



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<https://creativecommons.org/licenses/by-nc-nd/2.5/>

Feature-based interaction: an identification and classification methodology

M da Silva Hounsell¹ and K Case²

¹Computer Science Department, State University of Santa Catarina, UDESC, Joinville, Brazil

²Department of Manufacturing Engineering, Loughborough University, Leicestershire, UK

Abstract: Features are an established means of adding non-geometric information and extra geometric semantics to conventional computer aided design (CAD) systems. For some time it has been realized that, although feature-based modelling is necessary for the next generation of integrated design and manufacturing systems, the inherent feature interactions pose a difficulty in representing and manipulating geometric designs. This paper presents a structured geometric spatial feature interaction identification method based on a broad multilevel classification. Feature interaction definitions and classifications have been surveyed and it is evident that, although many feature interaction classifications have been proposed, there is a lack of a general framework. The classification presented here encompasses existing feature *interference* cases found in the literature and defines a singular framework that leads to a general classification structure. The framework is presented and applied at three different levels and each interaction case is defined by feature parameters rather than just geometric entities. The restrictions often found in other research concerning contact/non-contact and concave/convex situations are avoided. The resulting classification is easy to understand and implement because it uses simple rules based on commonly available Boolean operators. Finally, an example component is presented and the advantages, uses and applications of the classification scheme are discussed.

Keywords: CAD/CAM (computer aided design/manufacture), feature-based modelling, design-by- features systems, feature interactions

1 INTRODUCTION

Feature-based modelling (FBM) systems enhance traditional computer aided design (CAD) environments, making them more meaningful and easier to use, and facilitate their integration with other computer aided systems within the manufacturing context because they subsume extra non-geometric semantics. FBM systems use entities such as *holes*, *slots* and *bosses* that are close to the designer's own vocabulary (Fig. 1). Features are potentially a means of incorporating knowledge of the form, behaviour, function and related manufacturing processes into a single representation.

One way of implementing feature-based CAD systems is the design-by-features (DbF) approach which offers the designer a library of features that can be used to represent components. DbF systems are distinct from the alternative feature recognition (FeR) approach, where features are 'discovered' after a session using traditional CAD systems. Current thinking is that these elements of traditional CAD systems, DbF and FeR can be combined into a DbF-like

system producing a useful, powerful and flexible system [1, 2].

The use of more meaningful entities, such as features, suggests the need for more meaningful ways to establish the interactions between these entities. Feature interactions occur when it is not possible to consider features in isolation within the model because some influence is exerted among them. These interactions must be dealt with in terms of the meaning of the interaction and its importance (especially for manufacturing engineering purposes).

A lack of attention to formalizing the concept and classifying feature interactions can be seen in currently available DbF systems, even though this is a well-known, important and active issue of research. Interactions between features are at the heart of any feature-based modelling environment because they are directly and inevitably produced while manipulating the model [3]. Additionally, *intended interactions* are common practice in engineering

and can be found, for example, in tolerances, in assembly relationships or in the assignment of patterns of features [4].

Furthermore, feature interactions are the cause of some of the most serious problems in the development of generative computer aided process planning (CAPP) systems [5] and are important for determining process sequences and sometimes the processes themselves [2, 6–8]. For these reasons, feature interactions urgently need to be further investigated, precisely defined and established, with comprehensive coverage in a research context before being implemented in practical design systems.

A formal and comprehensive feature interaction identification and classification methodology is described in this paper. Firstly, the use of a common sense definition of feature interaction for prismatic DbF systems is established and differentiated from alternative definitions found in the literature. Existing feature interaction classifications are discussed and incorporated into a single, simple framework that is independent of its interpretation and application. The classification framework also constitutes a geometric analysis methodology and interaction identification structure based on Boolean operators. It is then shown that the same framework can be consistently applied to produce a hierarchical classification for feature interaction cases at each of the volume, boundary and face levels. Finally, the advantages, uses and implementation of the framework are discussed.

2 TERMINOLOGY

A distinction has been made between *feature relations* and *feature interactions* [2]. It has been argued that feature relations are concerned with non-overlapping situations, whereas feature interactions involve a change in the internal geometric representation of features (both volumetric and surface). However, both relation and interaction classifications include *touching* and/or *adjacency* cases and are based on geometric reasoning. This indicates that they could perhaps be based on a single unified framework.

For the purposes of this paper, a *feature interaction* is defined as a mutual action or influence existing between features that has significance to engineering. This definition emphasizes that interactions occur when features cannot be considered as isolated entities within the model and that this may happen with volumetric overlapping features as well as with non-overlapping and even non-contacting features.

Nevertheless, some confusion exists because the term *interference* is also used in the literature to mean interaction, although it is frequently associated merely with the volumetric overlapping cases of feature interaction. Interference has sometimes been used to refer to interactions as a whole because it represents one (of many) examples of a very important feature interaction with a direct impact on manufacturing decisions.

