CAD and the Product Master Model

Christoph M. Hoffmann [*]
Department of Computer Sciences
Purdue University

Robert Joan-Arinyo [†]
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

May 13, 1998

Abstract

We develop an architecture for a product master model that federates CAD systems with downstream application processes for different feature views that are part of the design process. The architecture addresses especially the need to make persistent associations of design information with net shape elements. Moreover, the design respects the need of commercial CAD systems (and of downstream applications) to maintain proprietary information that must not be disclosed in the master model.

Two case studies consider the requirements on the master model architecture, for geometric dimensioning and tolerancing, and for manufacturing process planning using NC machining. We discuss how to reconcile the associated feature views and how to update them under net shape redesign. The case studies indicate that many design changes that arise from these downstream views can be formalized by a well-defined problem on dimensional and geometric constraints.

Keywords: CAD, Master Model, Product View, Features, Constraints, Neutral Representations.

1 Introduction

There is a long-standing interest in product data bases that organize discrete manufacturing and the associated data processing. A key problem devising such a data base is to find appropriate mechanisms and information models for recording and interrogating CAD data, and to devise appropriate structures serving different information and process needs in manufacturing and product life cycle maintenance. This difficulty is further compounded by the intrinsic problems associated with accurate conversion of geometric representations, [14], and by the requirement of protecting the proprietariness of certain data and methods of commercial CAD systems and of other software components.

In this paper, we develop a plausible organization for a product master model and address some of the delicate questions of coordinating the CAD system with other software components that take care of downstream information processing. In two case studies, we consider geometric dimensioning and tolerance analysis programs and process planners for numerically controlled machining. These are only some of the many application domains involved in product design, management, and information processing, yet they already span a very broad subject area with a rich literature exploring many details. To focus our investigation, we concentrate on the following question:

> Assuming that the data in the master model originates from different domain-specific programs, how can this information be kept consistent and how is it maintained under design changes?

In our view, the CAD system is one of the clients of the master model, with the primary charge of creating and maintaining the net shape information. Thus, our focus includes the following important special case:

> How can we establish and maintain a persistent association between the geometry data contributed by the CAD system and data originating from other application programs?

To investigate the focus questions, we explore some of the detail information needed by the applications mentioned above. In the case studies, we trace the implications of editing a design with the intent of maximizing the preservation and maintaining consistency of the information in the master model. This exploration is pursued to the extent necessary to make progress on the consistency question. We do not seek to break new ground in feature recognition, process planning, or any other application domain.

We develop our approach based on the following concept: The master model is an object-oriented repository that provides essential mechanisms for maintaining the integrity and consistency of the deposited information structures. It has several clients, one of which is the CAD system by which the net shape has
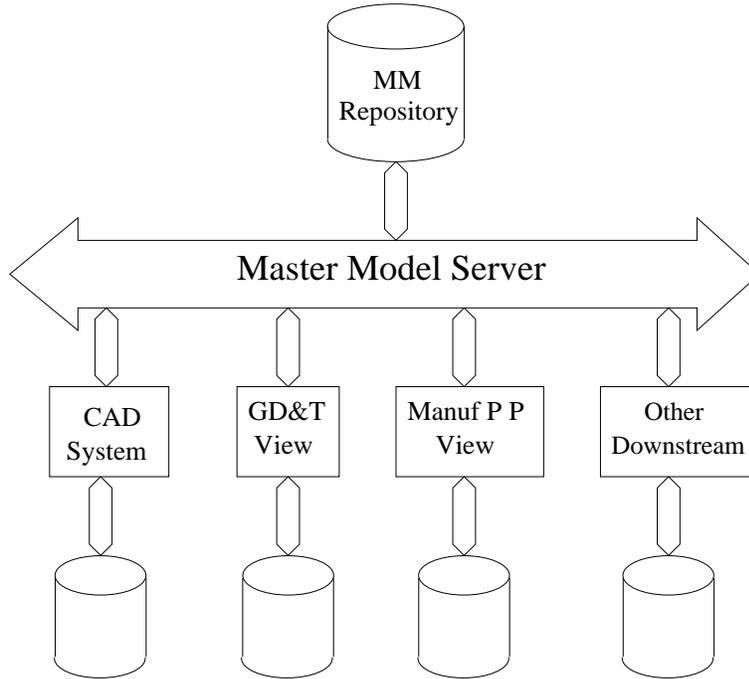
2

Figure 1: Master model architecture with client views

been designed and which must be used to change net shape. Net shape is one of the information structures in the master model.

The clients of the master model, in addition to the CAD system, are domain-specific application subsystems, dealing for instance with manufacturing process planning, geometric dimensioning and tolerancing, cost estimation, performance evaluation, and so on. Each client has a *view* of the product model.

We assume that each client deposits some of the product information it processes, in the master model. In addition, the client subsystem would maintain, at its option, a private data repository for all information that is either to be kept private from the master model, or else is irrelevant to any other application view.

When a change is made, a change protocol is followed to ensure that information structures maintain consistency. Here, we assume that this protocol is supported by the clients. The change protocol is, in effect, a blackboard architecture that posts, in a standard way, which information has changed and how. It is up to the clients to process this change information and to re-associate, adjust, change, etc., all domain-specific information they own. We believe that many of the needed adjustments and changes can be automated, but we allow that in certain circumstances user intervention may be required.

Overall, the architecture we are proposing is shown in Figure 1. The master

model is an object server in charge of coordinating all clients, including the CAD system and the downstream views such as GD&T analysis, manufacturing process planning for machining, casting, forging, and so on. The master model maintains an object repository, and we discuss the protocols and information structures needed to make it work.

Every client may maintain a private repository to do its work. For example, the CAD system would maintain a CAD data base storing shape designs in a proprietary format. The NC view might maintain a private data base to store delta volumes and other intermediate information; etc.

2   Literature Synopsis

The interest in integrated CAD/CAM systems built upon a unique product model dates back at least two decades; see for example the paper by Newell and Evans, [19], or the paper by Semenkov, [22], both published in 1976. More recently, work on this subject explicitly, or even implicitly, has proposed architecture designs for CAx systems. See for example the papers in [20]. We distinguish two major schools of thought.

One school of thought seeks to devise a feature concept that combines as much information as possible pertaining to different application domains and group that information into a compound feature. For example, the IPK Berlin work [16] begins with a CSG notion of form feature and then adds "semantic features" that associate with the form features to provide information and computational services for process planning, performance evaluation, and so on. This approach simplifies the associativity problem because the nongeometric information is clearly mapped to form features where applicable, or else is independent of form features. The main criticism of the approach would be that a primary form-feature view is too limited to address the structuring needs of all downstream domains, and that different domains need to work with different feature views that have form aspects that overlap, but are not necessarily compatible with, the form feature view.

The other school of thought holds that the form feature view is primarily useful for creating the geometry of the product net shape conveniently, but that for some of the manufacturing processes other feature views are needed that work with form aspects which are not clearly related to the form feature view. We illustrate the plausibility of this assumption below in the case study of Section 4.2. Work predicated on this assumption requires the notion of *feature mapping* or *feature conversion*, [24, 23], and magnifies the problems of changing the net shape of a part without losing consistency of the feature information for the other views. See, e.g., [2].

Current approaches handle the consistency and association problems by organizing the systems as a one-way architecture. The features in an applica-

tion view are derived from the features in a privileged view, usually the design view. The designer defines this view and conversion modules generate application dependent feature models. If a modification is required by a downstream application, it must be entered in the privileged view after which new application dependent views can be derived, [4]. It is left open how to respond to design changes, implying that the downstream information must be explicitly re-associated.

It has been argued that to support concurrent engineering, modifications required by an application should be introduced in the view in which the need for them arises, and that all modifications in any view should be propagated automatically to all other views, [2, 6]. Work based on this assumption encounters difficulties explaining precisely the ways in which a feature view other than the design view can change the net shape of the design, and there is a paucity of techniques to formalize such changes. The work by Bronsvoort *et al.* is a notable proposal in that respect, [2, 5, 7, 9]. Briefly, this work models the net shape by a cell complex where the cells are refined such that every feature of an application view is composed of entire cells. That permits to edit shape mechanically from any feature view and achieves consistency across all views. The difficulty of this sound approach is that complex shapes would decompose into very many cells, so that there are doubts that the approach scales in practice. There appears to be some acknowledgement of this problem expressed in [8].

