Int J Adv Manuf Technol (2005) 26: 405–414
DOI 10.1007/s00170-004-2195-3

ORIGINAL ARTICLE

Jian Gao · De Tao Zheng · Nabil Gindy · Doug Clark

Extraction/conversion of geometric dimensions and tolerances for machining features

Received: 28 January 2004 / Accepted: 31 March 2004 / Published online: 28 July 2004

Abstract It is important for a feature-based system to preserve feature integrity during feature operation, especially when feature interaction occurs. The paper presents a feature conversion approach to convert design features used in a design model into machining features for the downstream applications. This process includes both form features (geometric information) and non-geometric features conversion. Most researchers have concentrated on geometric information extraction and conversion without tackling the important problem of non-geometric feature information. This paper focuses on the extraction and conversion of feature geometric dimensions and tolerances (GD&T) for downstream machining application.

The main barrier to the integration of a feature-based CAD/CAPP/CAM system – feature interaction – is discussed in this paper, which alters design features in their geometries and non-geometric information. How to identify and validate these feature dimensions and tolerances is one of the key issues in feature interaction conversion. The development of robust methodologies for preserving feature integrity for use in process planning application is the main thrust of the work reported in this paper.

Keywords CAD/CAPP/CAM integration · Feature conversion · Feature interaction · Geometric dimensions and tolerances (GD&T) · Machining features

J. Gao (✉) · N. Gindy
School of Mechanical, Materials, Manufacturing Engineering and Management,
The University of Nottingham,
University Park, Nottingham NG7 2RD, UK
E-mail: Jian.Gao@Nottingham.ac.uk
Tel.: +44-115-951-4116
Fax: +44-115-951-3800

J. Gao · D.T. Zheng
School of Mechanical and Electronic Engineering,
Guangdong University of Technology,
729 East Dongfeng Road, 510090 Guangzhou, P.R. China

D. Clark
School of Mathematical and Computer Sciences,
Heriot-Watt University,
Riccarton, Edinburgh EH14 4AS, UK

1 Introduction

A main objective in modern design and advanced machining process is to achieve a significant reduction in product lead time through seamless integration between the various design and process planning activities. The concept of using component features to integrate a design system and a manufacturing system has been a major research direction in recent years [1–5].

In general, features are used to give meaning to component attributes, helping in dissecting component geometry into recognisable, meaningful regions, and improving communication between design and manufacture. Feature recognition and feature-based design are two major approaches to using the features. Feature recognition drives form feature information from a geometric database where form features are recognized, extracted and categorized using computer algorithms. Feature recognition has problems associated with the loss of design intent, incompleteness of design and manufacturing information and the complexity and computational load of feature recognition programs [3, 6, 7]. Feature-based design involves defining a component through the use of "building blocks," which represent either *design features* or *machining features*. Since features are process dependent, different applications require different sets of features. The features used in the design stage (called design features) are considered to be functional features, representing design intents, such as *rib*, *slot* and *step* features. The features used in the process planning stage (called machining features) are associated with machining operations, such as *hole*, *blind-hole*, *slot*, and *pocket* features. In addition, feature representations vary from application to application. These facts are the main barriers to correctly representing the machining features as required, and to realise CAD, CAPP and CAM system integration in a feature-based environment.

It is most reasonable for the designer to use design features to construct a component feature model and then convert the model in terms of machining features for use in a process planning system. This process is called *feature conversion* or *feature mapping*. Methods and algorithms for feature conversion have been studied and applied by many researchers [8–10]. Nevertheless, the majority of contemporary feature-based design is still incapable of fully handling one of the most important and perhaps the most difficult aspects in the development of feature-based systems, that is, the definition and representation of feature *interactions*. Feature interaction changes the geometries and attributes of the features previously used, but it does not change them accordingly in the database, which makes the downstream application incapable of obtaining the real component model from the design database. Therefore, preserving feature integrity during the design and machining process phases is one of the key issues in realising the feature-based CAD/CAPP/CAM integration [11–13]. Much research has been conducted into techniques for extracting machining features from a design feature-based system [5, 7, 9, 10, 14–16]. However, such research has mainly focused on the geometric information recognition or conversion. Little has been done in the aspect of non-geometric information of machine features, such as geometric dimensions and tolerances (GD&T). Shah proposed an approach in [17] to create a dimension and tolerance model for use both in design and process planning applications. This GD&T model was used to capture the designer's GD&T scheme on a feature-based design model, validate its completeness, and then transfer the GD&T to machining features extracted automatically by feature recognition methods. The method focused on how to model effectively the dimensions of a part both for use in the design and manufacturing stages. It is therefore a complex and difficult task to construct robustly such a GD&T model suitable for CAD and CAPP systems. For a commercial feature-based system (such as Pro/Engineer), a GD&T model was constructed once a component modelling process was finished. To make use of the CAD model created, the objective is to capture and convert what a process planning system needs in geometry and non-geometric information and to realise CAD/CAPP seamless integration.

To achieve a complete representation for a process-planning model, a component GD&T needs to be complete and correctly represented according to the requirements of the machining features. The work presented in this paper deals with the extraction/conversion of the GD&T from a feature-based model to preserve the integrity and consistency of a machining feature-based process model.

