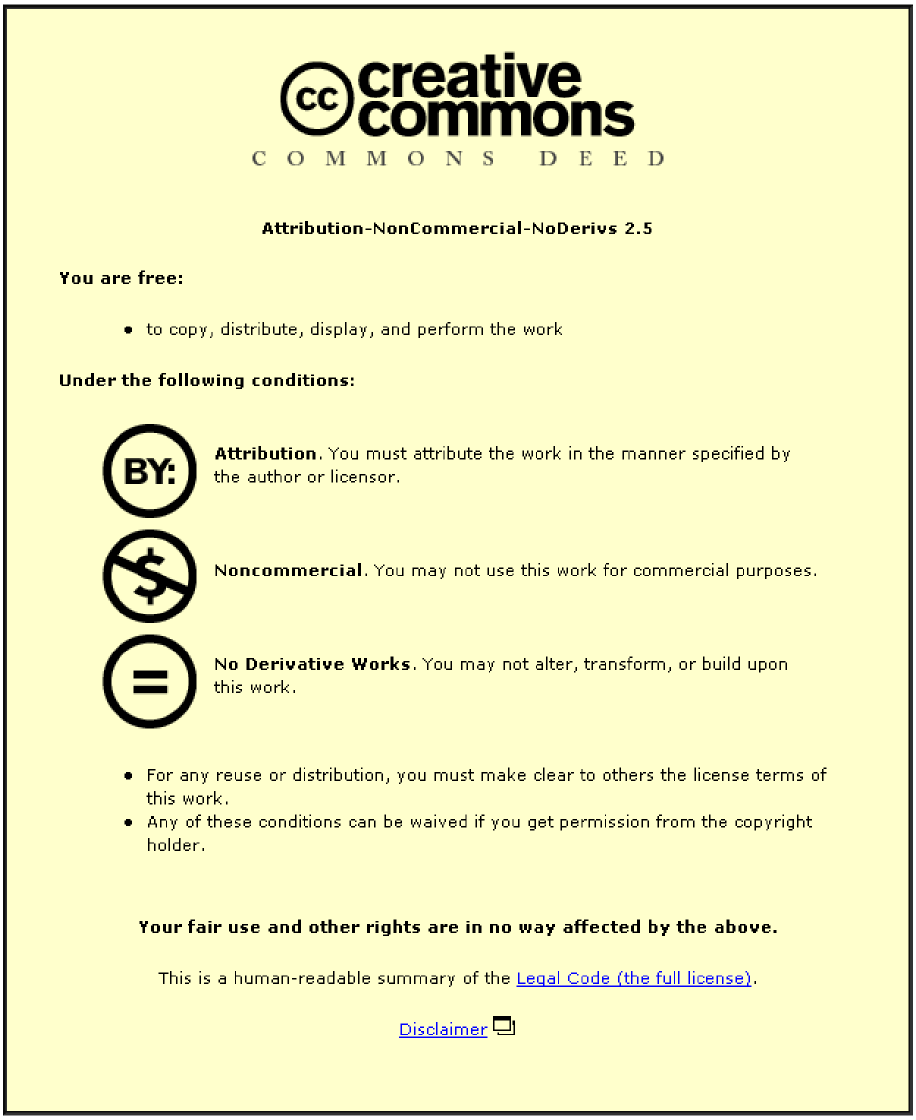

Proceedings of the 15th International Conference on Computer-Aided Production Engineering, CAPE'99
P.G. Maropoulous and J.A. McGeough (eds), University of Durham 19-21 April, pp15-22, 1999

EVALUATION OF A FEATURE MODELLING VALIDATION METHOD

K.Case[1] and M.S.Hounsell[2]
[1] Department of Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, UK.
[2] Computer Science Department, State University of Santa Caterina, UDESC, Joinville, Brazil, 89223-100
K.Case@lboro.ac.uk

ABSTRACT

Geometric modelling techniques for computer-aided design are provided with formal validation methods to ensure that a valid model is made available to applications such as interference checking. A natural and popular extension to geometric modelling is to group geometric entities into *features* that provide some extra meaning for one or more aspects of design or manufacture. These extra meanings are typically loosely formulated, in which case it is not possible to validate the feature-based model to ensure that it provides a correct representation for a downstream activity such as process planning. Earlier research established that validation methods can be based on the capture of designers' intents related to functional, relational and volumetric aspects of component geometry. This paper describes how this feature-based validation method has itself been validated through it's application to a series of test parts which have been either drawn from the literature or created to demonstrate particular aspects. It is shown that the prototype system that has been developed is indeed capable of meaningful feature-based model validation and additionally provides extensive information that is potentially useful to a range of engineering analysis activities.

KEYWORDS

CAD, CAM, CAE, Feature-based Design

INTRODUCTION

There is widespread acceptance that feature-based modelling has much to offer in enhancing computer-aided design systems [1,2]. Improvements are sought through increased capability for design (especially geometry specification and modification) and a better ability to act as the integrating agent for manufacturing applications such as process planning, assembly planning and inspection [3,4]. Typically, feature-modelling methods are developed as a layer on top of an established geometric modelling technique. This modelling technique will usually be some form of solid modelling (most frequently Boundary Representation (BRep)) although surface modelling has been used where appropriate [5].

