Incorporating Knowledge of Geometric Dimensioning and Tolerancing into a Feature-Based CAD System[1]

Jeremy F. Jacobsohn*#
Gerald M. Radack+#
Francis L. Merat*#

*Department of Electrical Engineering and Applied Physics
+Department of Computer Engineering and Science
#Center for Automation and Intelligent Systems Research
Case Western Reserve University
Cleveland, Ohio 44106

Abstract

A system is described which aids a designer in assigning dimensions and tolerances to designs of mechanical parts. The system is constructed using an object-oriented approach. The designer designs using geometric "features" such as slots and holes. Associated with the object for each feature is knowledge about appropriate ways to dimension and tolerance that feature. The tolerancing system used is unambiguous, and is similar to GD&T, the standard system used in industrial practice.

1. Introduction

This paper describes some experiments aimed at incorporating high-level knowledge into Computer Aided Design (CAD) systems for geometric design of mechanical parts. Specifically, the knowledge to be incorporated is that of functionally oriented dimensioning and tolerancing.

CAD systems represent parts using geometric primitives—such as cones and cylinders— and/or mathematical surfaces—such as quadric surfaces. All of these primitives describe ideal shapes. On the other hand, actual manufactured parts are imperfect. Dimensioning and tolerancing is necessary to plan the appropriate manufacturing process to achieve a functional part, and to decide whether a manufactured part is acceptably close to the designed ideal.

Current commercial CAD systems either ignore tolerancing completely or provide a mechanism for adding tolerances using the traditional plus/minus system to a drawing. The designer is left with the responsibility of ensuring that the tolerances are complete and consistent. The systems do not ensure that tolerances are reasonable or meaningful.

1.1 Approach

Our approach is to embed knowledge about function of certain common part features, *e.g.*, holes—as well as of the manufacturing processes—into the CAD system itself, in order to provide assistance to the designer in assigning dimensions and tolerances. The representation for dimensions and tolerances that we use is in the spirit of current U.S. standards on dimensioning and tolerancing. This tolerancing system is considerably more descriptive—and also more complex—than the traditional system.

This approach ensures that the output drawings produced by the system can be understood and interpreted by Quality Assurance (QA) engineers in industry. This work is embedded in a larger design system which includes additional features to aid the QA engineer, including the automatic generation of plans for the inspection of the part, and automatic interpretation of the results of the inspection.

1.2 Motivation

Advances in CAD systems have resulted in designs that are too complex for previous manual tolerancing practices. Even an experienced designer may have difficulty correctly assigning tolerances to a drawing. There is also a tendency for a designer to over-tolerance in order to be "safe." This leads to unnecessary manufacturing costs.

In addition, many designers, particularly at smaller companies, are not trained in modern dimensioning and tolerancing practice. With expert assistance from a knowledge-based AI advisor, more designers could use GD&T more effectively with less formal training. Another important benefit would be that the resulting designs would be more consistent, and easier for the manufacturers and inspectors to interpret.

1.3 Context of the work

This work is part of a larger project, sponsored by the U.S. Air Force, to develop a "Rapid Design System" (RDS). RDS is intended to support the fast and economical design of mechanical parts. RDS is being developed with the cooperation of a machine shop which specializes in (1) custom modification of aircraft, *e.g.*, to add new instrumentation, and (2) producing replacement parts which are not available from the manufacturer. In some cases, prints are no longer available for such parts, so they must be redesigned. Both types of jobs are characterized by low production runs—sometimes quantities of one. Often, the elapsed time from initial request to the need for the finished part is critical to the scheduling of a test mission. In such situations, reducing the design time can significantly lower costs.

The objective of the RDS project is to speed the design process by providing (1) an intelligent CAD interface which enables the designer to get his or her design into the computer faster than current systems allow; (2) integrated tools to check a design for manufacturability and inspectability.

[1]This work was supported in part by the U. S. Air Force under contract F33615-87-C-5250.

0-8186-1966-X/90/0000/0152/$01.00 © 1990 IEEE

1.4 Structure of this paper

Sections 2, 3 and 4 provide background material on dimensioning and tolerancing, feature based design, and parametric design systems, respectively. Sections 5 describes the work done, and Section 6 contains conclusions and areas for future work.

2. Dimensioning and Tolerancing

Dimensions and tolerances are generally prescribed on engineering drawings using either the traditional plus/minus system, or a system specified in a modern drawing standard.