The framework of the proposed feature-based CAD/CAPP integration system through a feature conversion module and a STEP file interface is introduced in Sect. 2. In Sect. 3 we discuss the difficult issues concerning feature interactions that cause feature information alterations and do not change them in the database. Machining features can be generated through the extraction and conversion of the feature information from a design model. In Sect. 4 we introduce the algorithms developed for the extraction and conversion of feature GD&T. A component example is discussed in detail in Sect. 5 to show the results of the geometric dimension conversion. In the final section, the conclusions of the paper are given.

2 Feature-based CAD/CAPP integration system structure

In a feature-based design environment, a component model is constructed from a series of design features, such as protrusions, slots, holes, etc. A downstream application such as process planning has a different view of a CAD model, it uses a series of machining features that correspond to machining operations. To implement the integration of the CAD/CAPP/CAM system, extraction and conversion of the feature geometric and non-geometric information for the representation of machining features is proposed. The framework of the integrated system through a feature conversion approach is shown in Fig. 1. In this approach, feature-based CAD/CAM software – Pro/Engineer – is used to create a component feature model. Through the feature conversion module and a STEP file interface, the corresponding component process-planning model can be created to test and verify the converted machining features, which is discussed in [18].

In the feature conversion module, the form feature mapping is developed to convert the geometric information in the CAD database into the meaningful "shapes" to the machining process, and the GD&T mapping is used to create a complete and correct representation for the converted machining feature attributes.

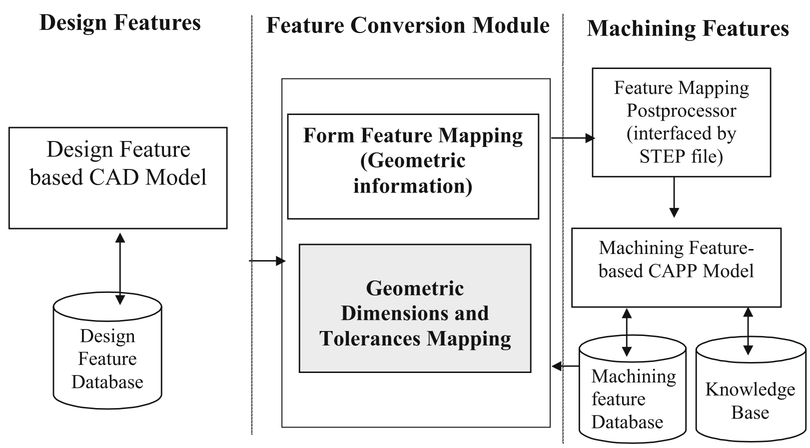


Fig. 1. Framework of the integrated feature-based CAD/CAM system

3 Feature interactions

In general, feature interaction is said to have occurred if it causes any change in feature class or feature attributes of the original features involved. In a feature-based design system, a component model is built up by a series of design features by the way of one feature modifying another. Figure 2 shows a component modelling process that involves the feature interaction of a *blind-slot1* and a *blind-slot2* features. When the interaction occurs, the previously used *blind-slot1* and *blind-slot2* features are changed to *through-slot3* and *blind-slot4* features with the possible alterations in feature dimensions and geometric elements.

Feature interactions have a wide range of effects to a component model because they not only change the predefined feature geometries but also alter their attributes. The problem is that these alternatives are not made available in the component database, which causes difficulties for the downstream process planning application. In Fig. 3, both geometries and dimensions of the previously used *pocket* feature were changed due to the interaction of the *step* feature. The depth dimension of the *pocket* feature in Fig. 3b was changed to h_1, instead of the original dimension h, and the *pocket* feature in Fig. 3c was changed to *blind-slot* feature, and the edge dimension of the newly formed feature has been altered to a_1 instead of the previous dimension a. However, the component database only records the originally used features, and does not contain the modelling results. For the example in Fig. 3 the original *pocket* feature with dimensions h and a is what we can obtain from the database for the two models in Fig. 3b,c.

In order to preserve the feature model integrity it is necessary to develop a robust feature extraction/conversion algorithm to detect inconsistent feature representations and convert them into the correct representations required by the machining features based processing planning system. The detection and conversion algorithms of design features to machining features in the geometric aspect were introduced in [17], here we focus on the extraction/conversion of feature GD&T in order to correctly represent the converted machining features.