In our view, application domains other than the design view have a need to structure the net shape geometry independently from the way the CAD model does. It is possible that some form features coincide with features of another application domain. For example, a hole in the design view is likely to coincide with a hole feature in the manufacturing view. Moreover, it is helpful to use such form features when creating the geometry, but it is not always possible to do so. Therefore, in our view neither school of thought should take precedence, and we feel that a middle road between them should be taken.

3   Net Shape Element Associativity

We develop a mechanism by which an association can be created and maintained between the elements of a net shape, i.e., the vertices, edges and faces of the boundary representation, and other information. The associated information may be attributes, such as surface finish, or structures that combine a set of shape elements into a feature form that may be of interest to an application domain such as machining. The salient features of the association mechanism are:

1. The client of the master model that made the association does not need to communicate directly with the CAD system and does not need to know which CAD system has been used to create or edit the net shape.

2. The CAD system computes a simple information structure that allows the client to reassociate the information without any knowledge, on part of the CAD system, of the nature of the associated information.

3. The mechanism can be supported by the CAD system using internal methods common to most commercial systems. Furthermore, the mechanism is separate from proprietary internal methods for persistent naming that the CAD system might use.

Thus, the association mechanism supports federating the CAD system with other component systems, through the master model, in a modular architecture. Moreover, the mechanism can also be used internally by CAD systems to support a multiplicity of application views, under control of the CAD system. This would modularize the CAD system architecture and facilitate its extensibility.

3.1 Geometry Certificates

We assume that net shape is represented in a standard way in the repository, as a boundary representation, and that there is free access to all of its elements. Moreover, the geometry representation is with respect to a global coordinate system. Each shape element (vertex, edge, face) has a unique identifier. It is not required that this identifier remains invariant under changes of the net shape. With every shape element, we associate a *characteristic point*. In the case of a vertex, the vertex coordinates define the point. In the case of an edge, any point in the edge interior, sufficiently distant from the incident vertices, will do. In the case of a face, an interior face point is chosen, again suitably distant from bounding edges and vertices. Since net shape is fixed in 3-space, there can be no confusion which shape element the characteristic point belongs to, assuming no errors of evaluating the geometry.

A *geometric certificate* is a triple (I, T, P), where I is the identifier of the shape element in the master model repository, T is the topological type (vertex, edge, or face), and P is the characteristic point. The identifier could be a serial number of the shape element.

Note that the CAD system can find every vertex, edge or face in an evaluated CAD model, from its geometry certificate, by simulating a "pick" operation using the point P. This can be done without knowledge of the net shape element identifier in the master model. Moreover, assume that the CAD client has constructed a net shape and deposited it in the master model, and has stored the CAD model in its private data base in a proprietary format. Then the proprietary CAD model can be re-evaluated at a later date, and the identification based on the geometry certificate will be equally valid in the newly evaluated copy of the CAD model.

The identification process, of a net shape element of the master model in the internal CAD model, based on the geometry certificate, works correctly as long

as the CAD model has not been edited. The topological type is, strictly speaking, redundant in the geometry certificate. Its presence is helpful, however, when the simulated pick delivers ambiguous results due to numerical imprecisions of the evaluation.

We require that the net shape is created by the CAD client, and that the deposition of the net shape in the master model adds to the shape description of each element a characteristic point. Moreover, the CAD system has to provide this net shape representation with a time stamp and an identification of the (private) CAD model stored in the CAD clients private data repository. This information is needed to ascertain that the CAD model is in sync with the net shape in the master model.

3.2   Change Protocol for Shape Edits

After the CAD system has deposited the net shape in the master model, every application domain is permitted to associate information with its shape elements in the master model. For each association, the master model maintains a list that records the geometry certificates of every shape element that has the association, and the name of the client system who made the association.

It is expected that a client informs the master model that an association has been made with a specific shape element. This should be done whether or not the associated information becomes part of the master model, so that the client can be notified of all shape changes that affect it.

When the CAD system edits the net shape, it is provided by the master model with the certificates of all shape elements that have an association. If no information has been associated with a shape element, or if the association has been deposited by the CAD system in the master model and is automatically updated by the CAD system, then the corresponding certificates are not in this *change list*.

The effect of editing net shape, on any shape element, is composed of the following events:

1. The element has moved,
2. the element has been deleted,
3. the element has been joined with another element,
4. the element has been split into several new elements,
5. the element has been enlarged or restricted,
6. the element has been created.

Event (6), the creation of a new element, can occur in combination with other change events. For example, it is possible that an edit joins a face with a newly created one. An example is shown in Figure 2. The box on the left is edited
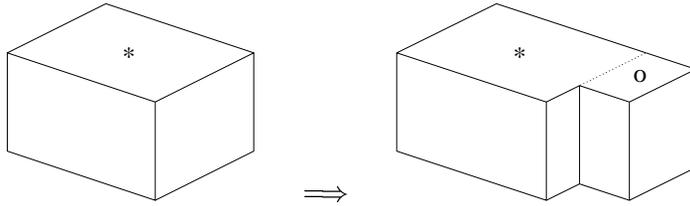
Figure 2: Box edit with event 6 in the change protocol

and a protrusion is added. The top face, marked *, is on the change list at the beginning of the edit. It is merged with a newly created face, marked o.

There may be several events impacting the same shape element. Ordered by dependency, the sum total of the changes becomes a directed acyclic graph in which the nodes are shape elements and the edges are events. The roots of the graph are the shape elements from the change list given to the CAD system by the master model, and the leaves and interior nodes are shape elements in the new net shape. Additional roots of the graph may be old net shape elements not on the change list, involved in a merge, and newly created net shape elements, also involved in the merge.

The graph can be reduced in many cases. For instance, a face may be moved several times during an editing sequence, resulting in a chain of "move" events. Such a chain can (and should) be combined into a single move event. We assume that such reductions are carried out when constructing the graph.

In many cases, the depth of the graph will be one. In particular, if the edit results in no topology change, then the (reduced) graph must have depth 1 and the possible events are restricted to (1) and (5).

3.3   Shape-Dependent Association Updates

After the net shape has been edited in the CAD system, the master model receives the new net shape, its identification in the CAD system's private repository, and the change protocol establishing the connection between the elements of the old and the new net shape. The master model now calls on each application client and requests an update of the information that was associated with the old net shape. How the reassociation should be done depends very much on the application domain; see also Sections 4.1, 4.2, and 6.

We consider as a simple example an association of textual notes with some of the net shape elements. Assume that there is an annotation program that is a client of the master model. The annotator can read the net shape from the master model and display it visually. It also has an interface that permits selecting net shape elements and associating new text with that element, or editing such text. Each annotation can be put into the master model and is associated with the respective net shape element by referencing the identifier of

8

the element. When the net shape has been edited by the CAD system, then the old net shape becomes invalid and the new net shape has to inherit the associations made by the annotator. That is, the net shape identifiers stored with the annotations have to be updated. The annotation program has to process the change protocol to update and re-establish the associations.

Re-association is governed by domain-specific rules. The rules must specify which action to take under each of the six events explained before. For example, in the case of textual annotations, the rules could be as follows:

(1) *move*   re-attach note to the moved element
(2) *delete*   notify user
(3) *join*   concatenate notes, attach to new element
(4) *split*   attach note to each descendant
(5) *change*   re-attach note to the changed element
(6) *new*   do nothing

More complicated rules might be appropriate, depending on the subject matter of the notes. They are under the control of the annotation program.

Considering a more complicated scenario, assume that it is possible for the user to supply some change action with each note. For instance, the user might attach the note "do not delete" to a face and stipulate the action *notify user* with event (2). This simple mechanism would alert designers to a net shape change that might violate some design assumptions.

3.4   Discussion

Traditionally, once a CAD system emits a standard Brep of a design, for instance in IGES or in STEP, the connection of the shape elements with the CAD model is lost. The CAD model, stored in proprietary format, can be edited conveniently, the standard Brep model cannot. Therefore, any associations that are made by downstream processes with elements of the CAD model are lost as soon as the CAD model is edited and a new net shape is emitted in standard Brep. Complicated geometric matching algorithms would be needed to achieve a measure of "external" associativity between downstream information and net shape, and such algorithms would fail when the topology of the net shape has changed appreciably.

The change protocol establishes a mechanism for preserving downstream associations. By factoring the work of re-association between the CAD system and the downstream programs, the CAD system is not burdened with having to understand the nature of the associated information and how it should be treated under edits.

Each downstream program, now in control of the re-association, can implement its own re-association semantics. This is appropriate because the downstream program has intimate knowledge of the application domain. On part of

9

the downstream program, re-association requires understanding the individual events comprising the change protocol graph. This is made easy by breaking the effect of shape change into a fixed repertoire of elementary events.