Any of these methods can provide enough information for the manufacturer to use in producing a part that attempts to satisfy the designers intent. It is important, however, that the compliance of the part to the specification be verifiable. One way to verify a part is to try it out—perhaps constructing the assembly of which it is a sub-part and testing that. This method does not work well in practice, because of cost or safety constraints that might preclude exhaustive testing. Imagine using this approach to verify components of a rocket engine, for example. Another important function of the dimensions and tolerances on the drawing, then, is to provide information for the pre-assembly inspection and verification of the manufactured part.

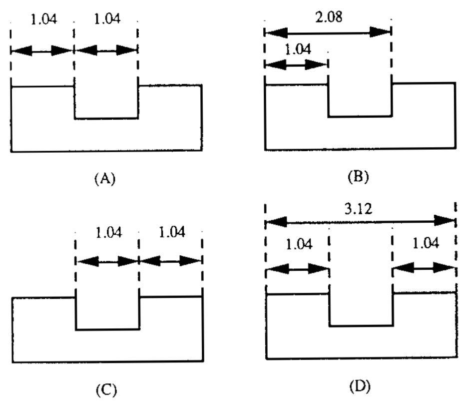


Figure 1: Alternative ways to dimension a feature.

The dimensioning and tolerancing of a drawing also conveys high-level information about the desired characteristics of the part. Figure 1 shows three different ways to indicate the position and width of a slot, by providing dimensions relative to different lines on the drawing. For a perfect-form part, the methods are all equivalent. On a real part, however, manufacturing and measurement inaccuracies will make the real dimensions diverge from the designed ones. The three dimensioning schemes shown each emphasize a particular relationship between features on the part. It may be that the relationship between the left edge of the part and the left edge of the slot is the only important quantity, making (A) the appropriate format; in another design it might be the right-hand edges that are important. A good dimensioning and tolerancing system will provide a rich vocabulary for making such distinctions. The two accepted dimensioning and tolerancing methods are described in the following sections.

2.1 Traditional system

The traditional formalism for tolerancing is the annotation of each dimension on the drawing with a ± allowance that describes the maximum oversize and undersize variations. A partially complete drawing using this system is shown in Figure 2, a block with four holes. Notice that the dimensions of the hole pattern are given as 2.000 m (this is the *nominal* value), with an allowance of 5 mm on a side. No measurement procedures are provided, which leads to measurement confusion. Are the hole positions measured relative to each other, or to the theoretical positions of the other holes derived from measurements of the block? Also, the tolerance zones defined by the allowances are square, while the designers intent might have been to place the holes within a particular radius of the intended center—a circular tolerance zone.

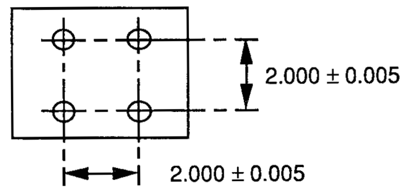


Figure 2: Example of ± dimensioning.

2.2 Tolerancing with GD&T

The current state of the art in dimensioning and tolerancing—called "Geometric Dimensioning and Tolerancing (GD&T)—is defined by the American National Standards Institute in ANSI Y14.5M-1982. This standard provides uniform requirements for presenting and interpreting tolerancing information on mechanical drawings, and provides a powerful set of tolerancing primitives. GD&T practices are followed by many major U.S. manufacturers, and a very similar ISO standard is followed internationally. When correctly used, GD&T resolves some measurement ambiguities associated with the traditional ± dimensioning [9]. Unfortunately, the GD&T standard relates to 2-d drawings, not 3-d CAD models.

With GD&T, a designer specifies the allowable variation in manufactured instances of the part. The notation allows the designer to relate these allowances to the required functional characteristics of the part. For instance, assume the function of a circular pattern of bolt holes is to provide a mounting base for an axially symmetric flange. Then, the GD&T annotations will specify the center and radius of the hole pattern, and the angular distance between the holes around the pattern, without specifying the actual Cartesian *xy* location of the holes on the base. The intent-based tolerancing description provides valuable communication between the designer, the manufacturer, and the Quality Assurance engineer who inspects the final part.

GD&T, then, is a useful and accepted way of embedding design knowledge into a mechanical drawing. In trained hands it can be concise and descriptive, otherwise it can become verbose and obscure. GD&T is like a natural language; each user has a unique idiom. A considerable amount of training and experience is needed to use GD&T effectively, by both the designer and the QA engineer.