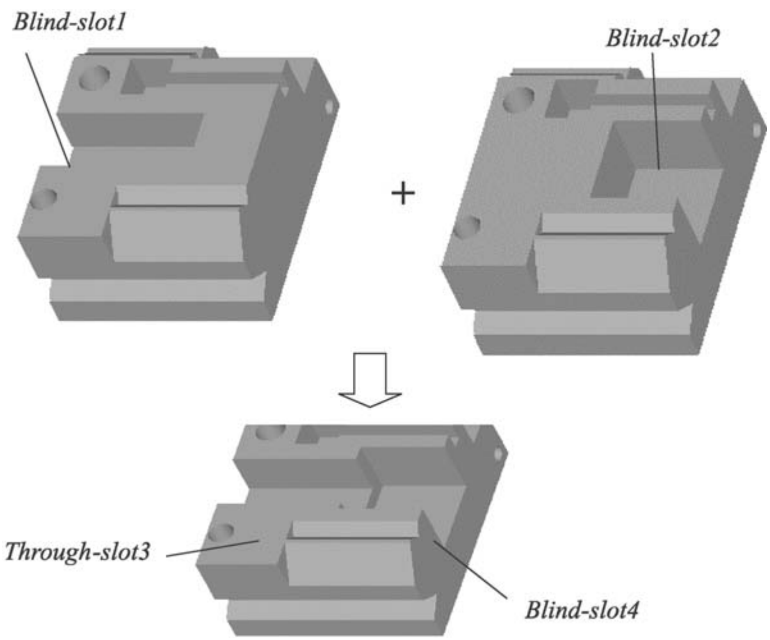
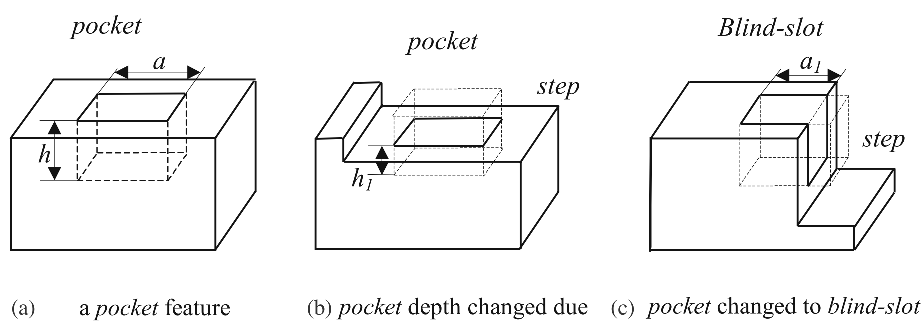


Fig. 2. An example of feature interaction between two *blind-slot* features, producing a *through-slot* feature and a *blind-slot* feature with different attributes or geometry



(a) a *pocket* feature

(b) *pocket* depth changed due to the *step* feature

(c) *pocket* changed to *blind-slot* due to *step* feature

Fig. 3. Inconsistency of feature type and dimensions caused by the interaction of a *pocket* feature and a *step* feature

4 Extraction of feature geometric dimensions and tolerances

4.1 Extraction of feature geometric dimensions

In a feature-based CAD system, the geometric dimensions representation of a component and a feature in the database contain the following items:

1. Dimension value and tolerance.
2. Type of dimension, such as linear dimension, angle dimension and diameter dimension.
3. Boundary elements of the dimension: a face, an axis and an edge. For instance, the boundary elements of the dimension d_4 (shown in Fig. 4) are the face f_1 and the edge e_1, and the boundary elements of the dimension d_5 are the face f_2 and the edge e_2.

Besides items 1 and 2 listed above, a process planning system has a different dimension boundary representation from a CAD model, which needs to describe the dimensions by a meaningful geometric element to machining operations, such as a *face* or an *axis* element, not an *edge* element. Since an edge is formed by two face elements, it is somehow ambiguous for the use of a machining process modelling. For example, the dimension "d" in Fig. 4 is described by the length $|d|$, and the two edge elements of e_1, e_2. As required by a machining process modelling, it should be described by the boundary elements of face f_5 and face f_8.

Because of the different dimension representations between the design features and machining features, it is necessary to distinguish and transform a dimension boundary element *edge* in a CAD model to its corresponding *face* element for process modelling. The algorithm is implemented by the following steps as one of the modules in the GD&T mapping module.

Step 1: Searching each dimension $dim(i)$ on a component, let $i = 1, 2, \ldots, N\_GD$, where N_GD is the total number of geometric dimensions of a component.

Step 2: Let $i = 1$, extracting the value, type, and boundary element identifications id_1, id_2 of the dimension $dim(i)$. *Step 3*: Checking the boundary elements of the dimension $dim(i)$. If one of the boundary elements isn't a real element, then the boundary of the dimension doesn't exist after feature operation. This means the dimension boundary was changed. In this case, an additional algorithm is needed to deal with the interactive features, which will be discussed in the next section. If both boundary elements are real, then the types of the dimension boundary can be determined directly.

Step 4: If one of the dimension boundary elements is an edge e_0, and the other is a face f_0, then one of the two faces f_1 and f_2, which form the edge e_0 must be determined (go to *step 5*) and used as the dimension boundary element of the machining feature. If both boundary elements of the dimension are edges e_1 and e_2, then go to *step 6* to convert both edges into faces as the dimension's two boundary elements. If two boundary elements of the dimension $dim(i)$ are two faces or two axes, or one face and one axis, then their identifications are recorded in an array, then go to *step 7*;

Step 5: Comparing the real value of a dimension with the distance d_{01} and d_{02}, which represents the distance between face f_0 and f_1 and face f_0 and f_2, respectively. If the distance d_{01} is equal to the $Value(dim(i))$, then substitute the boundary identification id_1 of e_0 with the identification of f_1, which means the representation of the edge element has been changed to the representation of the face element. If the distance d_{02} is equal to $Value(dim(i))$, then substitute identification id_1 of edge e_0 with the one of face f_2 and go to *step 7*.