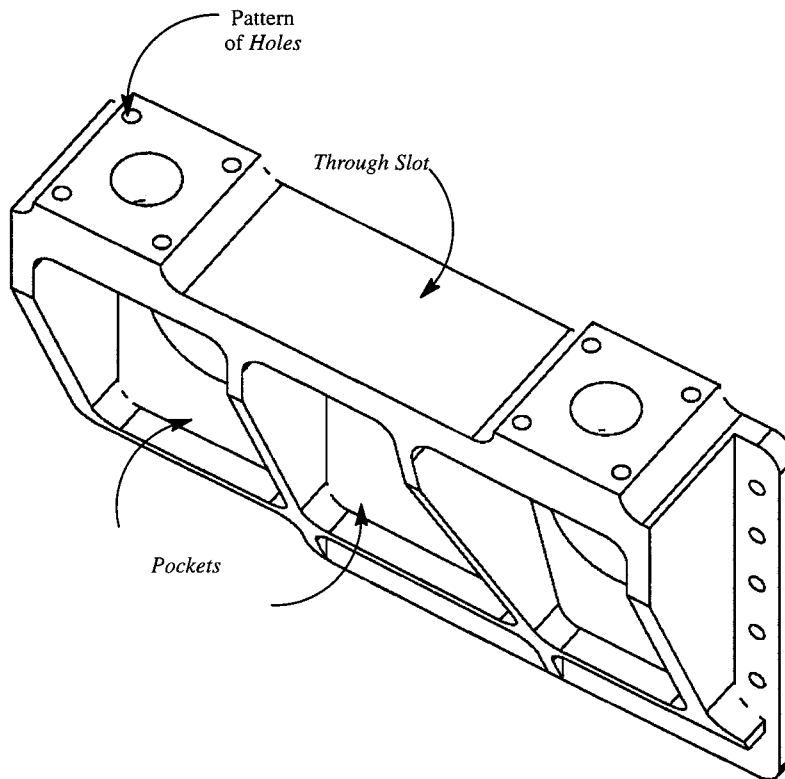


Fig. 1 Pattern of four through fastening *hole* features

Some authors would claim that interaction implies interference between the entities of a feature [1, 3, 7, 9–11]. It is understood here that, in fact, interferences are special cases of feature interactions.

3 RELATED WORK

Feature interaction is an active and important issue but has principally been explored by researchers involved with feature recognition methods. It is considered to be a significant challenge as, while the number of features may be finite, the features resulting from their interactions are infinite [5, 7, 12]. A consequence is that ‘no general approach to recognize all interactions is yet known’ [2, 7].

Design-by-feature systems have misconsidered (or at least misrepresented) feature interactions so that only simple and straight forward interrelationships can be found in the literature. It would be useful to both DbF and FeR systems if, instead of immediately operating with the interaction to produce a (set of) new feature(s), the interaction case were identified beforehand so as subsequently to be able to use this extra information in any downstream analysis.

3.1 Existing feature interaction classifications

Features have a strong volumetric significance [7, 9, 13, 14] which is considered to have the two aspects of volumetric nature [15] for adding (*positive*) or removing (*negative*) material and associated volume [3]. The feature associated volume (FAV) is an important means by which feature interactions are determined.

Pure boundary implementations of features have also been used as the foundation for feature interaction classifications. For instance, a fast interaction identification and classification has been based on polyhedral features [16], but the method was dependent on the internal boundary representation (BRep) scheme and adopted a separate approach for concave and convex features.

There is much evidence to suggest that interaction identification and the specification of its semantics should be maintained as separate processes. The evidence includes:

- (a) the distinction between *spatial* feature interactions and other types of interaction or relation [2, 8];
- (b) the effects or ‘reactional’ interpretation of the interaction phenomena [3, 9];
- (c) the widespread use of feature interaction cases to identify application dependent situations; and
- (d) the fact that different applications could be interpreted differently (valid or invalid) for the same feature interaction [4, 9].

The binding of the interaction case to a specific semantic should be a subsequent reasoning dependent on the application to allow the consideration of information concerning the designer, the product, standards with which to comply, manufacturing processes, etc.

It can be inferred that spatial feature interactions drive other types of interaction and thus should be as accurate, extensive and detailed as possible in order to be used by a great variety of applications. Detailed interaction classification should include various levels such as the volumetric and boundary ones, as suggested by Bidarra and Teixeira [3].

Existing feature interaction classifications, although possibly very efficient, do not comply with any comprehensive classification scheme, are oriented towards specific applications and are thus biased and constrained by their domain. Furthermore, ‘neighbouring’ or ‘adjacency’ of features has been considered to be of crucial importance for applications such as computing tool approach directions [9, 13, 17] but neglected in most classifications because they are not considered to be interactions [2].

It was also observed that many classifications mix different types of geometrical data during analysis [4, 6, 17], resulting in some confusion. For instance, Talwar’s classification placed a feature *contained* by another in a different class to *intersecting* features, but this contradicts the commonsense understanding of the volumetric intersection operation. This mixture problem suggests the need to have a classification framework that could be applied to various levels of geometric information but in a structured and consistent manner.

This paper presents a classification framework to identify spatial feature interactions within the same feature representation space, aiming to overcome the drawbacks listed above. The variety of classifications and interpretations referred to above not only shows how non-standardized this topic is but also shows the importance and widespread nature of the application of feature interaction identification and classification.

4 CLASSIFICATION FRAMEWORK

The idea is to have a basic classification framework that establishes the relationship between any two entities (A and B in Fig. 2) and to reproduce this framework at different levels using the same principle and identification procedure. In addition, simple Boolean expressions are used to identify each category. The elements of the framework, details of the identification procedure and the semantics, categories and levels will be presented first, followed by the overall classification structure.