Geometric modelling techniques are founded upon formal mathematical methods that include validity checking methods. Hence, for example, in the BRep domain the Euler-Poincare Law can be applied to a geometric model to confirm its topological validity in terms of the number of faces, edges, vertices, etc. The significance of this validation in the geometric domain is that it guarantees that valid operations to modify the geometry can be carried out.

In a feature-based representation, geometric entities are formed into groups that can be assigned extra meanings that make the 'features' so formed useful for manipulation in a design context, and which can convey application meaning to manufacturing activities such as process planning. However, in the design context operations such as modelling and editing can corrupt the validity of the feature representation. Feature interactions are a consequence of feature operations and the existence of a number of features in the same model [6,7]. Feature interaction affects not only the solid representation of the part, but also the functional intentions embedded within features. A technique is thus required to assess the integrity of a feature-based model from various perspectives, including the functional intentional one, and this technique must take into account the problems brought about by feature interactions and operations [8]. The understanding, reasoning and resolution of invalid feature-based models requires an understanding of the feature interaction phenomena, as well as the characterisation of these functional intentions. A system capable of such assessment is called a feature-based representation validation system. The research reported here had the objectives of studying feature interaction phenomena and designer's intents as a medium to achieve a feature-based representation validation system.

It was found that feature interaction classifications available in the literature are strongly oriented towards the feature recognition approach and are mainly inappropriate to design-by-features systems. A feature interaction classification and identification mechanism has been proposed, together with a taxonomy of designer's intents that makes explicit many of the expected behaviours of features [6]. The binding process that relates feature interactions to intents allows the validity assessment of the representation and also the identification of operations that contribute to the revalidation of the representation. This binding process leads to a reasoning mechanism that performs feature validation and is driven by designer's intents, and is known as **FRIEND** (**F**eature-based validation **R**easoning for **I**ntent-driven **EN**gineering **D**esign).

This paper briefly introduces the methodologies that support FRIEND**,** but concentrates on the evaluation of the approach. This 'validation of the validation' was carried out by investigating the performance of the model when presented with a range of test parts, some of which have been established by other researchers in the UK, USA and Europe and some of which have been designed specifically to test aspects of FRIEND.

FEATURE-BASED MODELLING

Current Computer Aided Design systems are based on Geometric Solid Modelling (GSM), but future technology is likely to be based on Feature-based Modelling [9] which offers the possibility of integration with other engineering applications such as manufacturing and process planning. Geometric solid modelling is well-established, popular and powerful as the method is founded upon sound geometric knowledge that permits *Geometric Validation*. i.e. at any time the validity of a geometric model within the specified domain can be determined by a set of functional or procedural evaluations, and thus the model can be guarantied suitable for a geometric application such as interference checking or rendering. For example application of the Euler-Poincare Law can identify topological inconsistencies as shown in (Fig. 1).

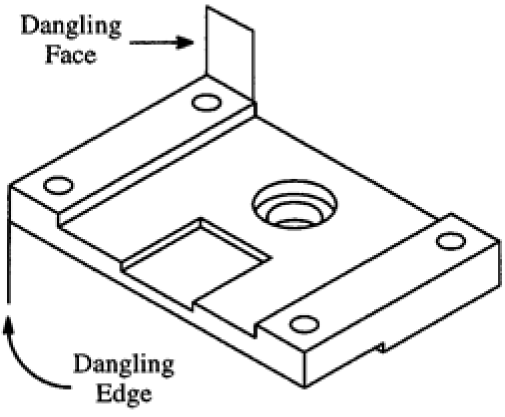

Figure 1. Topological problems detectable using Euler-Poincare Law

Validation with this degree of rigour is not available within feature based modelling systems, as features add a layer of complex semantics which are difficult to measure and subjective to implement (Fig. 2). Feature-based representation validation is nevertheless very important because it is the process responsible for guaranteeing the delivery of a valid (verified, useful and misrepresentation free) representation to downstream applications such as manufacturing planning.

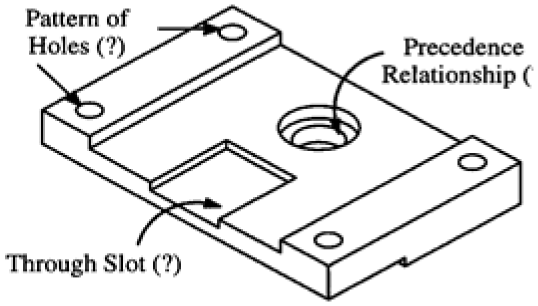

Figure 2. No formal rules for feature semantics

Designer's intents represent information that should be verified and maintained throughout the detailed design process and could be used to drive the decision-making for downstream applications. Because they are considered intrinsic to features, they are sometimes omitted from the formal and explicit description of a design. Nevertheless, Feature Based Designer's Intents (FbDI's) act as a suitable medium for the validation of feature-based representations.