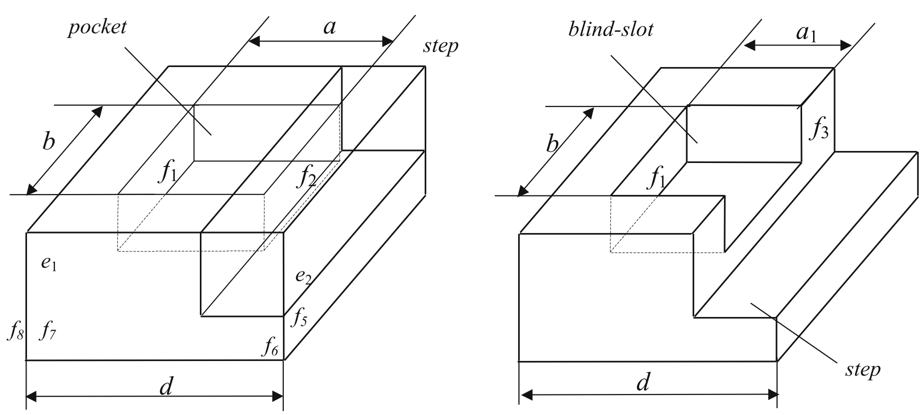
Step 6: Comparing the real value of a dimension with the distances $d_{11}, d_{12}, d_{21}, d_{22}$ (d_{ij} represents the distance between face f_{1i} and face f_{2j}, and i, $j = 1, 2$), there must exist a distance that has the value of $dim(i)$. If d_{21} is equal to the $Value(dim(i))$, then substituting dimension d_{21}'s two boundary identifications with those of face f_{12} and f_{21}, and go to *step 7*;

Step 7: If $i < N\_GD$, then $i = i + 1$, switching to *step 2*. Otherwise, output the results of the dimension boundary transformation.

4.2 Extraction of interactive feature dimension

When feature interaction occurs, the features involved will be altered both in geometry and dimensions, resulting in alterations or deletions of some feature faces and boundary elements of dimensions, but with the original feature geometry and GD&T in the database without alterations. For this reason, the extracted dimension boundary elements may be deleted and not exist on

Fig. 4. Feature interaction and dimension boundary elements



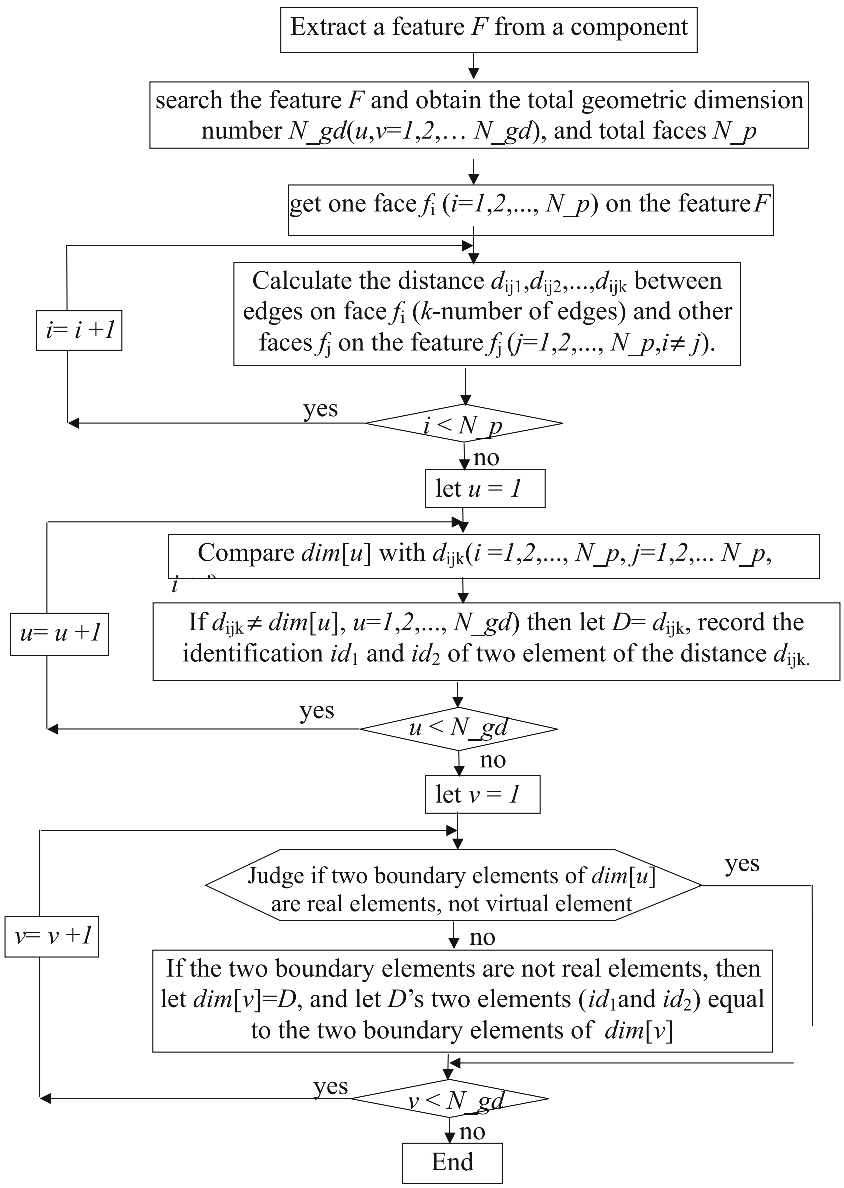
the component model any more. When the dimension a in Fig. 4 is extracted from the database through a Pro/Develop function, its boundary elements are f_1 and f_2. But, because of the interaction of the *step* feature, the face element f_2 doesn't exist on the component any more. That means one of the boundary elements to the dimension a is a virtual element and the dimension a needs to be converted to the newly created dimension a_1 with the real boundary elements of f_1 and f_3. To implement such virtual boundary elements detection and transformation, the algorithm used for extracting and converting the interactive feature dimension boundary was developed and the algorithm flowchart is described in Fig. 5. By means of the algorithm, the inconsistent dimension representation of a feature model in the case of feature interaction can be corrected and converted to the representation of machining features.