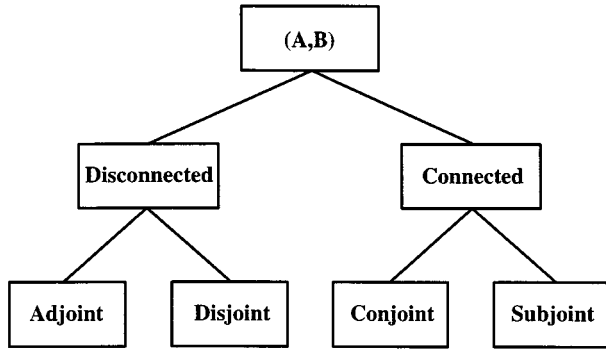


Fig. 2 Basic classification framework

4.1 Entities and levels

The analysis considers pairs of *elements*, called the *joint* A and B, from a specific *entity set*, Σ , with a *relative level*, n , denoted by $(A, B) \in \Sigma^n$. The classification is made according to the results of operations on the joint. Table 1 exemplifies entity sets at various relative levels, together with possible sources of these entities. The classification scheme can be applied to the three levels of volumetric interaction (VI), boundary interaction (BI) and facial interaction (FI). The feature associated volume (FAV) is used in determining volumetric interactions and has been described in Section 3.1. Similarly, the feature associated boundary (FAB) is a closed set of boundaries used in determining boundary interactions and the feature associated surfaces (FAS) is a set of individual faces used in determining facial interactions. It has already been shown that many applications need to know the type of interaction between features and their constituent geometric entities at each of these levels. The established framework is consistent and comprehensive for each of the levels (volume, boundary and face). *Relative level* is a term used here only to clarify and to distinguish between entities with respect to their relative complexity and comparative dimensional representation, and no mathematical meaning or relationship is used or implied.

4.2 Queries to the underlying geometric solid modeller

Two Boolean operators are used to make enquiries of

Table 1 Entity sets and possible sources

Entity set	Entity type	Relative level	Possible source
Σ^5	FAV	$n=5$	CSG representation
Σ^4	FAB	$n=4$	Boundary evaluation of CSG
Σ^3	FAS	$n=3$	Surfaces of a BRep
Σ^2	Edges	$n=2$	Degenerate result
Σ^1	Faces	$n=1$	Degenerate result
Σ^0	NULL	$n=0$	Absence of result

the geometric solid modeller (GSM). These are the non-regularized Boolean intersection (represented as \cap [18]) and the regularized Boolean intersection (represented as \cap^* [18]). Boolean intersection operations are commonly available in GSM representation schemes such as constructive solid geometry and boundary representation. These operators are used to obtain C and D, which are the result of the intersection operations on A and B for a particular Σ^n . Thus, the operations performed by the solid modeller and the results used to classify and subclassify interactions are

$$C = A \cap^* B, \quad (A, B) \in \Sigma^n$$

$$D = A \cap B, \quad (A, B) \in \Sigma^n$$

In addition to these Boolean operations, *set membership* tests are necessary. Tests include:

1. 'Which feature does face F belong to?'
2. 'Is entity X of the same type as entity Y?'
3. 'What is entity W (a volume, face, edge or vertex)?'

Some of this information may be obtained from the feature-based modeller because it is frequently kept in the FBM database as reverse reference pointers to entities in the GSM data structure.

4.3 Identification process

4.3.1 Connected or disconnected?

The regularized Boolean intersection \cap^* of A and B produced the result C that allows interacting entities to be classified as *connected* or *disconnected*.

1. Connected interacting cases occur when C is not NULL. The word 'connected' is used to emphasize that the connection between entities will only occur if an entity of the same relative level as the inputs is used to establish the relationship (the same can be said of the regularized Boolean intersection).
2. Disconnected entities occur when C is NULL or there is no relationship entity of the same relative level between A and B.

Connected and disconnected cases are subclassified by analysing the geometric result D as described in the following sections.

4.3.2 Subclassification of connected entities

Connected entities are those where entities A and B partially occupy the same space and can be sub-classified into *conjoint* (coincident) and *subjoint* (overlapping) interactions. Conjoint connected cases are those where one entity is completely superimposed or inserted into another because the result C of the Boolean operation is identical to one of the original entities ($C = A$ or B).

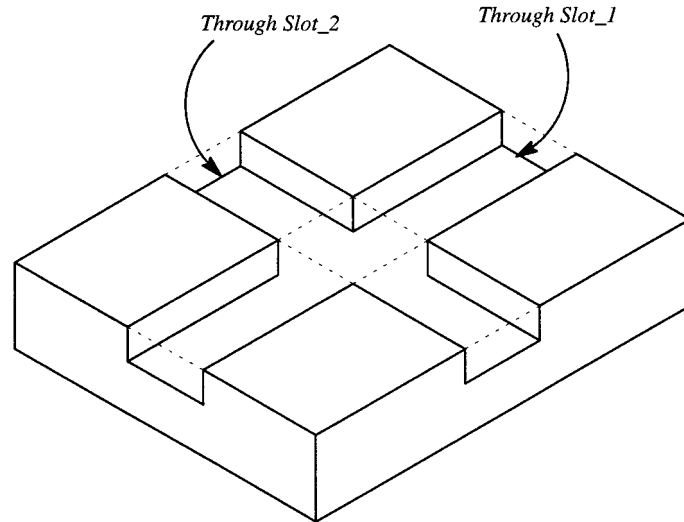


Fig. 3 Connected subjoint crossing volumetric interaction

Conjoint cases can be further subclassified as:

- (a) cases where the inputs A and B exactly *match* each other ($C = A$ and $C = B$, which means that A and B are the same);
- (b) where one entity is completely *inside* the other ($C = A$ or $C = B$ but $A \neq B$ or, simply, if they are conjoint connected but do not match).

