally.

This concept adds to the flexibility of HP PE/SolidDesigner tremendously, because it overcomes the limitation of the B-LOPs that only modifications can be done that do not involve topological changes.

Summary

This paper shows the strengths of the dynamic modeling techniques. Topology changes are possible in most cases. Model modifications can be defined when they become required within the design process. Design changes do not have to be anticipated when starting the model creation. No constraints within the model exist, and predictable results avoid the trial-and-error approach of parametric and history-based systems. Dynamic modeling's core component besides the relation solver is the tool body, which is defined by the system automatically for the Boolean operation during a model modification. Although some limitations exist, most design changes are possible in one or several steps.

Acknowledgments

We would like to thank all those who helped with the development of dynamic modeling, in particular the Mechanical Design Division HP PE/SolidDesigner team, which was supported by 3D/Eye of Ithaca, New York and D-cubed of Cambridge, UK.