4.3 Extraction of geometric tolerances

Geometric tolerances are described by the symbols, tolerances and datums in a frame, which is connected to the feature requiring geometric control. The dimensions positioning surfaces or features requiring geometrical control have basic dimensions and these are identified again by a frame around them. There are 13 geometric tolerances: straightness, flatness, profile (of a line), profile (of a surface), roundness, cylindricity, parallelism, squareness, runout, symmetry, angularity, concentricity and position. The first six geometric tolerances control form and do not require a datum. The remaining seven are relative features and require a datum as they control position.

In some feature-based modelling systems, such as Pro/Engineer, the geometric tolerances of a component model are stored

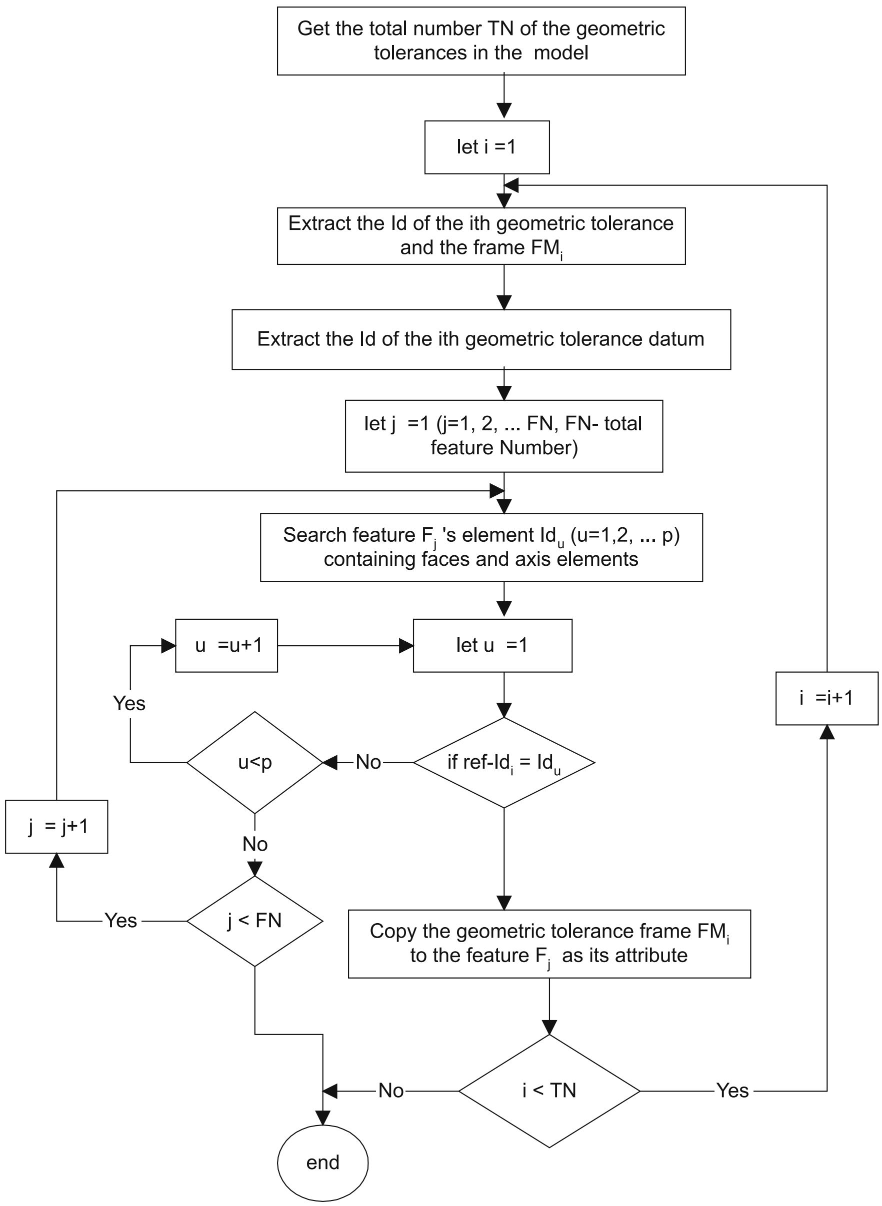
Fig. 5. Flowchart of the dimension boundary conversion algorithm



independently in a data file, which are not connected directly with the related features. In other words, geometric tolerances in a feature model do not exist as its feature attributes. For the CAPP/CAM application, it is important for the machining feature model to get the geometric tolerances from the machining features, so as to create relationships between features and reference datum for the machining processes. Therefore, one of the main tasks in the feature conversion is to extract and convert each geometric tolerance to its related feature as an attribute. The flowchart of the algorithm for the form geometric tolerances (the first six tolerances without datum) is illustrated in Fig. 6. For the positional geometric tolerances with datum, the conversion is divided into a two-stage process. The first stage is to relate the positional geometric tolerances to their corresponding features by a similar algorithm to the form geometric tolerances conversion. The second stage is to convert the datums required to control feature position to real elements that are meaningful to the machining process.