Subjoint connected cases (when added to nouns, the prefix ‘sub’ refers to an entity that is part of a larger one) is also known as *overlapping*. In this case C is the entity that is a part of A or B, is of the same relative level and occurs when complex non-standard topologies arise. Such interactions could not affect the entity meaning itself but could have a severe impact on downstream applications. For instance, if overlapping features (e.g. the crossed slots of Fig. 3) are not identified and represented properly, they will result in redundant machining operations if they have the same volumetric removal intention. Subjoint connected cases can be subclassified into:

1. *Enter*, when the end of one entity is completely inserted into another entity. The *end* of an entity is of a lower relative level than the entity itself. For instance the end of a feature is a face, in the same way as the end of an edge is a vertex.
2. *Cross*, when neither end of an entity is inside the other (at the same relative level).
3. A range of other cases that can be identified for pragmatic purposes is left here as a *general* subclass for simplicity.

4.3.3 Subclassification of disconnected entities

Disconnected interacting cases, partially considered by

Shah and Rogers [19], occur when C, the regularized Boolean intersection result, is NULL. Additionally, D happens to be an entity of an inferior relative level. Two situations can occur: *adjoint* (adjacent) and *disjoint* (separate) disconnected interaction.

Disjoint disconnected interaction (the prefix ‘dis’ describes the opposite state) occurs when there is no intersection whatsoever, C and D are NULL and the features are considered to be separate. Disjoint cases could be:

- (a) *Far* when entities are ‘really’ distant from each other (the distance between them is greater than a specified value),
- (b) *Near* when entities, although not touching, are close to each other with no other entity in-between.

Conversely, adjoint disconnected (adjacent or touching) cases occur when D is not NULL and the input entities ‘share a topological entity’ [9, 13] of lower relative level (e.g. the shared edge in Fig. 4).

4.4 Basic classification framework

The framework that uses the entities and procedures presented in the previous sections is shown in detail in Fig. 5 and has to be applied to the three different levels of interest to obtain the complete classification (see Section 4.5). The operations and set membership tests mentioned in Sections 4.2 and 4.3 are reproduced in Fig. 5 for the purpose of clarity. Each line represents a test and each box represents either an operation or a status of the interaction. A and B are the joint entities, C and D are the results of the operations and m and n are the relative levels of D and the inputs respectively. The lower part of Fig. 5 indicates how this classification framework is related to the levels of

interest and presents the few exceptions or special meanings (in brackets).

The framework works in a top-to-bottom fashion in the sense that an application would receive a joint (some A and B entities) that exists at a specified volumetric boundary of the face level. The application determines C and D using the equations defined in Section 4.2 ($C = A \cap B$, $D = A \cap B$) and performs the set membership tests ($D \in \Sigma^m$, $m < n$, $C \in \Sigma^n$) (see top of Fig. 5). This determines whether entities A and B are connected or disconnected and, according to the result C, further tests are performed to obtain the final classification. This classification (adjoint, disjoint, conjoint or subjoint) can be further subclassified or the whole framework is applied once more but at a lower level. Possible routes to other levels or to reach a detailed subclassification (such as near or far) are shown in the table at the bottom of Fig. 5. The symbol \Rightarrow is used in the table to identify where and how the classification framework can be applied again. For example, an adjoint volumetric interaction can lead to a facial interaction evaluation whereas a conjoint volumetric interaction leads to a boundary interaction evaluation. Thus, the classification can go deeper (if required) in order to distinguish between different cases that otherwise would be treated equally.

4.5 Complete classification structure

To apply the framework to the classification of volumetric interactions (VI), the inputs are feature associated volumes and the output for a connected volumetric feature interaction should be a valid geometric volume (solid). If conjoint connected cases occur at the volumetric interaction level (VIs), then the

same structure can be applied to obtain further details by dropping down the structure to consider interactions at the boundary level (BIs). This in turn will lead to a facial interaction (FI) analysis if an adjoint BI interaction case occurs. Similarly, if adjoint disconnected volumetric interactions occur, then various interaction cases could be identified with the same organization structure as the VI cases, but at the face level. These are called facial interaction (FI) cases and are further detailed as shown in Fig. 6. Therefore, adjoint VI or BI cases are linked to many FI interaction relationships as required for each face of the feature realization.

The complete classification tree is given in Fig. 6, where the framework Fig. 5 is repeated four times to include all aspects of the three different levels. A few special meanings and exceptions were found when applying the framework to the three levels (these are shown in Fig. 5 as bracketed words and in Fig. 6 as dashed boxes):

1. An adjoint facial interaction case is called a *limit* because it means that one feature is actually being limited by another.
2. Disjoint boundary interactions are *contained* because they identify that one feature associated volume (FAV) is totally inserted into another FAV and they do not touch from the inside.
3. The interactions presented are not always commutative and thus the interaction relationships have an *active* or *passive* response according to which input entity (A or B) is the reference. Hence, active or passive interactions include *crossing* or *crossed*, *inside* or *outside* and *limiting* or *limited*. The exceptions to the active and passive response are the commutative interactions *match*, *near* and *far*.