2.3 GD&T constructs

The descriptive elements of GD&T fall into three basic classifications, which are described in the following paragraphs.

Datum. A datum provides an ideal surface or line used in forming an explicit coordinate system for a particular measurement. The datum is 'attached' to a particular surface, or an axis of symmetry, on the part. When the real part is manufactured, the position of the datum, which is an abstract idealized entity, is calculated in a specific way using the measured position of the real object to which it is attached. Some measurements require a system of three datums to provide a 3-d coordinate reference frame, while other measurements use only one datum, and measure a distance in a 2-d planar projection.

Callout. Callouts are notational blocks, written inside a rectangular outline, which provide several kinds of tolerancing information. Callouts can be considered to be sentences, in the language of GD&T. They are made up of a verb specifying the type of GD&T constraint to be imposed, the nominal dimension and allowable variation for the measured value, and references to datums as required. There are also modifiers on the references which are used, in sophisticated applications, to provide additional 'bonus' tolerances when conditions will permit the assembly to function with larger allowances. Figure 3 shows a sample GD&T callout.

The callout can refer to the form of an individual feature, the profile of a feature (or a group of features), or relationships between features. The callout is attached to a particular feature by a line on the engineering drawing.

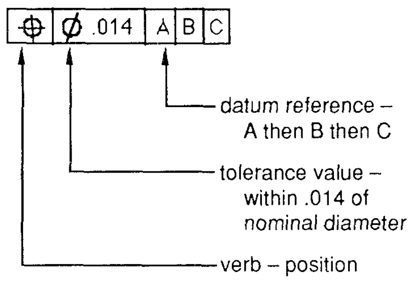


Figure 3: An example callout.

Note. Notes appearing on an engineering drawing can specify default tolerances on dimensions which are not otherwise toleranced. Notes can also provide additional information to the inspector, such as whether dimensions are to be measured before or after paint or coatings are applied.

2.4 Example

The drawing in Figure 4(a) shows a bolt hole pattern, intended for attachment to another part (e.g a pipe flange), which is axially symmetric. There are two datums, one through the axis of the bolt hole pattern and the other on the bottom surface of the block. The diameter of the bolt hole pattern is 32 mm, the angular displacement is 60 degrees, and the positional tolerance of each hole is 0.1 mm relative to the center (datum A) and then the surface (datum B). The diameters of the holes are 6 mm, with a maximum allowable size of 6.2 mm and a minimum of 5.9 mm. The positional tolerances have a modifier, signified by the circled "M", which signifies a bonus tolerance on the hole position, which grows with the holes (a larger hole can be a little farther off position and still allow passage of the attachment bolt). For more information on GD&T, see the ANSI standard [1], or a GD&T textbook.

Another part to be designed and toleranced is shown in Figure 4(b). Functionally, this is a cover plate for a mechanism with four clearance holes for mounting bolts. As shown all dimensions are basic part dimensions which are specified in the various feature descriptions. The hole diameter is shown as it would appear on a properly dimensioned drawing with the 4× specification to apply it to all four holes, a diameter of 0.262 inches and an asymmetric diametral tolerance of +0.005 and - 0.000 inches. In the RDS the diameter is an intrinsic parameter of the hole; the tolerance is an intrinsic parameter of the diametral tolerance feature associated with the hole. Note that this is a