Re-association also requires understanding how the associated information interacts with other application domains. Therefore, it is easiest in those situations where the associated information is not interrelated with other associated information. In such situations, simple rules suffice to successfully process graphs of arbitrary complexity, as demonstrated by the example of textual annotations.

If the information is interrelated, or if re-association cannot be carried out by a sequence of atomic actions at each node, then the change protocol graph has to be understood in a deeper way, and/or its complexity has to be restricted. For example, we might have to require that any path in the graph have length 1, that is, each shape element undergoes a single event. For edits that do not change the topology, this property can be guaranteed. Conversely, if the graph has depth 1, then the possible topology changes are especially easy to understand. This argues that if major design revisions would be carried out by a sequence of smaller shape edits, then the downstream re-association work is made easier, albeit carried out in several phases.

Many CAD systems offer mechanisms for association through programming interfaces that allow users to attach information with internal structures. Moreover, geometry kernels such as ACIS provide a call-back mechanism by which a user-provided routine can be called as the CAD system edit proceeds.[1] Since these mechanisms are specific to the CAD system, the various downstream programs have to be customized to conform to the particular conventions of each CAD system. This means that there would have to be a different customization for different CAD systems, a burden that is eliminated if our change protocol is implemented and the standardized representation of the net shape is CAD system neutral.

4   Case Studies

We consider the information requirements and changes of two application domains, geometric dimensioning and tolerancing, and manufacturing process planning for NC machining.

4.1   Geometric Dimensioning and Tolerancing

Geometric Dimensioning and Tolerancing, or *GD&T*, has developed a set of symbols to define part shapes, feature form, orientation, runout, profile and

---

[1]This call-back can be considered an automated strategy of breaking up the process of large edits into small steps that are interleaved with re-association computations.

location. Once tolerances are assigned, no doubt should be left as to what is desirable and acceptable, both regards function and cost, so as to obtain satisfactory parts and assemblies. Geometric dimensioning and tolerancing is a system based on function and interrelationship of mating features while keeping in mind manufacturing processes and inspection requirements; [13, 18].

The general geometric dimensioning and tolerancing problem has two parts: analysis and synthesis. *Tolerance analysis* attempts to predict the effect that the tolerances have on one or more design functions. *Tolerance synthesis* attempts to compute a set of tolerance values which satisfy constraints on the function of the design.

The first case study in this section traces the implications of implementing GD&T by a separate application program that communicates only with the master model. In particular, the GD&T application does not directly interact with the CAD system. All coordination between the GD&T model and the CAD model is the responsibility of the master model. We assume that some, but not all GD&T specifications to the product model may originate with the CAD system, for instance some parametric tolerances. If all GD&T information were associated by the design view, there would be no need to consider master model consistency between the design view and GD&T view.

4.1.1 Parametric and Geometric Tolerances

There are two primary types of mechanical tolerances: parametric and geometric. A *parametric tolerance* defines a limit or a pair of limits applied to some scalar parameter of a model. It is represented by the nominal value plus an upper and lower limit, expressed in a standard formalism.

A *geometric tolerance* defines constraints on the geometry of a feature, that is, the tolerance defines a zone in which the feature shape must lie. The features so toleranced include planes, cylinders, cones, spheres and tori. There are several kinds of geometric tolerances, controlling various aspects of form and placement deviations: *form tolerance* defines the permitted deviation from an ideal, nominal form, *orientation tolerance* defines the permitted deviation from the ideal, nominal orientation, *location tolerance* defines the permitted deviation from the nominal position, and *runout tolerance* defines the permitted deviation of the feature element from a nominal radius when revolving the part.

A number of researchers have addressed the difficult issues of giving a precise, mathematical foundation for tolerance specification and analysis; e.g., [3, 11, 27].

4.1.2 References to Net Shape Elements

Three types of entities are involved in a GD&T system: Geometric elements of the net shape, datums, and dimensions and tolerances.

11

The geometric elements to be dimensioned and toleranced are geometric features such as planes, cylinders, cones, spheres and tori.

Datums define the orientation and location tolerance zones. Datums are exact geometric entities, such as axes, planes or straight lines, to which tolerances are related. A *datum feature* is a geometric element of the net shape of a part, such as an edge, a planar face or a hole, that is used to establish the location of a datum. A *datum target* is a restricted area, a line segment, or a point, of specified location on the part geometry. Datum targets are used when the datum features on the physical part may have relatively large form deviations from the tolerance orientation or location. For instance, a nominally planar face of the CAD model may not be sufficiently planar on the manufactured part. So, we may define a target datum plane by locating three target datum points on the face that in turn define the datum plane. Rules are established how to measure and locate the target datums on the physical part when inspecting it.

Datums may be established in several different ways: by a single datum feature, by two or more datum features with some specified priority, or by datum targets having given priority. Priority defines a precedence order when fulfilling tolerance requirements. Three levels of precedence are usually considered; primary, secondary and tertiary.

Dimensions and tolerances would be built from a set of symbols according to rules set forth in standards such as [30]. Parametric tolerances can be represented as follows. The most widely used representation specifies the nominal value plus two signed maximum deviations. If the tolerance is symmetric, that is, the deviations have the same absolute value, only one value is given preceded by ±. Another representation is by two extreme values. A third by specifying the number of significant decimals.

Geometric tolerances have, in general, two components: a parametric tolerance and a *feature control frame* which defines the allowed deviation from the ideal size. The geometric tolerance applies to the feature being controlled. Feature control frames are composed of several fields: A geometric characteristic symbol, a tolerance zone descriptor, a tolerance of location, a material condition symbol, and primary, secondary and tertiary datums.

Figure 3 shows a dimensioning and geometrical tolerancing definition example. The annotation on top gives the parametric dimension; there are two holes with a diameter ranging from 0.250 to 0.260. The hole feature control frame tolerance is enclosed in the box and is interpreted left to right. The symbol in the first field indicates that the control frame is a location tolerance defining a true position. The second field prescribes that the actual hole axis should be contained within a cylinder of diameter 0.030 whose axis coincides with the theoretically exact position at maximum material condition. The third field describes that the theoretically exact position is defined by the theoretically exact dimensions with respect to datums A, B and C, where datum B is considered

12

2x   ⌀ 0.250- 0.260

⊕	⌀ .030 Ⓜ	A	B Ⓜ	C

Figure 3: A geometric dimensioning and tolerancing definition

at maximum material condition.

4.1.3   GD&T Features in the Master Model

Geometric dimensioning and tolerancing information units may refer to geometric elements belonging to different manufacturing features. Therefore, it is appropriate to define a new feature type to represent them. The feature will include:

1. Dimension
   (a) number of instances
   (b) type of dimension: diameter, radius, ...
   (c) nominal dimension
   (d) upper and lower limit deviations for the nominal dimension
   (e) references to geometric elements supporting the dimension

2. Feature control frame
   (a) geometric characteristic: a symbol
   (b) descriptor: symbol
   (c) location tolerance: value
   (d) material condition: symbol
   (e) primary datum: datum, symbol
   (f) secondary datum: datum, symbol
   (g) tertiary datum: datum, symbol

Instead of items (1c) and (1d), the limits for the dimension value may be given. A geometric dimensioning and tolerancing entity may have more than one feature control frame.

Since nominal dimension values are also used for the constraint schema of the part design, and may be toleranced in some design systems, adding GD&T features may lead to inconsistencies when editing the design. Therefore, we have to address how to maintain associativity and consistency between GD&T features defined by the CAD system and those defined by the GD&T view.

A datum in geometric dimensioning and tolerancing is always either a geometric element of the part's boundary or is derived from geometric elements of the part's boundary. In the first case, the geometric element referred to is part of the net shape description. In the second case, a new geometry must be

created, but it is functionally dependent on target datums that refer in turn to boundary elements of the net shape.

4.1.4   GD&T and Master Model Associativity

A downstream GD&T application creates three types of information that should be noted by the CAD system when editing:

1. Add a new dimension or tolerance.
2. Delete an existing dimension or tolerance.
3. Modify a dimension or tolerance.

Some dimensions and tolerances may originate from the design view. In that case, the dimensions are used for the constraint schema defining the variational CAD model. Deleting or modifying those dimensions impacts the net shape.

When new dimensions and tolerances are created by the GD&T application, the corresponding features and datums are generated in the master model. Datums that are geometric elements of the net shape are identified in the master model. Target datums are created as new geometric entities and reference the appropriate net shape elements and other datums. If some dimensions and tolerances have been deleted, they are eliminated from the representation in the master model. If some dimensions and tolerances have been modified, they are updated in the master model representation.