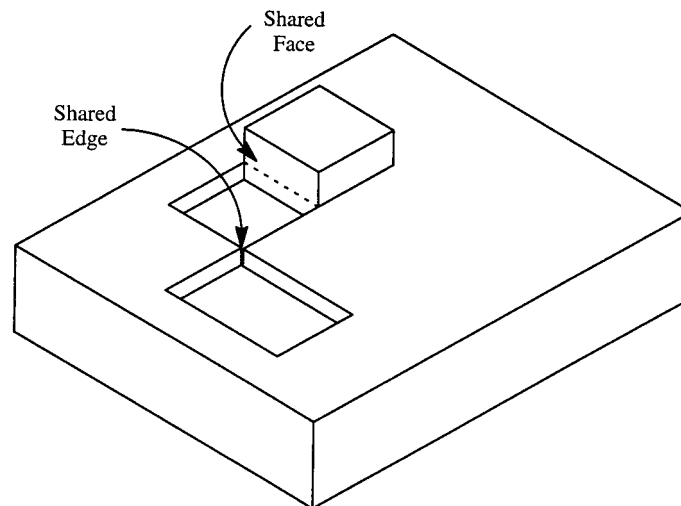


Fig. 4 Adjoint disconnected volumetric interaction caused by features sharing a face or edge

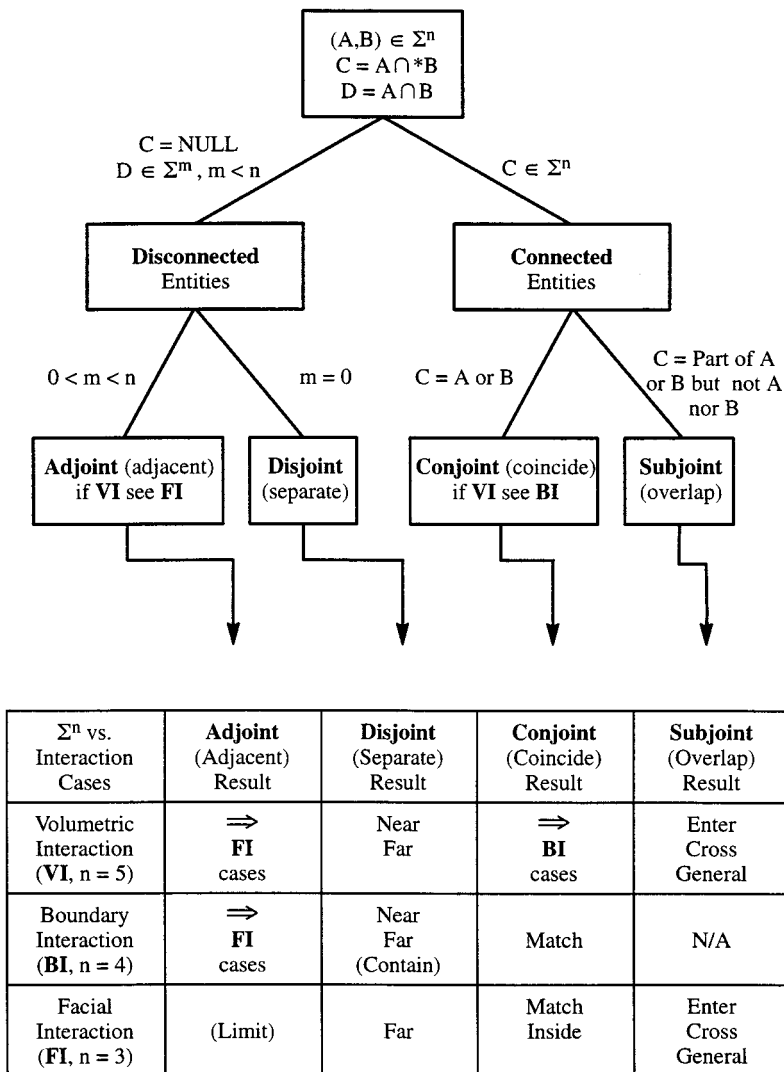


Fig. 5 Basic framework for classifying feature interactions

4. Boundary interactions (BIs) are considered only for conjoint volumetric interaction cases. A disjoint BI means that a feature that is inserted into another FAV does not have external access through the former. It must have an interaction with another feature in order to guarantee, for example, that it has ‘accessibility’ for machining.
5. Boundaries are considered to be a closed set of faces, so two conjoint VI features cannot have a subjoint boundary interaction, as the intersection operation would return an open boundary. Thus, this situation is marked in the table of Fig. 5 as not applicable (N/A).
6. Disjoint facial interactions in fact do not happen at all (if derived from boundary interactions) or have no useful meaning (if derived from an adjoint volumetric interaction).

5 IMPLEMENTATION

5.1 Prototype implementation

This methodology of feature interaction identification has been implemented and bound to a prototype system called FRIEND (an acronym for feature-based reasoning system for intent-driven engineering design) using the bounding box of the feature in order to achieve efficiency in operation. It was found that feature interaction cases can be accurately and quickly predicted using bounding boxes and references to the actual faces of the features. These face references allow the system effectively to apply the interaction determination scheme at lower levels if required.