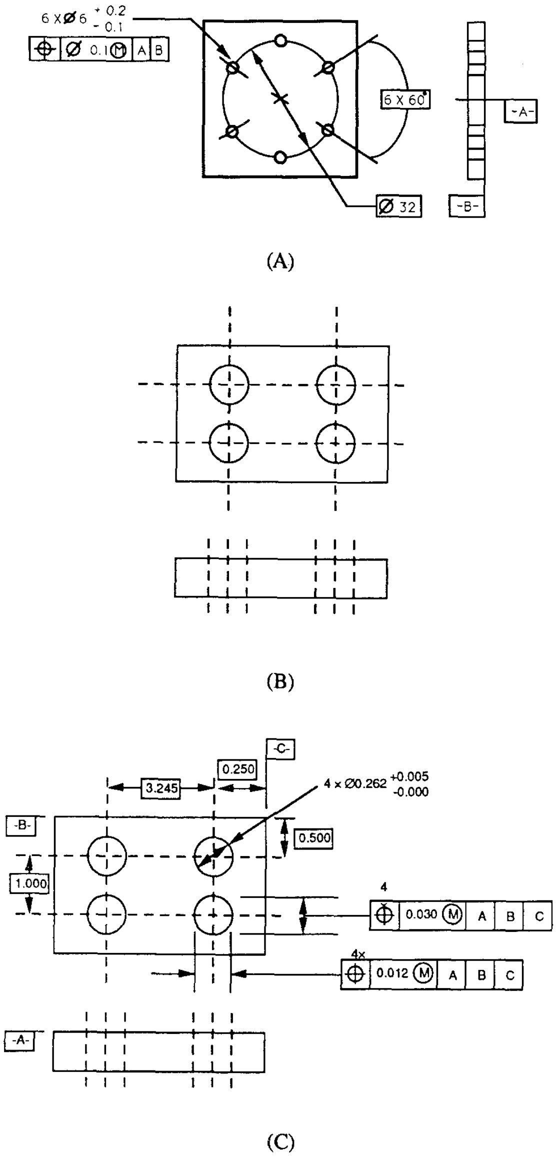


Figure 4: GD&T Examples

diametral tolerance as indicated by the presence of the ∅; the position of the hole is toleranced separately using GD&T callouts. The position is toleranced asymmetrically about the basic position of the hole—0.030 inches in the vertical direction and 0.012 inches in the horizontal direction. A symmetric, or diametral, positional tolerance would have been indicated by a ∅ preceding the actual tolerance value. In terms of the old ± tolerancing scheme these might be thought of as ±0.015 and ±0.006 inch tolerance zones respectively; however, their interpretation is unambiguously specified by the associated GD&T positional callouts as shown in Figure 4(c).

3. Feature-Based Design

Much interest has been focused recently on "design with features." See, for example, [2], [5], and [10]. In the context of computer aided design, features are "chunks" of the part which have some special meaning to the designer. Feature-based design is based on the premise that designers should be able to specify a part in terms of features which are meaningful to them instead of having to specify geometric primitives. The most obvious type of features is that of "form features." Form features represent certain geometric configurations on the surface of the part. Examples of form features include holes, slots and ribs. These various features have entered the designers' vocabulary either because they commonly arise in manufacturing (*e.g.*, a hole is the result of a drilling operation) or because they have some special function (*e.g.*, a rib is used to strengthen a thin section of a part).

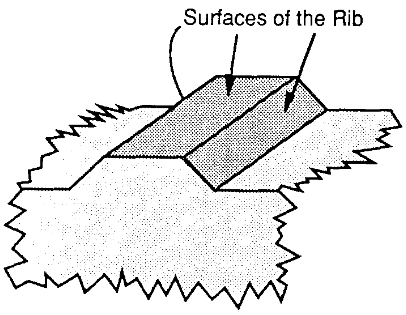


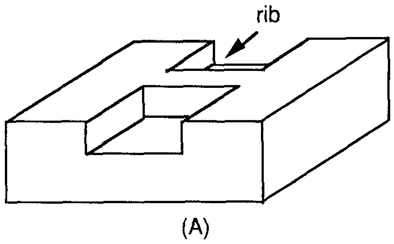
Figure 5: A rib.

A form feature can be thought of a collection of surface primitives which satisfy some which have a predetermined relationship both with each other and with the rest of the part's surface primitives. For example, a rib can be thought of as three flat surfaces, two of which join the third along parallel edges, and which meet surrounding surfaces of the part at two other parallel edges. See Figure 5 for an illustration. Note that this definition is "incomplete" in the sense that the geometry of the ends of the rib are left unspecified. Thus, Figures 6(a) and 6(b) show valid ribs, even though the geometry of the ends are different.

It should be noted that designers also use the term "feature" to refer to non-geometric properties of an object, for example, its surface finish. We will not deal with such features in this paper.

4. Parametric Design Systems

In general, CAD systems provide (1) a data structure for representing models, and (2) an interactive environment for creating and modifying the models. In a conventional system, the data structure contains elements representing graphical primitives—both geometric and textual—and possibly some elements to group other elements. These grouping operators in effect create a part-subpart hierarchy. In order to create a design, geometric primitives must be positioned correctly with respect to each other. Each primitive in the model can be considered as an instance of a generic primitive, *e.g.*, a particular rectangle in the model is an instance of the generic concept of "rectangle." Thus, it is necessary to provide positioning and shape information for each instance. If the system supports grouping, then groups can be moved as a unit. This is normally done by storing a transformation matrix with each primitive or group. These transformations are stored as concrete (numerical) values. One of the contributions of parametric design systems is to make these transformations functions of parameters supplied by the designer.