For a feature model created in Pro/Engineer, datums of a geometric tolerance are described by the duplicated feature elements but with different element identification (*Id*). Figure 7 shows such an example of a positional geometric tolerance. The bottom plane F_1 of the component has a parallelism tolerance with a tolerance value of 0.001 and a combined datum A-B. In the internal database, the datum element A is extracted as element A' with the identification (*Id*)#3, which is a virtual element since the real datum element is the face A with the *Id* of #2. From the viewpoint of the process planning ap-

Fig. 6. Flowchart of the form geometric tolerance conversion algorithm



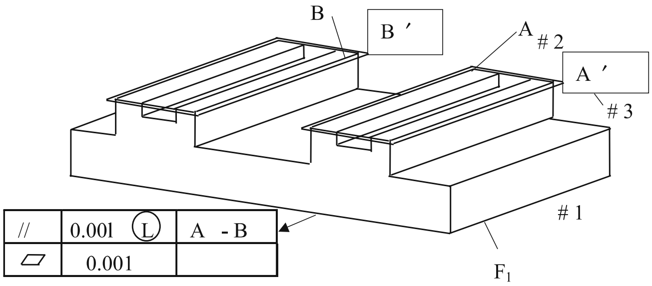
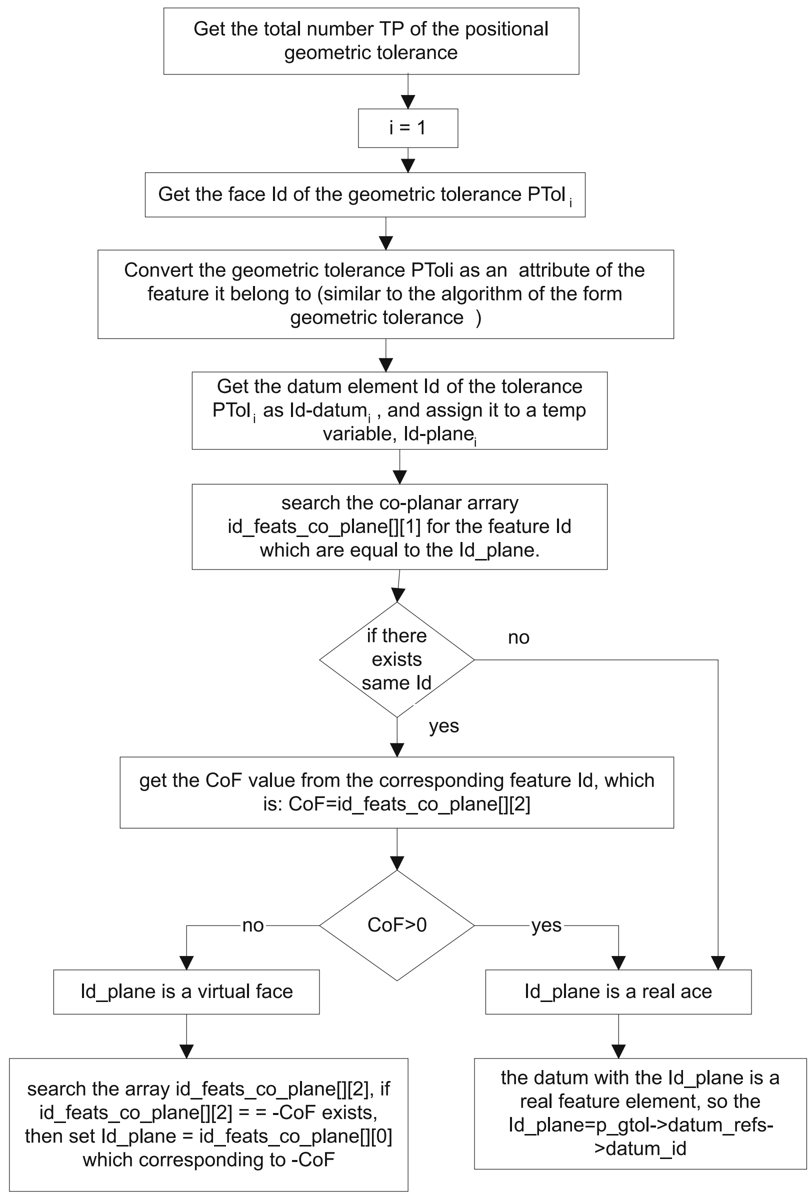


Fig. 7. An example of component positional geometric tolerances

plication, this virtual datum element is not recognisable and cannot be used directly. Therefore, it is necessary to convert virtual datums to real feature elements for the definition of a machining feature model. The developed algorithm is stated below, which is mainly through a co-planar judgement by zero distance between a datum element and one of the feature elements.