High levels of interaction can be used for some specific reasonings or can act as filters or approximations for further reasonings at lower levels. Thus, the implementation is

facilitated and accelerated because of this filtering aspect. In the past [2], various feature-based modelling systems have been implemented as layers on top of core geometric solid modellers such as ACIS and Parasolid. The Boolean operations within these core systems are then available for use by the feature modellers. Hence, in this respect the choice of underlying modeller is of no consequence providing that:

- (a) it can successfully perform Boolean operations to obtain intersections of type C and D as described in Fig. 5; and
- (b) it can answer the set membership tests described in Section 4.2.

The prototype FRIEND system is based on the Bentley MicroStation which is itself an implementation of ACIS, and the implementation of the Boolean operations uses

standard MicroStation facilities to operate on the bounding boxes of the features.

The DbF prototype system FRIEND has been implemented with special attention paid to representation validation of feature-based models [20]. Representation validation is the process of analysing the model against the expected behaviour and characteristics of features. Feature validation is required when features interact [1] and thus the study of feature interaction is especially useful for feature validation [2].

In order to carry out such a validation process a complete scenario of interactions is built, identifying each case between every single pair of features and, if required, its components (lower levels). This interaction scenario is then analysed by a knowledge-based system in a search for compliance of the model with predefined general validity properties that, sometimes, imply identifying actions to be taken. This binding of the interaction cases with an action