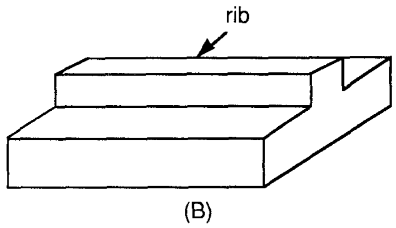


Figure 6: Two ribs.

Parametric design systems, as typified by the Concept Modeler [6] and Icad [3] generalize the above scheme by providing a language which can be used to express relationships between subparts in a model. An object-oriented approach is used in [7]. Three distinct relationships can be expressed: an inheritance relationship for objects, a part-subpart relationship for primitives or groups, and geometric relationships between primitives and groups. The instance variables of the objects are called parameters, and can be used by the designer—through statements in the definition language—to control relationships listed above. The systems keep track of dependencies between objects. Objects are only created when they are needed. Whenever the objects become invalid, due to changes in other parameters, they are marked as invalid, and are recreated the next time they are needed. This scheme is called "dependency backtracking." Note that in the above discussion, "objects" can include primitives, groups, parameters, and transformations.

5. Incorporation of GD&T into the System

5.1 Current practice

Conventional CAD systems are capable of incorporating GD&T callouts as graphical elements on drawings. At the lowest level of support, users can simply draw the GD&T symbols using the system's own drawing primitives, such as circles and letters. At a higher level of support, callouts can be automatically drawn from a template in which the user specifies the GD&T verb, the tolerance, datum references etc., with the CAD system drawing the GD&T callout frame. This capability can be added on to conventional systems such as AutoCAD or Intergraph, which have embedded programming languages that have access to drawing tools, as well as facilities for querying the user.

Designers can even build a CAD database of GD&T callouts and of design features which have appropriate GD&T already drawn. These might need only to have the dimensions adjusted for the particular application. This home-brew approach to the incorporation of dimensioning and tolerancing (as well as parametric feature-based design) might be very useful to a designer who works with a very restricted class of parts, all with similar geometry and function.

5.2 Knowledge-based approach

Ultimately, the CAD system should incorporate knowledge of GD&T semantics. To best aid the designer, the system should assist in the design of the GD&T annotations by providing expert suggestions, guidance and criticism. The achievement of such capabilities in a parametric, feature-based environment is especially promising. Here, knowledge of feature characteristics and applications can drive the incorporation of GD&T.

For example, let us consider the "bolt hole pattern" shown in Figure 4. The pattern shown is toleranced with respect to datums A and B implying that the angular separation between the holes is important, but that the exact position of the holes with respect to the square flange (which is not specified) is not. In general, such bolt hole pattern features may require precise alignment with respect to a key which will necessitate a different tolerancing scheme than that shown in Figure 4. The symmetry of the bolt hole pattern and of the feature to which it is attached can be tested to determine the correct tolerancing approach. In Figure 6, the asymmetric block in which the symmetric bolt hole pattern is found would cause the system to inquire of the designer as to whether an orientation for the bolt hole pattern should have been toleranced.

Similarly, the GD&T entities, also represented as programming objects, can check the design to which they are attached. For example, a GD&T position callout might reference three datums, intended to form a coordinate reference frame. The frame may have been valid when the callout was created, but due to changes in the part design might be invalidated, perhaps no longer forming a three dimensional basis. The callout could be alert to such changes, by requesting to the related design features and datums that they send the callout a message informing it of any changes made to them.