DESIGNER'S INTENTS

It has been acknowledged that □the information that constitutes intent, and how to capture and use intent are all research issues to be explored" [10]. This has been discussed in more detail in [11] where Feature-based Designer's Intents are defined as representing a variety of concerns that help decide on a specific feature attribute or configuration. They are factual peculiarities of the geometric design that are intrinsic to features or to the use of features in the design and have engineering-related purposes. FbDI's are properties that are expected to arise in the model because of the use of a feature in a specific location or because of the interactions between a feature and surrounding features in the model.

The exhaustive enumeration of all possible sets of FbDI's is a very cumbersome approach even in a limited domain, and so the objective was to explicitly categorise FbDI's in such a way that this extra information could be effectively and consciously instantiated into a model. In this way the capturing, verifying and maintaining of FbDI's could be performed by, and even automatically discovered by, a design-by-features system. A taxonomy of such intents has been defined and detailed in [11]. The following sections briefly outline some of the more important aspects.

FEATURE BASED DESIGNER'S INTENTS

Feature-based Designer's Intents (FBDI'S) are characterised as Theoretical, Relational or Morphological. Each of these types has a set of objectives and a tangible set of properties to enable their implementation within the geometric realm. The generic types specify general engineering concepts or behaviours while the specific FbDI's are computable relationships between features themselves or elements of the feature-based model such as feature faces (and their attributes) or feature parameters.

Theoretical Functional FBDI's

Features may be have a *functional* aspect which is defined as the behaviour of an object, an operation of energy, material, information or signal that tells what the design does" [12] and, include not only in-use purpose, but also manufacturing and life-cycle considerations [10]. The relationships between *form* and *function* cannot be formalised because of many difficulties [1] including the abstract nature and understanding of the *function* concept, the fact that functionality can be a composite result of many interacting sub-functions, and that a given *function* could be performed by several forms and one form might be used to perform a number of different *functions*.

This *function* concept has been implemented as physics-based or engineering-based laws, rules or formulae depending on the underlying theory such as heat propagation, torque or force transference or, stress analysis. Thus, they are called *theoretical functional* FbDI's.

Theoretical functional FbDI's are intents that make specific shape aspects appear on the part's surface, control the part's overall outlook and, are driven by a close relationship between a feature's theoretical functional behaviour and its form. This is possible by manipulating and controlling the hierarchy or dependency of parameters that establish dimensions, profiles (e.g.: quadric, circular, spherical), parameterised local operations (blending, chamfering, trimming), and so on. Theoretical functional FbDI's can be achieved via a parametric constraint-based approach and therefore are not discussed any further.

Relational Functional FBDI'S

Relational functional FbDI's (**RDI's**) express relationships between entities and are thus application dependent. Some important RDI's are geometrical facts that have a functional significance for an application. For instance, a *nested at the bottom"* RDI is a geometry-based and provable fact that could be used by a computer aided process planning system to establish machining precedences.

RDI's describe physical and/or spatial relationships between features and are categorised as being κapplication-dependent but primarily geometry-dependent, in which case they are called Geometric RDI's (**GDI's**), and geometry-dependent but primarily application-oriented, called Application Oriented RDI's (**AOI's**).

The importance of GDI's has been recognised by many systems that incorporate spatial reasoning in various ways [7,13,14]. GDI's are geometrical facts and intentional relationships between entities of a feature-based modelling system but they alone do not suffice for an application. Thus, a hierarchical GDI may be needed to define machining precedence but geometrical reasonings such as supporting walls" and tool accessibility" must also be considered.

Positional GDI's include concentric, opposite, planar, coplanar and concentric intents between features. Orientational GDI's include parallel, perpendicular, angularity, against, co-linearity and common External Access Direction intents. Hierarchical GDI's include nested at the bottom and nested at the side. Structural GDI's include patterns with linear, circular, planar or spatial distribution; radial, axial or mirror-like symmetry and co-radius intention.

Application-Oriented RDI's (**AOI's**) arise from the intentions of manufacturing engineers, process planners, etc becoming a part of the design information. Many of these intents are concerns to be fulfilled that guarantee the physical realisation of the design constrained by pragmatic and technological requirements such as cost, quality, time, accessibility and feasibility.

Application-oriented FbDI's include: same or different set-up intents; parent-child and precedence intentional relationships; T-slot, cross feature, entry feature, counter-bore, counter-sink and cut-out compound intentions between features and thin-wall proximity intentions.

Morphological Functional FBDI'S

The extra descriptive factors that are added to the topological and geometrical aspects of the geometric solid model are frequently used to better specify the elements of a feature family. Thus Neilson and Dixon [13] describe how a *cylindrical boss* family of features could be specialised into a *disk for a* certain height-to-diameter ratio range or into a *rod* with an alternative ratio.

Hence features clearly have a *morphological function*, which in the geometric domain have been implemented as Volumetric Designer's Intents (**VDI's**) to define expected geometric behaviour FbDI's for features.