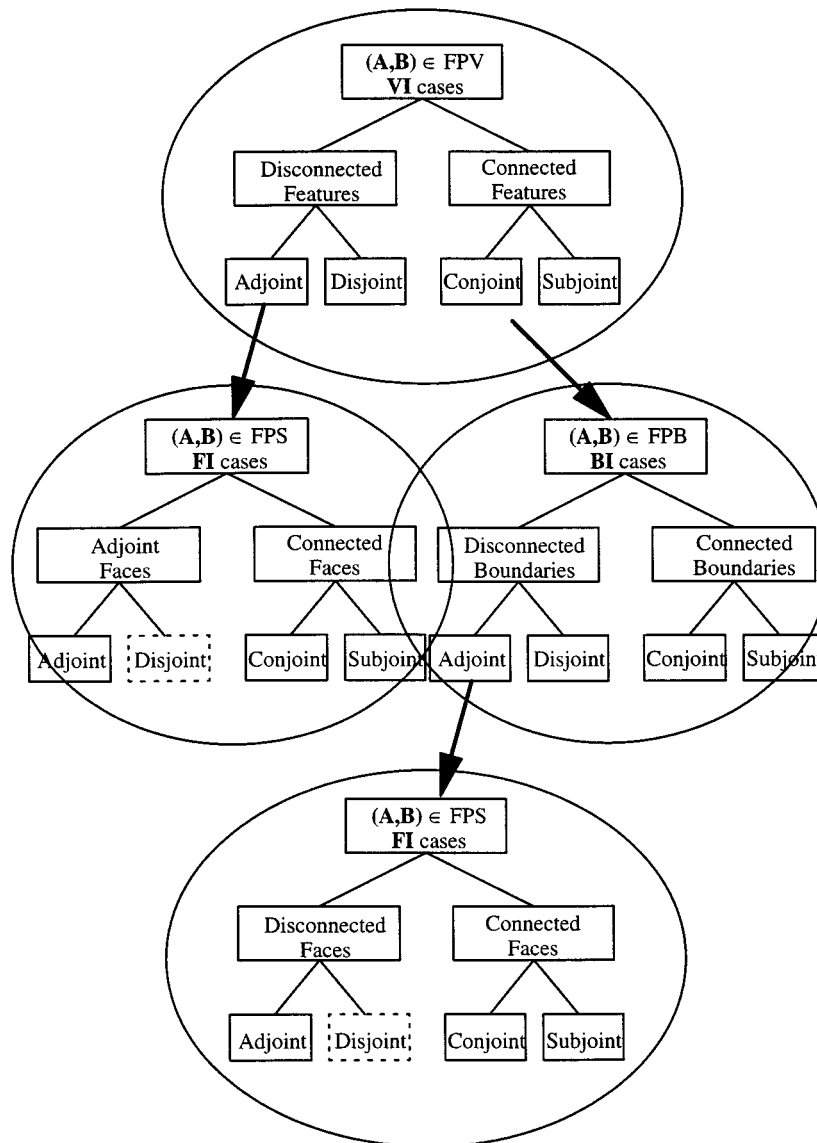


Fig. 6 Complete feature interaction classification tree

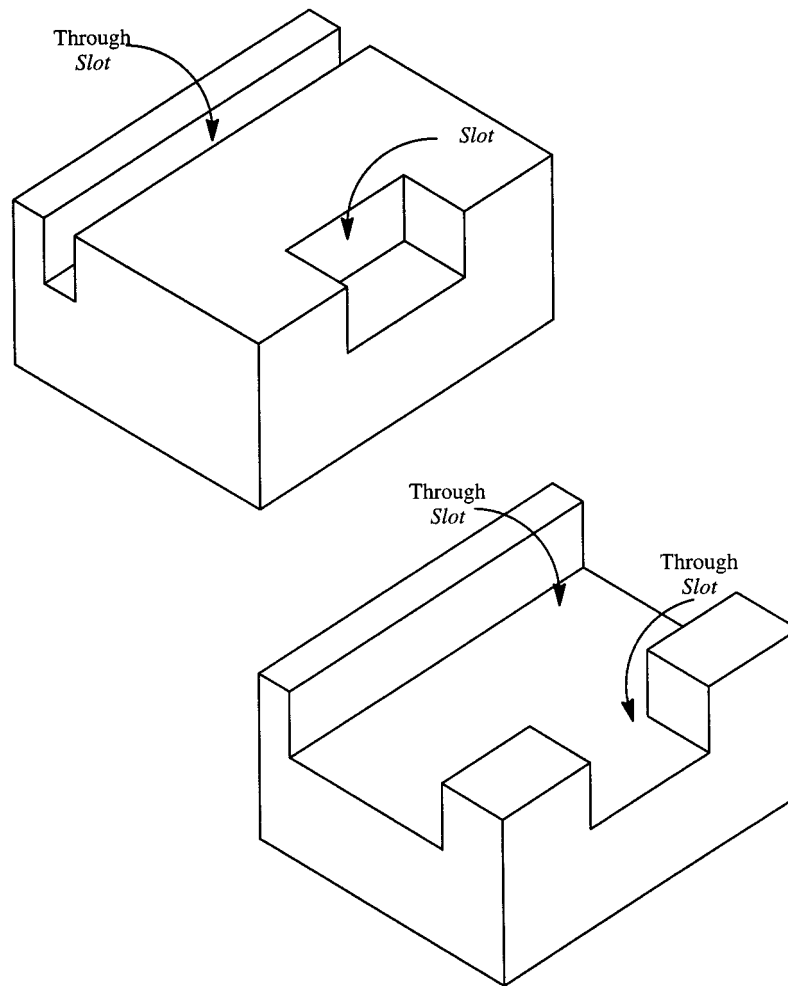


Fig. 7 Blind slot becomes a through slot

via rules is the process that allows FRIEND to perform its allotted task of feature-based representation validation. The feature interaction case together with the face characteristics (*real* or *virtual*) and nature of the feature comprise a vocabulary for FRIEND that makes it capable of reasoning about morphological functional intents [21].

5.2 Using feature interaction cases

As an example of reasoning, a disjoint boundary interaction means that one feature is contained within another and analysis of their feature associated boundaries will lead to the determination of a near or far case. A near condition may imply an ‘internal thin wall’ problem, while a far condition with no other interaction may be interpreted as a *hollow* in the part. With adjoint volumetric interactions there is the possibility of a merging operation (if a *matching* conjoint facial interaction (FI) has occurred) or a change in the feature properties from *blind* to *through* (if an *inside* conjoint FI case has occurred, see Fig. 7). For disjoint volumetric interactions and features as entities, the

threshold between near and far should be computed by a separate ‘thin wall reasoning’. Disjoint VI cases can be further distinguished according to their spatial inter-feature relationships such as *parallelism* and *coplanarity*.

Additional examples of using feature interaction cases can be found scattered in the literature (a brief survey of applications can be found in reference [8] as life cycle interactions) but, as mentioned in Section 3.2, determination of the meaning of the interaction case should be left to a separate reasoning because it depends heavily on the application being considered.

It is important to understand that feature interactions can occur at any stage of an ongoing geometric design process. Indeed it was a major objective of this research to provide methodologies that would be appropriate to detecting potentially invalid conditions as they are generated by the designer, rather than simply relying on a post-design evaluation of the finished geometry.

Hence, several of the interaction cases formally identified here are considered by some feature-based systems [3, 5] as invalid situations that should be reasoned with and

preferably eliminated. For instance, a feature that is disconnected from the stock material through a volumetric interaction is most probably an error, regardless of its nature. In other situations the very same interaction can help confirm (validate) the properties expected from a particular feature type. For instance, a through hole should have both its ends *inside conjoint FI adjoint VI* to other features (in this way it will have clearance at both ends and then behaves as a 'real' through feature).

For these reasons it is not possible to illustrate all feature interactions by reference to manufacturable components, although some of the intrinsic meanings have been explored in Sections 4.5 and 4.6. The methodology has been more formally validated [22] using a range of test components employed by other researchers [1, 5, 22–25] and some aspects are illustrated here using a typical engineering component.

The component (Fig. 8) shows a number of interactions at various levels:

1. At the volumetric level there is a pattern of holes A that are *far disconnected volumetric interactions*.
2. However, the three large holes B inside the pocket are *near disconnected volumetric interactions* and could be indicative of thin wall problems.
3. One of these large holes, B1, is *disconnected adjoint VI adjoint FI* to the uppermost pocket C because it shares a topological entity (a face).
4. Holes B are all *inside conjoint FI adjoint VI* to lower pocket D.
5. Holes E placed in the wall around the pocket are all *limited (adjoint BI conjoint VI)* to pocket C.

Many other interactions that are no longer evident may have occurred while designing this part. For instance, it is most probable that holes A in the pattern would have entered the step feature F that surrounds the pocket.

5.3 Advantages

The feature interaction classification framework and the identification methodology presented here have several advantages:

1. The scheme is DbF-aware and encompasses existing approaches/classifications from both the feature recognition and the narrower design-by-features domains.
2. It adds a comprehensive unbound (and therefore, unbiased) coverage and a clarification of the *interference* and *interaction* terms to avoid misunderstanding.
3. It is multilevel which allows reasoning to be performed at all three levels.
4. All levels share the same structure and concept of classification (except for a few minor details), promoting the consistency of the scheme. This also avoids misunderstandings because there is no mixing up of entities at each level.
5. The classification categories are formally defined through simple rules using commonly available GSM Boolean operators and tests. This facilitates its integration with GSM modellers, but the method remains independent of the underlying GSM implementation.