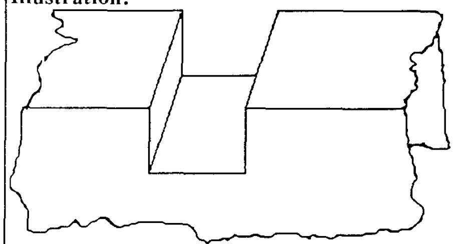
5.3 Alternative GD&T descriptions

Each feature type is defined as an object class in the parametric design system. Parameters (*i.e.*, instance variables) defined in the object class specify the dimensions and tolerances. For example, a perpendicular flat through slot is specified in terms of the parameters shown in Figure 7. (Those parameters shown in brackets are not relevant to this particular discussion.) The supplied parameters provide enough information for a default tolerance description of the part. It is important to realize that these parameters do not imply only one possible GD&T representation, they provide *minimum* information required to fully specify the slot. We refer to this as the *intrinsic parameterization* of the design feature. Associated with the slot feature are GD&T objects which either directly implement the dimensions and tolerances shown in the parameters, or describe them in a completely different way if that is appropriate for the particular function of the slot. These objects are known as the *extrinsic parameterization*. An example extrinsic parameterization is shown in Figure 8.

Class name: perpendicular-flat-through-slot

Description: An elongated depression, running clear from one edge of the material to the opposite edge, with side walls perpendicular to the material surface and a bottom perpendicular to the side walls. The sides and bottom are flat.

Illustration:



Attributes:

width	2.350 ± .003
depth	1.759 ± .003
geometry	<as required>
attachment	<reference to another design object>
position	<spatial transformation matrix>
filleted?	False
fillet_radius	n/a
finish	<finish_spec>
function	<function_spec>

Figure 7: Intrinsic parameterization.

For each design feature, we can provide the default GD&T description and also offer on demand information about other available tolerancing approaches. Since the system has information about multiple features, it can make intelligent suggestions based on the context of the current feature. It is important, however, not to constrain the designer to use one of the suggested methods—it must be possible to add GD&T features at his request and according to his preference. Designers would be unhappy if we limit their creativity; this is intended to be an enabling technology, not a straitjacket.

To return to the example earlier of the three different ways to dimension a slot, we could provide a GD&T parameterization which is appropriate for each dimensioning. The GD&T objects for (A) would consist of two datums, one attached to the left side of the outer block (into which the slot is cut) and the other to the left wall of the slot. One position callout would place the slot's left wall relative to the first datum, another position tolerance would place the slot's right-hand wall, relative to the second datum. The one for 2.0 (B) would assign a datum at the left side of the outer block, and reference it in two position callouts, one attached to the left wall of the slot and the other to the right-hand wall. The system contains rules to determine which method is more appropriate, given the function of the slot.

5.4 Design agents

The knowledge required to correctly specify GD&T, and to take advantage of GD&T's expressive power, is distributed among design features, GD&T features, and the user himself. The system under development allows this distribution by representing each feature as an independent design agent. These

Class name: perpendicular-flat-through-slot **Attributes:** width nil depth nil geometry <as required> attachment <reference to parent object>	position <spatial transformation matrix> filleted? False fillet_radius n/a finish <finish_spec> function <function_spec>
Class name: datum **Attributes:** name "A" attachment <left wall of the slot>	**Class name:** position callout **Attributes:** attachment <right wall of the slot> nominal 2.350 tolerance 0.003 datum ref "A"
Class name: datum **Attributes:** name "B" attachment <top of the block>	**Class name:** position callout **Attributes:** attachment <bottom of the slot> nominal 1.759 tolerance 0.003 datum ref "B"

Figure 8: Extrinsic parameterization.

agents cooperate within the object-oriented programming environment, where they can send messages to each other, and communicate with the user when necessary.

Each GD&T feature owns a list of design agents, implemented as object methods. These methods are run when the GD&T object is first instantiated. Running the method establishes, as a side effect, a chain of dependencies on data that is referenced by the design agent. This chain is double-linked, so each data item in the system has an I-depend-on list as well as a depends-on-me-list. Thus, when a change is made to the part design the system can evaluate the dependency reference chains and re-run design agents as required.The "on demand" update capability is important, because in a large design it would cause an unacceptable delay to check the validity of the constraints on every object in the system—this way only those objects that may be affected are checked. The dependency backtracking capability of the Concept Modeller can be used for this purpose.

As an example, consider a design agent that operates on position tolerances of holes. This agent checks the thin-wall minimums of the design, to verify that repositioning of the hole cannot cause it to move too close to the edge of the part. This method establishes dependencies on the size of the feature into which the hole is drilled, and also on the position and radius of the hole. If any of these items are modified by the designer the design agent will be reactivated, and the thin-wall condition reevaluated.