Four Volumetric Designer's Intents (**VDI's**) are of particular interest. The *labelling* VDI identifies the relationships between all the feature's faces and their attributes. The feature's additive or subtractive *nature* implies that a change in the feature-based representation must result in a change in the volume and surface of the component being modelled. This requirement and the ability of a feature to change the existing model is called the *changeability* VDI. A feature must have adequate parameters to exactly fit and define the intended form (in the same way as an edge is limited by its two exact ends, called vertices) thus, the feature must fit within the limits of where it is intended to be placed. This ability to fit is called the *fittability* VDI. Furthermore, interesting and difficult situations arise when redundant intents are found. Features that have overlapping volumes usually present a *redundant* VDI.

VALIDATING THE VALIDATION

This section presents some feature-based part models as test cases for FRIEND. Some of these models have been used in the literature as test cases for feature-based modelling system implementations. It aims to show that the prototype system is able to represent and reason with components which have been modelled by and used to test the capabilities of other feature-based modellers.

The parts shown are adaptations of the original parts because dimensions are frequently not specified for the parts or the feature taxonomy used to describe the part could be different from that used by FRIEND**.** Some invalid situations have been deliberately introduced in the part definitions to observe the response from FRIEND, some features implemented in other systems are not available in the prototype system and some geometric configurations have been simplified. Figure 3 illustrates a typical feature-based component model with many of the feature types implemented in FRIEND.

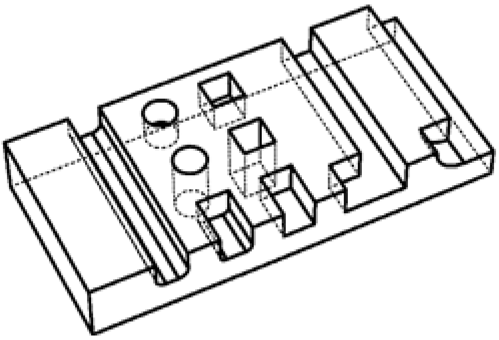

Figure 3. Typical component modelled in FRIEND

The model is defined via a part description file which is a simple way of describing all features that represent a model, and avoids the task of redefining the model feature by feature every time an analysis is to be performed. The part file defines the total number of features in the part, their assigned names, their volume type, a radius to be used if the feature is cylindrical, an orientation and a bounding box.

The stock material is considered to be a rectangular satellite feature of positive nature which contains the remaining negative features. Blind holes are classified as pocket features with round or rectangular profiles, as in earlier research [15,16].

Labelling

Figure 4 shows a part described by Martino and Giannini [17] where the labelling problem is highlighted as the addition of a feature into the model could change the label (type) of all existing features. Figure 4(a) represents the original part containing a pocket on the bottom face (elsewhere called a non-through or blind hole) and a square (through) hole. The addition of a step feature renders the existing hole and pocket features invalid, and Figure 4(b) represents the final part comprised of the newly defined step, a through slot (originating from the through hole) and a new hole (originating from the pocket).

In this example, the major differences detected between the valid and invalid representations are:

- the through hole feature, initially labelled as a hole, is detected as invalid and is split into two new through holes, one of which is redundant to the volume of the step and is therefore made obsolete and receives the intentional status. The remaining through hole actually affects the stock and thus receives the validated through slot label and an active/valid status.
- Similarly, the blind hole feature, labelled correctly at the beginning as a pocket, is split into invalid and valid parts. The valid part is labelled as a hole feature and receives the active status.
- The step feature is found to be correctly labelled as a step feature but its orientation is changed to a standard form.

Both the obsolete through hole and the blind hole become intentional features because their volumetric intention can reappear if the step feature is deleted.

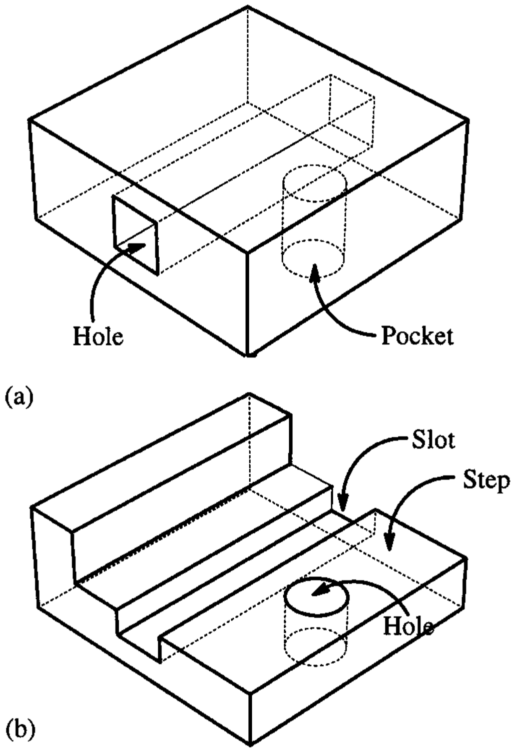

Figure 4. Martino and Giannini's Part.

Valid Part Description

Figure 5(a) shows an example part consisting of a set of feature volumes before validation, and Figure 5(b) shows the same part after validation reasoning.