Figure 4 illustrates a modification process taken from Henzold, [13]. In the figure shown on the left, the dimensions and tolerances are as defined by the designer. The part has a hole with a nominal diameter of 30 and a maximum upper deviation of +0.1. The hole surface is labeled as datum B. Symmetrically placed on the surface of the hole three slots have to be machined. Each slot has a nominal width ranging from 5 to 5.25. The slots feature control frame defines a true position for the middle plain of each slot at maximum material condition. Finally, the theoretically exact position of each slot is defined by theoretically exact dimensions with respect to datum B at maximum material condition. Note that here the total tolerances are indicated at nominal size, probably because, for the designer, this representation of the design is the closest to his view.

Since these conditions are difficult to attain, manufacturing recommends to change the dimensions and tolerances to distribute the tolerances on both size and geometry. The new values are shown on the right side of Figure 4. The nominal diameter of the hole is still 30 but now it can range from 30.1 to 30.03. The actual hole axis should be contained in a cylinder of diameter 0.03 whose axis coincides with the theoretically exact postion at maximum material condition. The slots width range from 5.12 to 5.25, a range narrower than the previous one. This narrower range is compensated by the fact that now the true position of the middle plain may depart ± 0.12 from the theoretically exact position at maximum material condition.

3x 5 + 0.25

⊕ | 0 Ⓜ | B Ⓜ

3x 5.12 + 0.13

⊕ | 0.12 Ⓜ Ⓡ | B Ⓜ

3x ⌐120°

3x ⌐120°

B | 30+0.1

Ø 30 +0.1 / +0.03
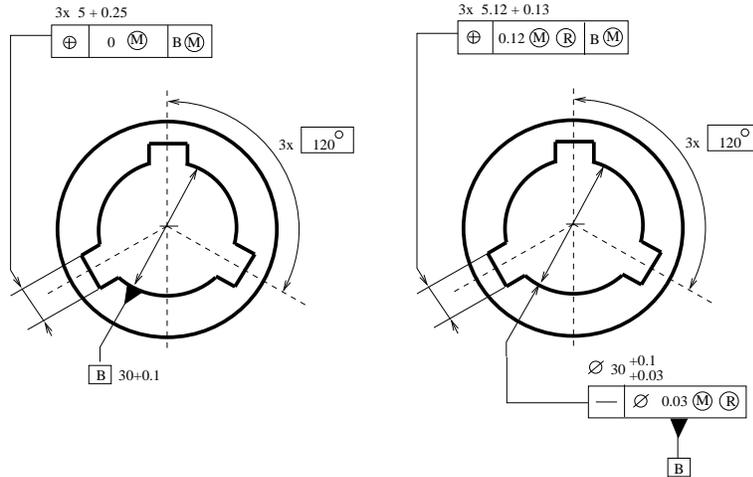
— | Ø 0.03 Ⓜ Ⓡ

B

Figure 4: Modifications in a dimension and tolerance specification.

As a consequence of the interaction between the GD&T application and the master model, the master model may have changed. The changes are communicated by the master model to the CAD system in a suitable format, and the geometry certificates are used to assure that the shape element references in the master model can be correctly collated with internal CAD structures needed to edit the CAD model.

The CAD model will have defining and driven dimensions. Defining dimensions are valuated explicitly by the designer and define the instantiated net shape that was communicated to the master model by the design view. Driven dimensions are valuated implicitly from the instance. Roughly speaking, driven dimensions are measured from the constructed net shape.

If the CAD model is not fully constrained, then the addition of new dimensions will impose new constraints that likely affect the net shape. The constraint solver of the CAD system typically infers appropriate additional dimension values in the course of solving the constraint schema, using a private, internal strategy. Those assumed dimensions may now become obsolete because of the added GD&T dimensions. This requires resynchronizing the master model with the design view.

Some GD&T dimensions may overconstrain the CAD model requiring reconciliation of the conflicting dimensions (see Section 5). The GD&T application can be expected to be capable of detecting conflicting dimensions, thus offering an opportunity to override design view dimensions and altering the constraint schema used to define the net shape. Two scenarios can be implemented:

1. The design dimensions are considered immutable. In this case, the GD&T application would not be allowed to alter or delete design dimensions,

and reconciliation would have to be done by appropriately changing the dimensions and tolerances introduced by the GD&T application. This can be done by the GD&T view alone if the CAD system has fully constrained the model.

2. The design view dimensions and tolerances can be changed by the GD&T application. Here, reconciliation must be done by the design view, altering or deleting any dimension or tolerance until consistency is achieved, giving due priority to the GD&T view dimensions.

In the first scenario, no change of the net shape can ensue because all design dimensions are preserved, provided the part is fully constrained. In that case, no further action is required by the master model. If the part is only partially constrained in the design view, then net shape changes are likely and must be computed as described next.

In the second scenario, we allow the change of design view dimensions and tolerances. The complication here is that the change of a defining design dimensions implies a change of driven design dimensions. The implied changes cannot be evaluated by the GD&T view without substantial duplication of CAD system functionality, and are therefore best done by the design view. Moreover, the inferred dimensions used by the CAD system to handle underconstrained specifications are not normally disclosed. We can determine whether such an inferred dimension or geometric constraint has been changed when the imposed GD&T dimension is at variance with the net shape in the master model.

In either scenario, the interaction between the design view and the GD&T view concerns the constraint schema of the design view and the implications on net shape. Therefore, the second scenario requires a two-step update to synchronize the master model: First, update the design view (from the master model). This update may involve net shape changes according to the deletions or modifications to design view dimensions, and accounting for the newly added GD&T dimensions. Second, post the net shape changes to the master model which in turn calls on the GD&T application so that the application can examine the change protocol and update its internal data structures and recompute datums that are functionally dependent on the net shape.

4.2 Manufacturing Process Planning

Given a product to be manufactured, manufacturing process planning maps the design of the product onto the methods used to create it. Process planning involves [31]:

- Recognizing the constituent elements of the product;
- Defining work tasks for the construction or manufacture of each element;

- Choosing manufacturing or construction technologies and resources used in these tasks;
- Estimating duration and cost of individual tasks;
- Preparing project schedules.

The resulting plan consists of the selected resources and technologies associated with the tasks, and the assignment of the tasks to time slots in the production schedule.

In the following, we will focus on mechanical manufacturing processes employing numerically controlled machining (NC). In this field, the conceptual elements of the product are *machining features* and the purpose of the planning task is to detail step-by-step the machining processes required to realize the final manufactured part. Typically these tasks are concerned with high-level processing such as defining:

- How to prepare the stock,
- the number of setups of operations that are required,
- the machining tools to be used for the setups and operations,
- stock fixturing and mounting details, and
- the areas and shapes to be machined; i.e., the machining features.

As part of the plan, a manufacturing method is assigned to each machining feature. These methods, in turn, refer to the various manufacturing steps or tool paths required to machine the shape by NC-driven manufacturing machinery. Note that several features might be manufactured in the same step, for instance when gang drilling, or in the same setup.

A part can have one or more manufacturing plans, each prescribing a different way to machine the part. This offers some degrees of freedom that allow developing different strategies to machine the part using machines of different types, studying alternative machining strategies, and assessing the effects of a design change.

Restricting in the following to machining features, we curtail the diversity of the feature set under consideration and restrict the manufacturing processes. Note, however, that the question of how to organize the information so as to maintain a valid master model, is paradigmatic of the larger manufacturing context.

4.2.1 Design Features and Machining Features

By design features we mean those features that are used by the designer when creating the solid model from which to derive the net shape. The repertoire and semantics of available design features as well as the way in which they are used, depend to a large extent on the considered CAD system. Common design features available in most of the systems include cuts and protrusions. The

meaning associated with a cut is material removal while a protrusion means material addition. Typically, design features that are cuts include holes, slots, steps and pockets. Bosses, pads, and ribs are protrusion features. User-defined features can add to this repertoire and allow the introduction of verifiably valid design feature use; [15].

A manufacturing feature is commonly defined as a collection of related geometric elements that correspond to a particular manufacturing method or process, or which can be used to reason about a suitable manufacturing method or process for creating that geometry; [25].

Each machining feature has a boundary and a set of characteristic parameters that completely and unambiguously define it. The elements in the boundary definition can be tagged with specific names. For example, a pocket would have floor and side faces. Moreover, the shape elements adjacent to the feature boundary can also be tagged. For example, an open slot would have an entry and an exit face. The parameters describe the feature's size, location and orientation. Examples include axis, diameter, depth, width, etc.