The design agents are also available on demand by the user, for updating or rechecking. The user's display has menus and icons to invoke them. As stated before, the designer has considerable freedom in annotating the design. This might lead to situations where the design contains, temporarily, inconsistencies. There are actually two modes of operation of the system. In one mode, the system will automatically attempt to resolve inconsistencies arising out of changes. As a simple example, if the intrinsic attribute for the width of a slot is changed, a corresponding position callout will be modified. On the other hand, the designer might intentionally introduce such an inconsistency during a reorganization of the design, with the intention of fixing it later. Therefore, there is also a mode in which the system reminds the designer of an inconsistency but does not force it to be resolved. Again, placing limits on the designer's freedom would hinder the acceptance of this design environment.

5.5 Design rules

The design agents are the most general form of checking that the system provides. In many cases, however, their full generality is not needed and it is desirable to perform design checks that can be described declaratively as rules. This is implemented in the system by providing a general design agent that is passed, as a parameter, a list of rules which are stored in a rule base within the part model.

There are two broad categories of rules in our system- design constraints and inspection planning rules. The former are mainly used for semantic checks on the GD&T callouts, the latter choose inspection modalities and provide an inspection plan for inspecting that particular tolerance. The inspection planning rules are described in more detail in [8].

Design constraints take up where the user interface leaves off in terms of validating the application of GD&T. The interface provides syntactic checking, for example it requires that any datum referenced by the rule exists somewhere in the part design. The design constraints provide semantic knowledge. To continue the datum example, a semantic check can determine whether the datum references provide a valid coordinate reference frame as defined by the ANSI standard (i.e. fixes the part in space). Another simple semantic check determines whether the tolerance on a feature of size is larger or smaller than allowed by standards of the manufacturing shop, or by physical realizability—such as a tolerance so large that it would allow a hole to be of zero size and disappear.