The output produced by validation lists all features in the model and, where appropriate, also includes invalid/inactive and intentional features in addition to the valid/active ones. The output gives the name of the feature, the label, the volume type, the status (valid, invalid or intentional), the validated envelope (bounding box), orientation and location.

In this example the independent adjacent notch and slot features are merged to compose a single feature that is labelled as a notch. The solid cylinder used to define the 'hole' in figure (a) has been defined such that it extends beyond the stock material. Hence it is split in two with one part made inactive, and the other correctly labelled as a pocket (as it is not a through hole). Both these reasonings are related to the fittability VDI where the features had parameters too small or too large, respectively. The feature originally incorrectly defined as a slot has been corrected to a through slot feature, this being a typical example of the result of reasoning related only to labelling.

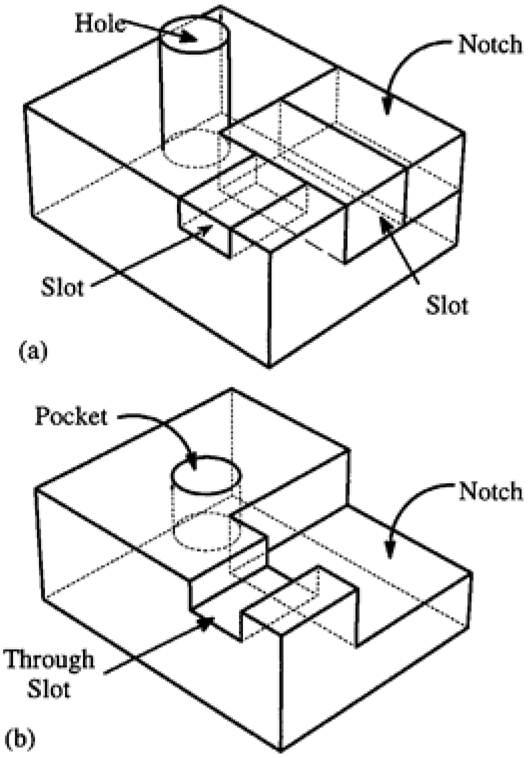

Figure 5. A part before (a) and after (b) validation.

Morphological Reasoning Test

Figure 6 illustrates a part where a complete conceptual morphological validation process is carried out. Figure 6(a) shows the part with the original volumes of the features while Figure 6(b) shows the output after the application of the Boolean operations associated with the construction of the feature-based model.

The validation results in the part of the radiused slot outside the stock-material and the part overlapping the rectangular slot feature both being discarded.

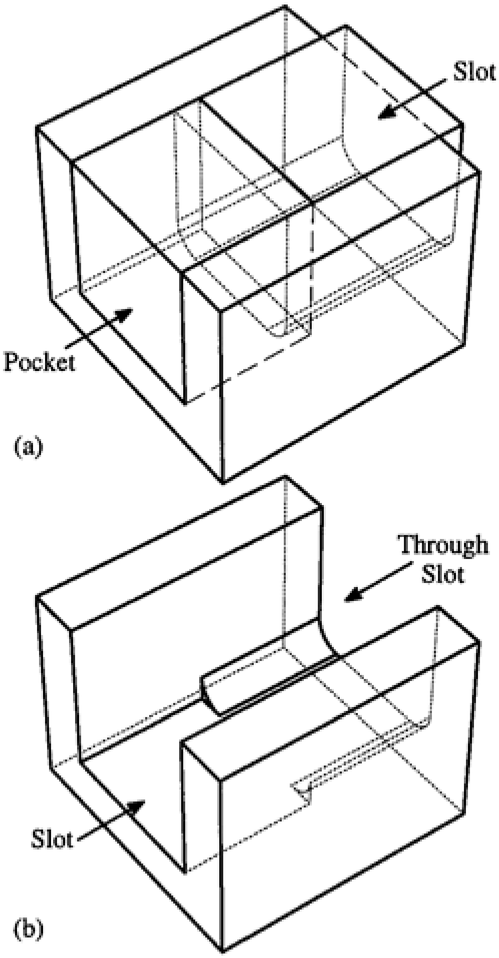

Figure 6. Morphological Validation Reasoning.

The incorrectly labelled pocket is re-labelled as a slot. The two resulting slots are not merged because the features have different radii. Nevertheless, the original slot (with the floor radius) is redefined as a through slot. Part of the original slot has a redundant VDI with the original pocket, and the feature resulting from the split revalidation operation is assigned the intentional status. This means that if the original slot is deleted from the model the overlapping part of the original pocket can again become active.

Thin Wall Test Cases

Figure 7 shows an example part produced to demonstrate the identification of proximity/thin wall conditions - an example of an Application Oriented Intent (AOI). Thin-wall reasoning can be built upon feature interaction cases where features are adjoint to (touching) other features or the stock material or disjoint (separated by a 'small' distance) from other features or the stock material.