The repertoire of machining features is limited by the type of machining tools and processes available to create them, and their semantics is determined by the fact that NC machining is always a subtractive process. For example, the ISO standard [1] classifies machining features as *machining*, *transition*, and *replicated* features. Machining features include slots, holes, pockets, steps, etc. Replicate features are patterns built by replicating a given machining feature. Transition features are chamfers, edge rounds and fillets.

4.2.2   Generating Machining Features

When a part requiring NC machining has been designed, it can be manufactured only after the machining features have been identified on it. The derivation of machining features from the available design data is therefore an important task. In general, design features differ from machining features in type, semantics and way of use. As a consequence, deriving manufacturing data from design data, in the most general cases, turns out to be a difficult problem.

To illustrate the difference between the design and machining features, consider the pocket of Figure 5. The pocket contains a bridge across the bottom. It is plausible that the designer creates this shape by first making a pocket of maximum depth, and then adds the rib as a protrusion with fillets at the edges. So, the design feature view sees a three-step progression, namely block $\rightarrow$ cut $\rightarrow$ protrusion. See also Figure 6.

Since machining is a purely subtractive process, the feature structure for machining purposes might be as illustrated in Figure 7, namely block $\rightarrow$ shallow pocket $\rightarrow$ deeper pocket to the left of the bridge $\rightarrow$ deeper pocket to the right of the bridge.
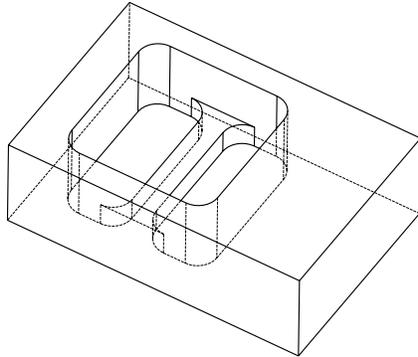
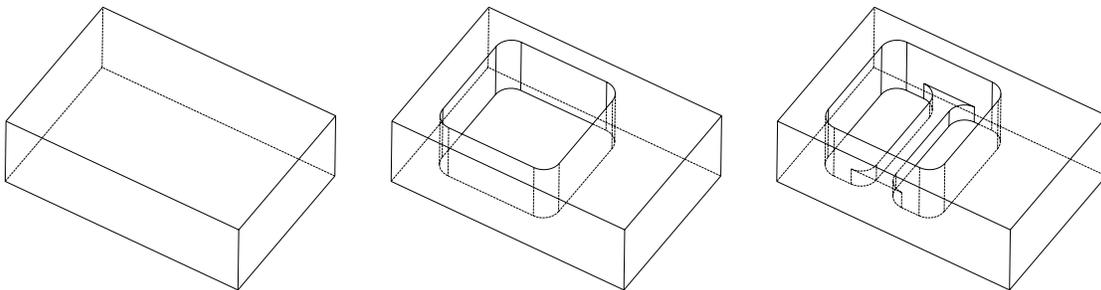Figure 5: Block with a pocket containing a hole through a boss.

Figure 6: Design view sequence: make a cut, then add a protrusion.
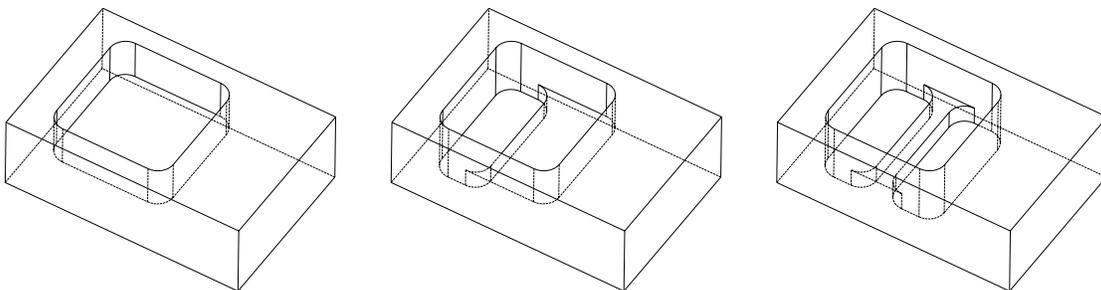

Figure 7: Machining view sequence: make a shallow pocket, then deepen the pockets to the left and right of the bridge.

It is clear, therefore, that the design view and the manufacturing/machining view may structurally diverge. There has been high quality work on this *feature conversion problem*; see [5, 7, 9, 23], and the references in [26].

We believe that the problem of *automatic* feature conversion cannot be solved completely, because the range of possible manufacturing shapes is too vast; [12]. Further complications arise from feature compounding and, particularly, from feature collision. Feature conversion can be partially automated, however, especially for many prismatic part designs, and the work on feature recognition [17, 21, 28, 29] plays a role.

When mapping design features in the CAD model onto machining features, several types of mappings can arise, [23]. They can be conceptualized by the following three situations:

1. Some design features coincide in form and geometric semantics with some machining feature. In this case, the mapping is straightforward. In general these features belong to the class of features that remove material from the workpiece. Examples include holes, slots, and steps. Note that a machining feature may also correspond to a group of design features.

2. Some design features do not exactly coincide in form and geometric meaning with any machining feature but they are closely related. In this case, the mapping has to do some additional work to recognize the machining feature. An example is a blind cut with an arbitrary closed profile and flat bottom where every edge of the bottom has been filleted. The cut can be naturally mapped onto a pocket machined with a milling tool that has an appropriate radius.

3. If there is no clear relationship between design features and any machining feature, a feature recognition process can be invoked to analyze the part's boundary. The recognition process may succeed in identifying the machining features needed to manufacture the part. Furthermore, we could rely on a manual feature identification process, supplementing automatic recognition, or as sole mechanism for feature conversion.

In the first and second situation, there is an automatic connection between design and machining features that can be formalized simply by matching feature names and shape parameters. In the third case, the machining feature recognizer must include the appropriate logic to carry out the mapping.

4.2.3   Machining Features in the Master Model

Machining features are used for several purposes, including setup, operation plan and tool path motion generation. We want them represented explicitly in the master model.

Every machining and transition feature can be defined generically by means of a set of parameters. For example, an open slot can be represented by a numerical parameter that defines the slot depth; by a profile, consisting of a set of 1-dimensional geometric elements; and by a trajectory for the sweep, again a set of 1-dimensional geometric elements. If the slot is not open, then appropriate end conditions must be defined as well. Replicated features can be defined generically by describing the element feature plus the set of parameters that define the pattern; rectangular, circular, etc. When representing these features, we do not want to duplicate information already present in the master model.

First, consider machining and transition features. If such a feature coincides with a design feature, we simply identify the design feature in the net shape in the way the CAD system has done. If the machining feature includes several design features, the machining feature is represented by a list of design feature identifiers, one for each component design feature. Finally, if the machining feature is composed of geometric elements identified individually on the net shape, and the elements do not comprise complete design features, then the machining representation should include the identifiers of these geometric elements plus the identifiers of fully included design features.

Since replicated features are composed from machining features, the same rationale applies to the description of their components. Therefore, a replicate feature will be represented as a list of machining features plus the information defining the pattern.

Note that machining features include surface finish and tolerance information. Surface finish is represented as an attribute associated with referenced elements of the net shape. The reference could be directly or via a reference to an element of a design feature, depending on the machining feature. Tolerance information is represented by identifying the corresponding tolerance feature.

4.2.4   Machining Features and Master Model Associativity

We assume that all the machining features in the part to be manufactured have been successfully generated by either an automatic or a manual process, or by a combination of both.

While design feature information originates in the CAD model, some of the machining feature information could have been derived by other clients of the master model. The deposited information includes a standard description of the net shape plus all information on design features, parameters, and constraints that the CAD system discloses. This information must be appropriately associated in the master model, and must be made persistent and consistent with all feature views supported by the master model.

We already discussed how to persistently identify elements of the net shape and how to publish their metamorphosis under CAD system edits of the net

shape, in the change protocol. Thus, associations of surface finish are easy to maintain persistently. Moreover, maintaining the information associated with machining features that correspond to one or more design features is straightforward. Finally, the maintenance of tolerance feature information has been discussed before in Section 4.1.

It remains to explain the persistent maintenance of machining features that are composed of several net shape elements that do not comprise entire design features. To do so, we need to consider the nature of the changes that arise from the manufacturing process planning work. These changes apply to all machining features whether they correspond to one or more design features or not.