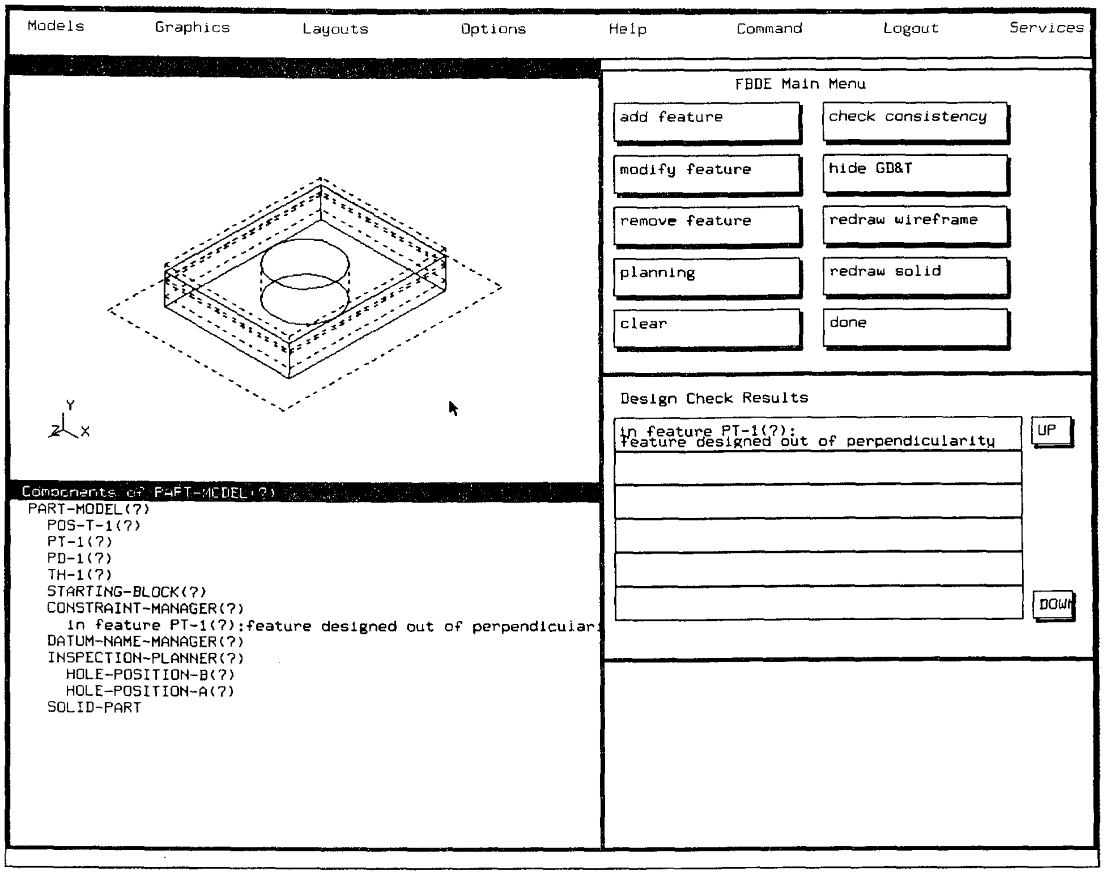


Figure 9: Sample Screen View

Design constraints can relate GD&T features to each other. They can detect redundancies (e.g. two radius constraints on the same hole), or cases where one tolerance subsumes another (e.g. a cylindricity tolerance and a circularity tolerance of the same value applied to the same hole).

The design agent that processes rules cannot make changes to the design in response to the left-hand-side conditions of the rule being true; the agent's role is just advisory. For inspection planning rules the design agent passes the result on to the global inspection planner, which is another subsystem still under development. For design constraints the system places a notification in a list of constraint violations which is displayed to the user for browsing. The notice stays in the list until the user modifies the design in such a way that the dependency manager causes a recheck of the rule and finds that the left-hand-side conditions are no longer true.

The screen display in Figure 9 shows an example of a notification. The middle pane on the right side of the user's display is labelled "Design Check Results." There is one notification visible: an indication that in feature PT-1 (which is the name of a perpendicularity tolerance feature in the part model of the example) the selected face to which the tolerance is applied is not perpendicular to the primary datum of reference, and is thus out of tolerance in the part as designed. This notification appeared as the result of executing a design agent on a rule that compares the normal direction of the face to which the tolerance is attached to the normal direction of the datum referenced.

6. Summary and Conclusions

We have described a system which assists a designer in the design of mechanical parts. The specific contribution of this work is to incorporate expert knowledge of Geometric Dimensioning and Tolerancing into a design system. Geometric features are used to guide the system in determining appropriate choices for dimensioning and tolerancing. The designer has the freedom to select from a series of choices, and even override completely the suggestions of the system. There is no attempt to force a particular style of dimensioning and tolerancing on the designer.

The system is currently being implemented The partial implementation operates in a limited domain of features, and is currently being used by part designers to evaluate its usefulness. These designers, who have substantial experience with commercial CAD products, are enthusiastic about the potential of this system.

References

[1] American National Standard Y14.5M–1982, "Dimensioning and Tolerancing," American Society of Mechanical Engineers, New York, 1982.

[2] J. R. Dixon, "Expert Systems for Mechanical Design: Examples of Symbolic Representations of Design Geometries," *Engineering with Computers* **2**, 1987, 1-10.

[3] J. G. Gregory, "Rule-based definition of geometric models," SIAM Conference on Geometric Design, Tempe, Ariz., Nov. 1989

[4] E. C. Libardi, J. R. Dixon, M. K. Simmons, "Designing with Features: Design and Analysis of Extrusions as an Example," ASME Paper No. 86-DE-4.

[5] S. L. Luby, J. R. Dixon, M. K. Simmons, "Designing with Features: Creating and Using a Features Database for Evaluation of Manufacturability of Castings," *Proceedings of the ASME Computers in Engineering Conference,* Chicago, July 1986.

[6] M. P. Lukas and R. B. Pollock, "Automated Design Through Artificial Intelligence Techniques," Artificial Intelligence and Advanced Computer Technology Conference, Long Beach, May 1988.

[7] M. P. Lukas and M. I. Meerbaum, "Improving Productivity by Capturing and Automating Design/Manufacturing Knowledge," technical report TP88-20, Bailey Controls, Wickliffe Ohio, 1988.

[8] Radack, G. M., and F. L. Merat, "The Integration of Inspection into the CIM Environment," 23rd Hawaiian International Conference on Systems Science, Vol. 2, Kailua-Kona, 1990, 455-462.

[9] Requicha, Aristides A. G., "Toward a Theory of Geometric Tolerancing," *International Journal of Robotics Research* **2**, 4, 1983, 45-59.

[10] Requicha, Aristides A. G., "Representation of Geometric Features, Tolerances, and Attributes in Solid Modelers Based on Constructive Geometry," *IEEE Journal of Robotics and Automation* **RA-2**, 1986, 156-166.