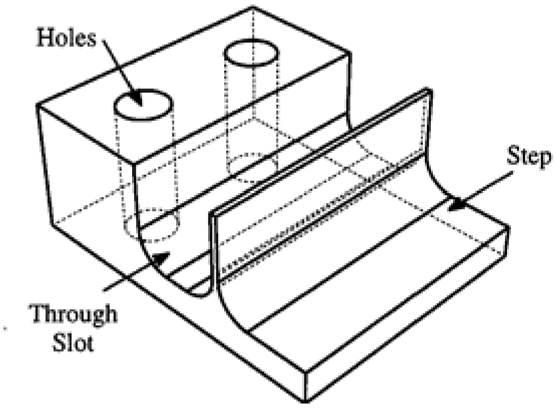

Figure 7. Thin-Wall (disjoint) Interaction.

The application of rules concerned with proximity testing of volumetric (VI) and boundary (BI) interactions obtained from the model are used to determine the AOI's. In the example shown, potential thin-walls were identified between the step and the through slot features, between each of the holes and the through slot.

Process Planning

Chang [18] studied expert process planning for manufacturing, and used a test part to discuss the problems and reasonings related to the generation of automatic process plans (Fig. 8).

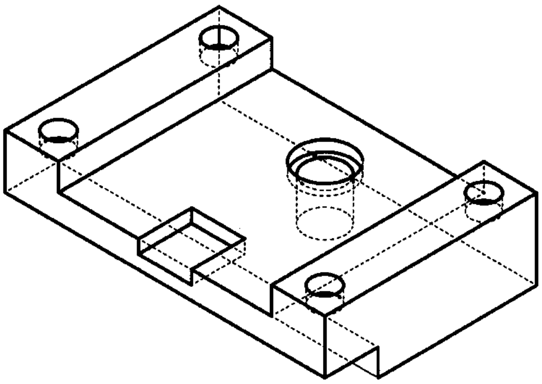

Figure 8. Chang's Process Planning Part.

One strategy adopted by Chang was to identify clusters of features that share the same tool and/or tool access direction. This information is used to reason about set-up planning. A hierarchical graph that identifies various types of precedence (such as structural precedence due to process geometry constraints and loose precedence due to good manufacturing practice) is considered for reasoning about precedence planning.

Although generating plans is not FRIEND's major concern, it gathers valuable information during the design process that can be readily used for similar clustering and hierarchical reasoning. Many GDI's and AOI's are obtained while validating the part and represent potentially valuable information. For example, the existence of a compound AOI representing the counterbore intent is detected, as are the common diameters and access directions of the four holes comprising a rectangular pattern.

Mantyla et al. [19] were also concerned with process planning problems, and considered parts such as that shown in Figure 9.

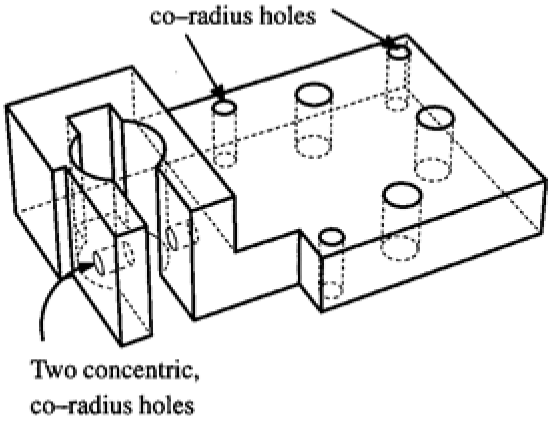

Figure 9. Mantyla et al's Part.

Relational Functional Intents (RDI's) can be obtained from this part and may be used for process planning. In particular, co-radius, parallel and concentric Geometric RDI's help to identify groups of hole features to be machined in the same set-up, perhaps with the same process and tool.

The indicated concentricity and co-radius GDI's were obtained through guided enrichment. i.e. the two holes were originally part of a single, longer hole that was split by a slot and this knowledge *guides* the system into assigning the intent to the model. The co-radius GDI indicated for two of the holes were obtained from blind enrichment rules. i.e. an exhaustive search identified the possible intention that was left to the user to confirm.

A Lost Intention?

Perng and Chang [20] studied the problems associated with editing a feature-based model, and used the part shown in Figure 10 as an example. The conceptual validation problem arose where the enlargement of the top part of the T-slot results in the disappearance of the Hole feature. The question of how to handle this situation is fundamental to the validation process.

This problem is dealt with in the following way: Every time a feature volume becomes contained within another feature volume, the former is made obsolete and receives an intentional status. In the example shown this happens to the Hole at two levels. Firsly, the long hole is split into three by the Through Slot. Two of of these holes are shown as cylindrical holes in Figure 10(a). The remaining part of the original hole is obsolete as it is contained within the volume of the Through Slot - however, it is an intentional part of the modelling and is marked as such. Increasing the dimensions of the T-Slot (Fig. 10(b)) results in both remaining hole sections being made obsolete. The intentional status means that if the T-Slot is subsequently removed or reduced in size the hole feature can reappear in the model.