When a manufacturing method is assigned to a machining feature, two different outcomes are possible: the method can be completed successfully or it fails. Some factors that affect the assignment process are material properties, geometric structure of the part, dimensioning and tolerancing values, and available tools. Examples of outputs generated by a manufacturing process assignment that fails are:

- the stock cannot be prepared
- the part cannot be clamped in a setup
- an entry face of a machining feature cannot be accessed
- the machining tool cannot perform as required
- the machining tool is not available

To illustrate the nature of changes that manufacturing process planning may require, we consider only two factors that may affect manufacturing methods assignments: the geometric structure of the part, and dimensioning and tolerancing values. Factors such as material properties are not a problem because changing them would not impact the (nominal) CAD model.

First, consider the situation where a manufacturing method cannot be assigned because of the geometric structure of the part. In general, this situation cannot be eliminated with a simple edit of the part. Instead, a major revision of the part design may be required. A typical example is that an entry face of a machining feature cannot be accessed. The process planner will report the machining feature that cannot be manufactured as well as those part geometries that preclude access to the entry face.

Consider now the situation where the assignment of a manufacturing method fails because a nominal dimension requires a tool for machining that is not available. For example, a slot may have been designed with a width smaller than the smallest mill available. The process planner will generate a report describing the machining feature and the range of nominal values for which tools are available. Here, the required geometry changes are driven by changing the respective parameter that is already present in the master model. Most

likely, there will be no fundamental change of the topology, implying that the change protocol is a graph of depth 1.

Finally consider the case where the assignment fails because tolerancing requirements are too tight and the available machine cannot perform the operation. For example, a surface finish may require a high speed mill, but the machine does not have the sufficient horse power to cut the material at that speed. The process planner will report on the performing range of the available tool (given the material and tolerance stipulated). Here, the required changes do not affect the shape of the part. Hence, editing is reduced to updating tolerance specifications already defined in the master model. They have to be coordinated with the GD&T view, and the CAD model should update its internal model if some of the tolerances originated from CAD model specifications.

In summary, the information to record the shape of the manufacturing features can be obtained by referring to design features, or else be built up by mappings from net shape and attributes on net shape elements. In the latter case, the structuring information such as parameters and dimensions may be in conflict with the design view parameters and dimensions. They are valuated initially as driven dimensions. This feature information is supplemented by GD&T features, and by surface finish and material attributes.

The changes to net shape that arise out of the machining and process planning view either are major design revisions, and then they should not be automated, or else they are changes of feature parameters that imply minimal topology changes. Where the parameters are in conflict with the design parameters and dimensions, a constraint schema reconciliation is needed as discussed in Section 5.

5 Constraint Schema Reconciliation

Both the GD&T view and the machining view may request changes of dimension and add dimensional constraints that conflict with design constraints. Both views may base this change request on features that are not design features or natural groupings of design features. Therefore, the specific dimensions they request to change may be dimensions that are driven by design parameters and dimensional constraints. We abstract the situation in the following *constraint schema reconciliation* problem.

5.1 Problem Formulation

Given a spatial arrangement of net shape elements E, consider a set of constraints C on them, such as dimensional constraints, constraints of angle, tangency, perpendicularity, and so on. We consider a maximal subset of constraints $C_0 \subseteq C$ such that the constraint schema is not overconstrained. We require that
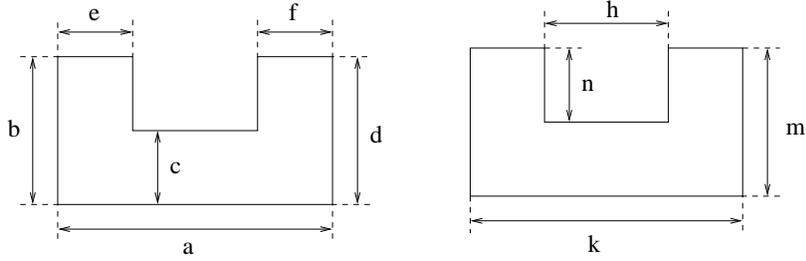
Figure 8: Net shape with two different constraint schemata

the constraints in $C - C_0$ are all dimensional constraints of distance or angle. That is, the constraints in $C - C_0$, if any, are *driven dimensions* whose value is determined by the constraints in C_0.

Similarly, we consider a separate constraint schema C' on the same net shape elements E, and a maximal subset C'_0 that is not overconstrained. Again, the constraints in $C' - C'_0$ are all dimensional, and are also driven.

By *reconciling C with C'*, we mean adjoining to the constraints in C_0 constraints drawn from C' such that the resulting, enlarged set C_1 is not overconstrained. Note that the definition of C_1 is nondeterministic. Note also that we allow adding to C_0 driven dimensions from C'.

Similarly, by *reconciling C' with C*, we mean adjoining to the constraints in C'_0 constraints drawn from C such that the resulting, enlarged set C'_1 is not overconstrained. Again we allow adding constraints of $C - C_0$, and again the process is nondeterministic. An analogous definition holds for the 2-dimensional case.

We believe that this constraint schema reconciliation problem is at the heart of making a change to the CAD model based on a view different from the design view that created it. That is, given the feature parameters and constraints drawn from a different, not necessarily compatible, application domain, the reconciliation problem points out how to translate to the design view in a procedural sense. After reviewing a simple example, we explore the implications of this conceptualization.

5.2 Example

Consider the two-dimensional net shape of Figure 8 with the two constraint schemata as shown. Assume that the left one is schema C, and call it the *design view*. C is well constrained, i.e., $C_0 = C$, assuming that all sides meet perpendicularly. The right constraint schema is C', and we might call it a *machining view*. It is underconstrained, but the constraint set $C \cup C'$ is overconstrained. Then a reconciliation of C with C' would be C. This means that every dimensional constraint of C' is driven. Conversely, a reconciliation of C' with C might

Figure 9: Constraint reconciliation problem for four points: Expressing x symbolically in terms of a, b, c, d, y is possible but is impractical.

be the set[2]

$$C_0' = \{h, k, m, n, b, f\}$$

That is, with this reconciliation, the constraints of C_0' are used to change the design view, as requested by the machining view, and the design dimensions $\{a, c, d, e\}$ are driven in that case.

Here is how the reconciliation would be used. In the initial design, the values for schema C are chosen and are used to construct the net shape. The design is then given to the machining view. In the machining view, the dimensions that are meaningful are those of C'. They are initially valuated by computing them so as to be consistent with the design view dimensions.

Now consider that the machining view requires that dimension h be changed, as it directly corresponds to a parameter of the machinery cutting the slot. We cannot translate this change easily into corresponding changes to the design view dimensions. Instead, we reconcile the schema C' with C, solve it, and valuate the remaining design dimensions as required by the solution.

5.3 Technical Issues

In the example, the schema reconciliation is essentially a one-dimensional problem because of the simplicity of shape and the constraint schemata. More generally, however, reconciliation requires a systematic approach. Thus, in 2D, the techniques of [10] can be applied to determine which constraints can be adjoined with the guarantee that the schema will not become overconstrained. For spatial constraint schemata, a more general degree-of-freedom analysis must be used.

The reader may object to the procedural approach advocated here. To understand our motivation, consider the simple planar reconciliation problem of Figure 9. Here, four points are placed with respect to each other. A symbolic approach to reconciling the two constraint schemata entails expressing the distance x as expression in the distances a, b, c, d, and y. In this simple situation, a complicated polynomial expression is obtained for x.[3] It can be solved, but it is

---

[2]If the top edges are collinear, then constraint b is redundant.

[3]An implicit expression can be derived relating all six variables. The expression is homo-
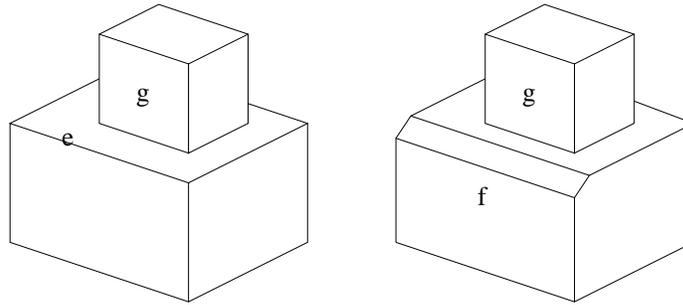
Figure 10: Constraint target transfer: A distance constraint between edge e and face g can be transferred to an equivalent constraint between face f and face g after a chamfer removes edge e.

unreasonable in view of the simplicity of the problem. We consider a symbolic approach to the problem practical only in severely limited circumstances.