Step 1: Initialise an array *id_feats_co_plane*[][*3*], set $i = 1$, $j = 1$
Step 2: Extract the ith datum plane p_datum_i by the function of *prodb_first_datum*() defined in Pro/Develop tool
Step 3: Measure the distance between the p_datum_i and one of the feature elements, p_face_j, $dis_{ij} = prodb\_measure(\ )$; the *prodb_measure*( ) is a function of Pro/Develop
Step 4: If $dis_{ij} = 0$, then the p_datum_i and the p_face_j are co-planar, then go to *step5*; otherwise, set $j = j + 1$, then go to *step 3*

Fig. 8. Flowchart of datum conversion for component positional geometric tolerances



Step 5: Determine if the p_datum_i is a real face (denoted by *RF*) or a virtual face (denoted by *VF*) by following steps:

(a) Find out the *Id* number of the p_datum_i through the *Pro_element_info*( ), denoted by *i_face*, then get the p_datum_i's feature *Id* (that is, to know the p_datum_i belong to which feature) through the *Prodb_get_surface_feature*(), denoted by *i-feat*

(b) Check if the p_datum_i was recorded by the co-planar array *Id_feats_co_plane*[][]. If it wasn't, then assign the array with *i-face*:

$$Id\_feats\_co\_plane[][0] = i\_face,$$
$$Id\_feats\_co\_plane[][1] = i\_feat,$$
$$Id\_feats\_co\_plane[][2] = \text{-}CoF(\text{means a virtual face})$$

similarly, by applying process (a) and (b) to the p_face_j, we obtain:

$$Id\_feats\_co\_plane[][0] = j\_face,$$
$$Id\_feats\_co\_plane[][1] = j\_feat,$$
$$Id\_feats\_co\_plane[][2] = +CoF(meansa\ real\ face)$$

Step 6: If $i >$ datum number N, then end; otherwise, set $i = i + 1$, go back to *step 2*

By means of the above algorithm, the co-planar array stores the datum elements corresponding to the real feature elements. The algorithm flowchart is shown in Fig. 8. Combining with the two-stage conversion algorithms, the positional geometric tolerances can be fully converted.

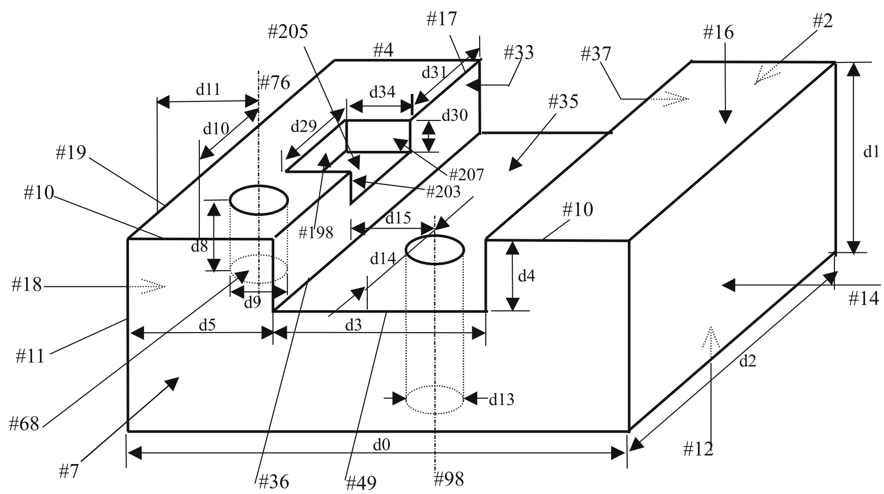


Fig. 9. A component example for geometric dimension conversion

Table 1. Dimension boundary before feature conversion

Feature id	Dim sign	Dimension type			Dimension boundary element1			Dimension boundary element1		
#id_feat	#d	LIN	ANG	DIA/RAD	SRF	AXIS	EDO	SRF	AXIS	EDO
#1	dO	Yes			#14			#18		
base	dl	Yes			#16 #7			#12		
feature	d2	Yes						#2		
#20	d3	Yes			#37			#33		
slot	d4	Yes					#10	#35		
	d5	Yes			#33					#11
	d8	Yes			#16			#68		
#59	d9			Yes	#71					
blind-hole	dIO	Yes					#10		#76	
	dl 1	Yes					#19		#76	
#81	d!3			Yes	#93					
hole	d!4	Yes					#49		#98	
	dlS	Yes					#36		#98	
	d29	Yes			#207			#203		
#190	d30	Yes					#17	#205		
blind-slot	d31	Yes			#207					# 4
	d34	Yes			#198			#33		

Table 2. Dimension boundary after feature conversion

Feature id	Dim sign	Dimension type			Dimension boundary element1			Dimension boundary element1		
#id_feat	#d	LIN	ANG	DIA/RAD	SRF	AXIS	EDO	SRF	AXIS	EDO
#1	dO	Yes			#14			#18		
base	dl	Yes			#16			#12		
feature	d2	Yes			# 7			#2		
#20	d3	Yes			#37			#33		
slot	d4	Yes			#16			#35		
	as	Yes			#33			#18		
	as	Yes			#16			#68		
#59	d9			Yes	#71			#73		
blind-hole	dlO	Yes			# 7				#76	
	dl 1	Yes			#18				#76	
#81	dl J			Yes	#93			#95		
hole	d!4	Yes			# 7				#98	
	dlS	Yes			#33				#98	
	d29	Yes			#207			#203		
#190	d30	Yes			#16			#205		
blind-slot	d31	Yes			#207			# 2		
	d34	Yes			#198			#33		