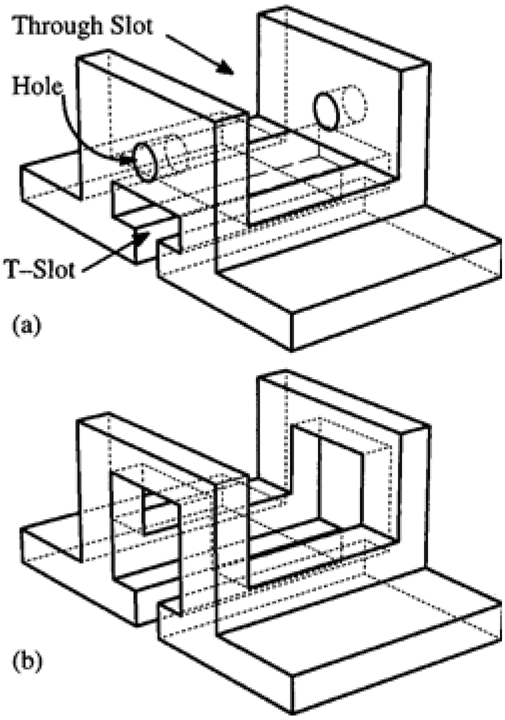

Figure 10. The Vanished Hole Feature.

Redesign

Das et al [21] were concerned with set-up planning and automated redesign, and Figure 11 represents a typical reported example component. The slotted cross-shaped feature-based part was built and validated by FRIEND which produces a list of all valid features resulting from the validation reasoning.

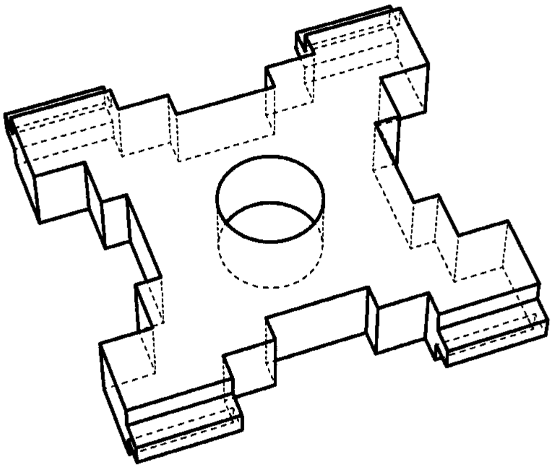

Figure 11. A Slotted Cross-Shaped Part.

Note that all features have a quadrangular volume type, except the central hole feature. A large number of nesting and common access direction FBDI's are identified and it is possible to envisage these being used in conjunction with decision-making software to suggest alternative redesigns related to function or process planning.

Edinburgh Composite Component

Mill et al [4] have defined the Edinburgh Composite Component (Fig. 12) as a test part for investigating process planning conflict situations. Again, although FRIEND does not generate a process plan, it obtains a plethora of information that can help in analysing and solving some of the planning difficulties. This valuable extra information comes in the form of VDI's (e.g. the splitting of Hole1 into two parts by Hole2), GDI's (e.g. the parallelism between the through slot and the step) and AOI's (e.g. the common access direction for the component features of the nested slots).

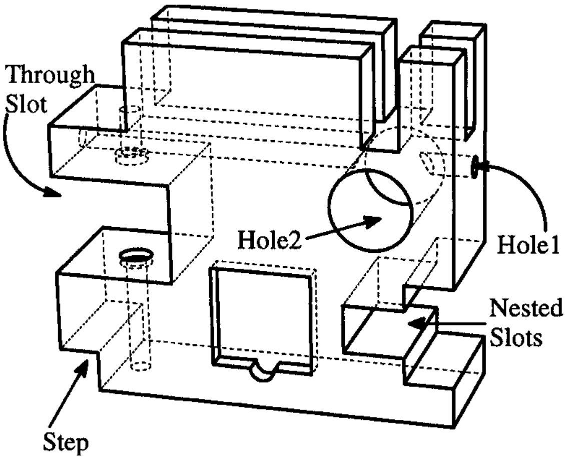

Figure 12. The Edinburgh Composite Component.

The major concern of FRIEND is to make these intentions explicit to the designer and if appropriate assign them to the model. No strategy for planning the processing or production of the part is suggested.

DISCUSSION

It was found that a comparison between the functionalities of FRIEND and other systems is not straightforward because most of the systems studied perform some variety of geometric reasoning on the *complete* model (and therefore, as a post-processing procedure) while FRIEND accumulates knowledge *throughout* the design process by analysing the part model every time an operation is performed.

Furthermore, some of the test cases presented were obtained from literature more concerned with feature-based process planning problems (of the complete part model) while the major concern in FRIEND is in the correctness of the representation and the FbDI's that can be gathered from and during the design process.

In carrying out this validation FRIEND is capable of producing much more information than most feature-based modellers and this information can be used for various engineering-related activities, not only process planning. Some parts of this reasoning are direct derivations from the feature-based designer's intents (FbDI's) identified by FRIEND while others would require extra technological information to reach a conclusion.