For variational constraint solvers, the reconciliation problem is not difficult to solve, at least when approached procedurally as we advocate here. The reconciliation problem is more complicated for history-based CAD design systems. In these systems, the geometry is constructed sequentially, using parametric spatial and variational planar constraints. So, the reconciliation has to be approached as a spatial constraint problem.

Since parametric constraint schemata are inherently less powerful than variational ones, it would appear that the parametric spatial constraints are readily processed. Moreover, it is relatively simple to change a planar constraint schema on a profile to a corresponding spatial constraint schema, for example in the case of extrusions and revolutions.

However, in the case of general sweeps, the profile schema and the particulars of the trajectory may not be recoverable. Worse, the constraints may refer to geometric elements that exist at some point in the construction history but are not part of the net shape geometry. This possibility becomes an issue when devising algorithms for the reconciliation problem. In some cases, constraint targets can be transferred to other net shape elements. An example is shown in Figure 10. In the design, a protrusion is created on top of the block and its position on the top face is determined by a distance constraint between the edge e and the edge in the profile by which the protrusion was created. Since the protrusion is an extrusion, the dimension is equivalently between the edge e and the face g. In a subsequent step a chamfer is added, thereby losing the edge e. Since the face f is adjacent to e and is perpendicular to the top face of the block, the dimension can be transferred to a distance constraint between

---

geneous of degree 6 and has 22 terms. It has a bi-quadratic structure in x, and therefore can be solved symbolically.
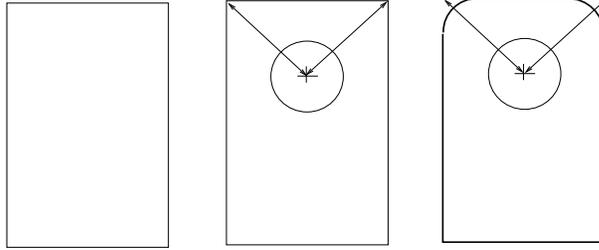
Figure 11: A poor design history

the face f and the face g.

Clearly, it is not always possible to transfer constraint references to be on the net shape, and in those cases the constraint reconciliation problem is substantially harder. However, good design practice argues that dimensions and constraints should ultimately be between net shape elements.

5.4 The Role of Design Style

For history-based CAD systems, design style is an issue because a poorly structured design history can be an obstacle to redesign.[4] Constraint reconciliation can be facilitated by design style as well.

Consider the following, simple design example illustrated in Figure 11. The base shape shown on the left is defined as a rectangle. Then the designer adds a circle and places the center with distance constraints to the two vertices of the base rectangle. Finally, the vertices, used to place the center of the circle, are rounded. This design history is problematic. In the net shape, shown on the right of Figure 11, the distance dimensions placing the center of the hole are not anchored to net shape elements, nor to datum geometry defined by the net shape. It is possible to reconcile such a constraint schema, by unwinding it across the design history, but this is not necessarily desirable because, from a product design perspective, this design is inferior and should be discouraged.

A better design is shown in Figure 12. Here, explicit construction geometry has been introduced to define the points that anchor the dimensional constraints on the hole, and that construction geometry can be captured by datum definitions based on the net shape. While more acceptable than the first design, this design still has problems when we take the inspection view of the design into account: It is difficult to measure the position of the hole accurately.

Considering geometric constraints and their reconciliation, we conclude that the following two design rules are desirable when using constraint-based, history-based CAD systems:

---

[4] According to private communication with engineers at GM Delphi Saginaw, design feature dependencies may be created that later come into conflict with the requirements for redesign.
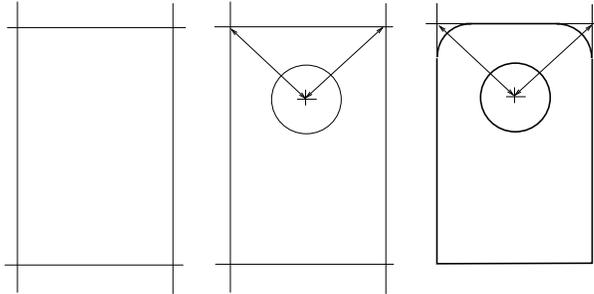
27

Figure 12: An improved design history

1. Where possible, a dimensional constraint in the design view should refer to net shape elements or to datums derived from net shape elements.

2. When defining constraints on a sketched cross section, silhouettes and projected vertices and edges should not be referred to. An exception could be made for the silhouette of a spherical surface.

When following these style rules, constraint schema reconciliation is simpler, and so is maintaining downstream views such as GD&T and inspection. Therefore, the rules promote that the net shape design already consider some of the requirements of the GD&T and inspection views.

6 Protocols for Information Flow

We assume that every information item in the master model is placed by a client system in charge of doing the primary editing of the item. Net shape, in particular, is created and edited by the CAD system. The change protocol is a mechanism to account for the fact that shape information changes affect the information that is processed and maintained by other application domains, in particular the shape aspect of the domain-specific features downstream applications may work with.

Clearly, information that has been created and is owned by a client with one view can affect many other clients, including the CAD system. For example, assume that the net shape was prepared by the CAD system using some dimensional constraint that, after analyzing geometric dimensioning and tolerancing information in the GD&T view has to be changed. Then the change, required by the GD&T view, affects the CAD view. Potentially many other application views are affected as well. The dimension changes may be indirect, since there is no guarantee that the shape design is dimensioned in a way fully compatible with the GD&T view.

This cross-dependency of information between views requires explicit protocols that ensure that information is added to the master model in an orderly

manner and is updated maintaining consistency of and between all views. In many cases, the protocols mirror the organizational structures now in place in corporations for carrying out to design changes. In terms of relational data base design, transaction serializability is an approximate counterpart to such protocols.

We restrict to three views: the CAD design view, the GD&T view, and the manufacturing view. For the manufacturing view we assume that the NC machining discussed in Section 4.2 is carried out on stock that comes from casting, forming, or is from externally manufactured supply. If the stock has shape that is functionally dependent on the net shape, for example in the case of casting, the relationship between net shape and intermediate shapes should be formalized. For example, the functional dependency in the case of machining from cast stock could include the following rules:

1. The volume to be removed by machining must exceed a minimum thickness condition. That is, we need to enlarge the net shape by a particular offset from the machined surfaces.

2. All edges should be rounded or filleted with a radius not less than a specific value.

3. Side surfaces in the mold should have a draft angle.

Such guidelines could be part of the master model, or they could be rules private to manufacturing process planning clients. The rules would advise the manufacturing engineer how to derive an intermediate shape from the net shape. Whether the intermediate shapes are deposited in the master model is up to the client. However, the net shape elements from which the intermediate form derivation keys off should be tagged in the master model so that the client can respond to relevant net shape changes.

The design begins with the CAD model of the net shape. The Brep is deposited in the master model. Then, the GD&T view is activated. If the GD&T view requires net shape changes, they must be carried out first. They would concern changes to dimensions and tolerances, and affect function, cost and manufacturing yield. After those changes have been made, possibly requiring constraints schema reconciliations, the manufacturing view is activated. This view examines the machinability. If dimensional changes are required, they are fed back to the CAD system for constraint schema reconciliation and are then handed to the GD&T view, starting a new design cycle. If the geometric structure cannot be machined, then human intervention is needed and a major redesign is in order. The information flow between the views thus should be made consistent with the flow diagram of Figure 13, which is essentially the traditional waterfall model of product design.

By analyzing the data flow at a finer level of granularity for each view, as we did in the case studies before, more can be achieved. For example, assume
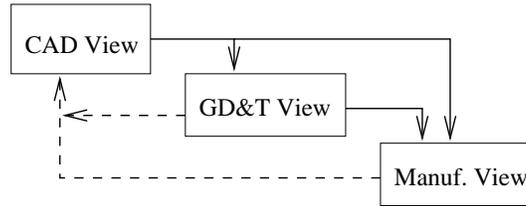
Figure 13: Information flow and feedback, waterfall approach. Feedback paths shown dashed.

that the CAD view does not utilize or define tolerancing information. This assumption is consistent with assuming that the CAD view defines the nominal net shape using only dimensions and constraints. In that case, the GD&T and the manufacturing views are the only ones concerned with tolerancing. Therefore, a feedback loop between them can be created safely and restricted to tolerance changes. Dimensional changes affecting net shape would still follow the feedback pattern of Figure 13. The new information flow between the views is shown in Figure 14 without showing explicitly the object structure and services provided by the master model server to implement this information flow and notify the clients as required.