5 A component example

An example of geometric dimension conversion to a prismatic component is presented in this section. The component model used is shown in Fig. 9. Through the form feature conversion module developed and introduced in [19], the component in the viewpoint of process planning application is a compound of the *base* feature, *slot*, *blind-hole*, *hole* and *blind-slot* features. Feature attributes (non-geometric feature), especially the geometric dimensions of the model, need to be converted so as to describe correctly these machining features. Before the conversion algorithm is applied to the component model, the geometric dimensions are extracted and listed in Table 1. It is found that some boundary elements of the dimensions are edge elements and these need to be converted. After the conversion, all dimensions with edge boundary elements are converted into the dimensions with face boundary elements (shown in Table 2).

6 Conclusions

Feature technology has been an important research topic used for CAD/ACPP/CAM integration. Due to the variety of feature representations from application to application, the main unsolved problem now is to completely explain the design-feature-based component model, in particular regarding feature interactions. Based on the investigation and analysis of the design feature modelling process, it was found that the inconsistency between the final component model and the information available in database is the key issue to be solved for downstream applications. In order to correctly represent the component model by a number of machining features, this paper focuses on the extraction and conversion of the non-geometric features, that is, the geometric dimensions and tolerances (GD&T), which play a key role in the process planning modelling process.

In this paper, GD&T feature conversion algorithms were developed and described in detail. Due to the different representations between the design features and the machining features an algorithm was developed to convert the dimension boundary elements to recognisable elements for process modelling. In the situation of feature interactions, since the feature information in the database does not change accordingly, the algorithm was implemented to detect whether the feature dimensions are consistent with the feature model and to convert those virtual dimensions into the real dimensions that correctly describe the machining features. For geometric tolerance conversion, the main task is to locate each tolerance of the component to the corresponding feature, and convert the virtual datum elements in the design model into recognisable elements for the machining application.

Combined with the geometric information conversion described in [17], the machining features can be generated in terms of the process-planning system requirements. Through the interface of the STEP file, the feature-based CAD/CAPP integration can be implemented [18]. The feature-based Pro/Engineer system was used to build the design model, and the feature conversion algorithm was implemented through the Pro/Develop tool written in C++ on a Sun Sparc 20 workstation.

References

1. Shah JJ, Mantyla M (1995) Parametric and feature-based CAD/CAM: concepts, techniques, and applications. Wiley, New York
2. Liu X (2000) CFACA: component framework for feature-based design and processing planning. Comput-Aided Des 32:397–408

3. Han J, Requicha A (1998) Feature recognition from CAD models. IEEE Comput Graphs Appl 18(2):80–94
4. Gindy N, Yue Y, Zhu CF (1998) Automated feature validation for creating/ editing feature-based component data models. Int J Prod Res 36(9):2479–2495
5. Wang H-F, Zhang Y-L (2002) CAD/CAM integrated system in collaborative development environment. Robot Comput Integr Manuf 18:135–145
6. Han J (1996) 3D geometric reasoning algorithms for feature recognition. Dissertation, The University of Southern California, Los Angeles
7. Yang MH (1997) Recognizing machining feature through partial constraint satisfaction. In: Proceedings of IEEE International Conference on systems, Man, and Cybernetics, pp 12–15
8. Bidarra R, Kraker KJ, Bronsvoort WF (1998) Representation and management if feature information in a cellar model. Comput-Aided Des 30(4):301–313
9. Narayan GA, Nalluri SRPR, Gurumoorthy B (1997) Feature-based geometric reasoning for process planning. Sadhana 22(2):217–240
10. Lee JY, Kim K (1999) Generating alternative interpretation of machining features. Int J Adv Manuf Technol 15:38–48
11. Gao J, Zheng DT, Sun J (1998) Study on mathematical model of feature conversion in domains between design and manufacturing. Chin J Mech Eng 8:4–11
12. Zheng DT (1997) The study on feature mapping for components from design domain to manufacturing domain. Dissertation, Tsinghua University, China
13. Chen YM, Wen CC, Ho CT (2003) Extraction of geometric characteristics for manufacturability assessment. Robot Comput Integr Manuf 19(4):371–385
14. Yan X, Yamazaki K, Liu J (2000) Recognition of machining features and feature topologies from NC programs. Comput-Aided Des 32:605–616
15. Tseng Y, Joshi SB (1994) Recognizing multiple interpresentations of interacting machining features. Comput-Aided Des 26(9):667–688
16. Suh YS (1995) A feature-conversion CAD system for the concurrent engineering environment. Dissertation, Rensselaer Polytechnic Institute, Troy, New York
17. Shah JJ, Yan Y, Zhang BC (1998) Dimension and tolerance modelling and transformations in feature based design and manufacturing. J Intell Manuf 9:475–488
18. Gao J, Zheng DT, Sun J (1997) Research on feature conversion and representation of product process information model. International Conference on Manufacturing Automation, Hong Kong, 28–30 April 1997
19. Gao J, Zheng DT, Gindy N (2004) Extraction of machining features for CAD/CAM integration. Int J Adv Manuf Technol, DOI: 10.1007/s00170-003-1882-9