CONCLUSIONS

This paper has presented test parts that were adapted from the literature. FRIEND could model the parts and correct some of the definition mistakes (introduced deliberately), and although the production of process plans was not the objective, it was able to produce a plethora of information that could help such downstream applications.

Some difficulties were found in comparing the functionality of FRIEND with other work because FRIEND gathers intentions during the ongoing feature-based modelling task while most of the other systems perform a post-processing analysis on the final and static feature-based model.

It can be inferred that the way the model is built can affect the resulting amount and type of information produced by FRIEND and this is consistent with the non-commutability characteristics of the Boolean operations (which are implied by feature-based models).

REFERENCES

1. O.W. Salomons, F.J.A.M. van Houten and H.J.J. Kals, Review of research in feature-based design, *Journal of Manufacturing Systems*, 12(2), 113-132, 1993.
2. J.J. Shah and M. Mantyla, *Parametric and feature-based CAD/CAM: concepts, techniques and applications*, John Wiley and Sons, USA, 1995.
3. N.N.Z. Gindy, X. Huang, and T.M. Ratchev, Feature-based component model for computer-aided process planning, I*nternational Journal of Computer Integrated Manufacturing*, **6**(1-2), 20-26, 1993.
4. F.G. Mill, J. C. Salmon and A. G. Pedley, Representation problems in feature-based approaches to design and process planning, *International Journal of Computer Integrated Manufacturing*, **6**(1-2), 27-33, 1993.
5. S.R. Mitchell, *A Feature-based Approach to the Computer Aided Design of Sculptured Products*, Ph.D. Thesis, Loughborough University, UK, 1996.
6. M.S. Hounsell and K. Case, Morphological and Volumetrical Feature-based Designer's Intents, in: *Advances in Manufacturing Technology XI, (ed. D K Harrison), The Proceedings of the 13th National Conference on Manufacturing Research*, Glasgow, Scotland, **1**, 64-68, 1997.
7. R.E. da Silva, K. L. Wood and J. J. Bearman, Representing and manipulating interacting interfeature relationships in engineering design for manufacture, *(ASME) Advances in Design Automation*, DE-32-1, 1-8, 1990.
8. K.F. Zhang and H.A. Elmaraghy, Validity check for a functional-oriented modeler, *(ASME) Advances in Design Automation*, DE-65-2, 293-300, 1993.
9. M. Mantyla, D. Nau, and J. Shah, Challenges in Feature-based Manufacturing Research, *Communications of the ACM*, 39(2), 77-85, 1996
10. J.R. Dixon, E. C. Libardi Jr, and E. H. Nielsen, Unresolved Research Issues in Development of Design-with-Features Systems, in: *IFIP WG 5.2/ NSF Working Conference on Geometric Modelling*, ResselaerVille, USA, Elsevier Science, 1, 183-196, 1990.
11. M.S. Hounsell and K. Case, Feature-based designer's intents, in: *IMC-15 Manufacturing Strategies for Europe, Proceedings of the Fifteenth Conference of the Irish Manufacturing Committee* (eds. C Hepburn and D M J Harris), University of Ulster at Jordanstown, UK, 193-202, 1998.
12. T. Tomiyama, Y. Umeda and H. Yoshikawa, A CAD for functional design, *Annals of the CIRP*, 42/1, 143-146, 1993.
13. E.H. Nielsen, J.R. Dixon, and E.E. Zinsmeister, Capturing and using designer intent in a design-with-features system, in: *Proceedings of DTM'91: 3rd International Conference on Design Theory and Methodology (ASME-DE)*, Miami, Florida, USA, 31, 95-102, 1991.
14. J. Vancza and A. Markus, Features and the principle of locality in process planning, *International Journal of Computer Integrated Manufacturing*, **6**(1-2), 126-136, 1993.
15. N.N.Z. Gindy, A hierarchical structure for form features, *International Journal of Production Research*, 27(12), 2089-2103, 1989.
16. J.X. Gao and K. Case, Information mapping between feature-based design and and integrated process planning system, in: *Proceedings of the 28th MATADOR Conference*, Manchester, UK, 1, 551-558, 1993.
17. T.D. Martino and F. Giannini, The role of feature recognition in future CAD systems, in: *Proceedings of the IFIP International Conference on Feature Modelling and Recognition in Advanced CAD/CAM Systems*, Valenciennes, France, 1, 343-355, 1994.
18. T.-C. Chang, *Expert process planning for manufacturing*, Addison-Wesley, Reading, USA, 1990.
19. M. Mantyla, J. Opas and J. Puhakka, Generative process planning of prismatic parts by feature relaxation, in: *Proceedings of (ASME-DE) 15th Design Automation Conference*, Montreal, Quebec, Canada, 19-1, 49-60, 1989.
20. D.-B. Perng and C.-F. Chang, Resolving feature interactions in 3D part editing, *Computer Aided Design*, 29(10), 687-699, 1997.
21. D. Das, S.K. Gupta and D.S. Nau, Generating redesign suggestions to reduce setup cost: a step towards automated redesign, *Computer Aided Design*, 28(10), 763-782, 1996.