7 Summary

Integrated product master models are of long-standing interest, but seem to be difficult to realize. We believe that a major barrier is to establish a persistent association of information structures with net shape elements, and to maintain consistency of all information under distributed updates. Further complications arise from the fact that CAD models and CAD systems have to maintain the privacy of some internal representations and methods.
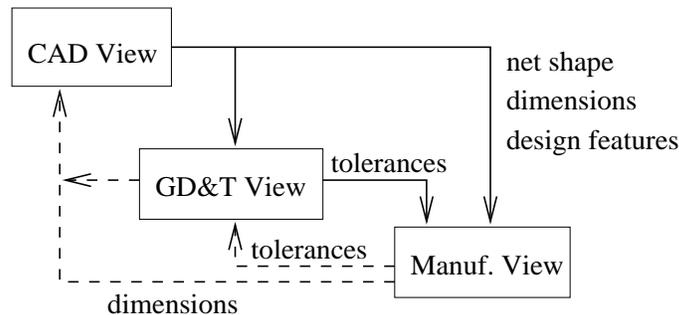

Figure 14: Information flow and feedback differentiated by information type. Feedback paths shown dashed.

30

The change protocol described in this paper offers a realistic approach to externalizing the asociation mechanism without compromising proprietary data and methods, and shifts the effort of updating domain-specific data associated with net shape to the application clients of the master model who best understand their domain.

In two case studies we have given evidence that the organization we propose here solves the information needs and offers the necessary services to maintain consistency in the master model. They illustrate how the architecture works, and point to the importance of formal information flow mechanisms as discussed. Moreover, they underscore that the feature conversion problem can be solved, in large part, by constraint schema reconciliation, a problem that is exacerbated by history-based CAD systems and ameliorated by certain design rules.

By design, our architecture is modular and constitutes a significant step to an extensible master model that does not need new modifications of the CAD system every time another application domain is added.

References

[1] ISO/DIS 10303-224. *Industrial automation systems and integration — Product data representation and exchange — Part 224*, 1997.

[2] W.F. Bronsvoort and F.W. Jansen. Multi-view feature modelling for design and assembly. In J.J. Shah, M. Mäntylä, and D.S. Nau, editors, *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, 20, chapter 14, pages 315–330. Elsevier Science B.V., 1994.

[3] C. W. Brown. Feature-based tolerancing for intelligent inspection process definition. In V. Srinivasan and H.B. Voelcker, editors, *Dimensional Tolerancing and Metrology*, volume CRTD-27, pages 249–258, Dearborn, MI, June 17-19 1993. ASME.

[4] J.J. Cunningham and J.R. Dixon. Designing with features. the origin of features. In V.A. Tipnis and E.M. Patton, editors, *Computers in Engineering Conference and Exhibition*, volume 1, pages 237–243, San Francisco, 1988. ASME.

[5] K.J. de Kraker, M. Dohem, and W.F. Bronsvoort. Multiple-way feature conversion. opening a view. In M. Pratt, R.D. Siriram, and M.J. Wozny, editors, *Product Modeling for Computer Integrated Design and Manufacture*, pages 203–212, London, UK, 1997. Chapman and Hall.

[6] K.J. de Kraker, M. Dohmen, and W.F. Bronsvoort. Multiple-way feature conversion to support concurrent engineering. In C. Hoffmann and

J. Rossignac, editors, *3rd Symp. on Solid Modeling and Applications*, pages 105–114. Salt Lake City, UT, 1995.

[7] K.J. de Kraker, M. Dohmen, and W.F. Bronsvoort. Feature validation and conversion. In D. Roller and P. Brunet, editors, *CAD Systems Development*. Springer Verlag, Heidelberg, 1997.

[8] K.J. de Kraker, M. Dohmen, and W.F. Bronsvoort. Maintaining multiple views in feature modeling. In *Fourth Symposium on Solid Modeling and Applications*, pages 123–130. ACM Press, 1997.

[9] M. Dohmen, K.J. de Kraker, and W.F. Bronsvoort. Feature validation in a multiple-view modeling system. In *16th ASME International Computers in Engineering Conference*, Irvin, NY, USA, 19-22 August 1996. ASME.

[10] I. Fudos and C.M. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics*, 16(2):179–216, April 1997.

[11] J. Guildford and J. Turner. Advanced tolerance analysis and synthesis for geometric tolerances. In V. Srinivasan and H.B. Voelcker, editors, *Dimensional Tolerancing and Metrology*, volume CRTD-27, pages 187–198, Dearborn, MI, June 17-19 1993. ASME.

[12] S.K. Gupta, W.C. Regli, and D.S. Nau. Manufacturing feature instances: Which one to recognize? In C. Hoffmann and J. Rossignac, editors, *Third Symposium o Solid Modeling and Applications*, pages 141–152, Salt Lake City, Utah, May 17–19 1995. ACM Press.

[13] G. Henzold. *Handbook of Geometrical Tolerancing*. John Wiley & Sons, Chichester, UK, 1995.

[14] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann, San Mateo, Cal., 1989.

[15] C.M. Hoffmann and R. Joan-Arinyo. Symbolic constraints in constructive geometric constraint solving. *Journal of Symbolic Computation*, 23:287–300, 1997.

[16] F.-L. Krause, E. Rieger, and A. Ulbrich. Feature processing as kernel for integrated CAE systems. In *Feature Modeling and Recongition in Advanced CAD/CAM Systems*, volume II, pages 693–716, Valenciennes, France, May 1994. IFIP.

[17] L.K. Kyprianou. *Shape Classification in Computer Aided Design*. PhD thesis, University of Cambridge, UK, 1980.

[18] J.D. Meadows. *Geometric Dimensioning and Tolerancing*. Marcel Dekker, Inc., New York, 1995.

[19] M.E. Newell and D.C. Evans. Modeling by computer. In J.J. Allan, editor, *IFIP Working Conference on Computer-Aided Design Systems*, pages 291–297, Austin, TX, USA, 12-14 February 1976. IFIP, North-Holland Pu. Co.

[20] D. Roller and P. Brunet, editors. *CAD Systems Development. Tools and Methods*, Berlin, 1977. Springer-Verlag.

[21] H. Sakurai and C.-W Chin. Definition and recognition of volume features for process planning. In J.J. Shah, M. Mäntylä, and D.S. Nau, editors, *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, 20, chapter 4, pages 65–80. Elsevier Science B.V., 1994.

[22] O.I. Semenkov. An experimental cad/cam system. In D. McPherson, editor, *3rd Inrtenational IFIP/IFAC Conference on Programming Languages for Machine Tools*, pages 397–403, Stirling, Scotland, 15-18 June 1976. IFIP/IFAC, North-Holland Pu. Co.

[23] J. Shah, D. Hsiao, and L. Leonard. *A Systematic Approach per Design-Manufacturing Mapping*, volume Geometric Modeling for Product Realization, chapter B-8. Elsevier, 1993.

[24] J.J. Shah. Feature transformations between application specific feature spaces. *Computer Aided Design*, 5(6), 1989.

[25] J.J. Shah, M. Mäntylä, and D. Nau. Introduction to feature based manufacturing. In J.J. Shah, M. Mäntylä, and D.S. Nau, editors, *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, 20, chapter 18, pages 1–11. Elsevier Science B.V., 1994.

[26] J.J. Shah, M. Mäntylä, and D.S. Nau, editors. *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, 20. Elsevier Science B.V., 1994.

[27] V. Srinivasan and R. Jayaraman. Geometric tolerancing II: Conditional tolerances. *IBM J. of Res. and Dev.*, 33:105–124, 1990.

[28] J.H. Vanderbrande. *Automatic Recognition of Machinable Features in Solid Modeling*. PhD thesis, Computer Science Dept. and Institute for Robotics and Intelligent Systems, University Southern California, 1990.

[29] J.H. Vanderbrande and A.A.G. Requicha. Geometric computation for the recognition of spatially interacting machining features. In J.J. Shah,

M. Mäntylä, and D.S. Nau, editors, *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, 20, chapter 5, pages 83–106. Elsevier Science B.V., 1994.

[30] ANSI Standard Y14.5M. *Dimensioning and Tolerancing*. Am. Soc. of Mech. Engr., New York, NY, 1994.

[31] C. Zozaya-Gorostiza, C. Hendrickson, and D.R. Rehak. *Knowledge-Based Process Planning for Construction and Manufacturing*. Academic Press, 1989.