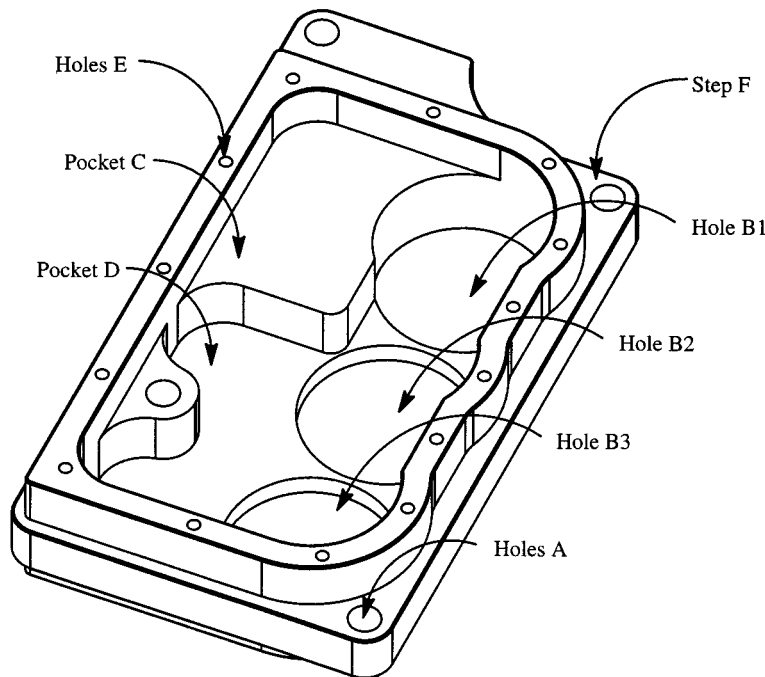


Fig. 8 Example component showing feature interactions

6. Neither concave/convex nor planar/non-planar assumptions are made, to the minor detriment of efficiency. However, many of the operations and tests can be quickly and accurately predicted using bounding boxes.
7. Particular attention was paid to selecting words to describe each category, aiming to produce a more clear, meaningful and easy to understand vocabulary.
8. The cases are as detailed and accurate as required, allowing specific actions to be taken for apparently similar cases and also allowing it to be used by and separated from different application analysis.

Because the feature interaction scheme presented here allows a system to build a scenario of the model, including information about the feature surroundings, it can be used by applications where this information is essential. Some functional meanings (or functional features) can be identified, such as a *slot* being *near* the stock end, resulting in a *wall* feature or a *slot near* another *slot*, creating a *rib* feature. These identified features have totally different functional purposes [18].

6 CONCLUSION

A new feature interaction classification framework has been presented that allows a comprehensive and unified feature interaction classification structure to be conceived. The classification framework has many advantages such as accuracy (even using bounding box data), power (identifies complex cases), elegance (easy to understand), consistency (has a formally defined structure that repeats itself), a multilevel nature (works at volumetric, boundary and face levels) and simplicity (uses simple GSM-based operators and tests). It requires almost no knowledge of the intricacies of GSM representation schemes, although some efficiency is lost because of this.

The identification has been bound to a representation validation reasoning system [20] and has become an essential element to the vocabulary of this application. The interaction classification also allows the application to anticipate some manufacturing related analysis and facilitates many others.

REFERENCES

- 1 **Martino, T.D.** and **Giannini, F.** The role of feature recognition in future CAD systems. In Proceedings of IFIP International Conference on *Feature Modelling and Recognition in Advanced CAD/CAM Systems*, 1994, Vol. 1, pp. 343 – 355.
- 2 **Allada, V.** and **Amand, S.** Feature-based modelling approaches for integrated manufacturing: state-of-the-art survey and future directions. *Int. J. Computer Integrated Mfg*, 1995, **8**(6), 411 – 440.
- 3 **Bidarra, R.** and **Teixeira, J.C.** A semantic framework for flexible feature validity specification and assessment. *Trans. ASME, Computers in Engng*, 1994, **1**, 151 – 158.
- 4 **Zhang, K.F.** and **Elmaraghy, H.H.** Validity check for a functional-oriented modeler. *Trans. ASME, Adv. in Des. Automn*, 1993, **DE-65-2**, 293 – 300.
- 5 **Mill, F.G., Salmon, J.C.** and **Pedley, A.G.** Representation problems in feature-based approaches to design and process planning. *Int. J. Computer Integrated Mfg*, 1993, **6**(1 – 2), 27 – 33.
- 6 **Anderson, D.C.** and **Chang, T.C.** Geometric reasoning in feature-based design and process planning. *Computers and Graphics*, 1990, **14**(2), 225 – 235.
- 7 **Tseng, Y.-J.** and **Joshi, S.B.** Recognizing multiple interpretations of interacting matching features. *Computer Aided Des.*, 1994, **26**(9), 667 – 688.
- 8 **Regli, W.C.** and **Pratt, M.J.** What are feature interactions? In Proceedings of ASME Design Engineering Technical Conference and International Computers in Engineering Conference, Irvine, California, 1996, Vol. DRM-1285, pp. 1 – 12.
- 9 **Shah, J.J.** Philosophical development of form feature concept. Report P-90-PM-02, CAM-I Inc., Arglington, Texas, 1990.
- 10 **de Silva, R.E., Wood, K.L.** and **Bearman, J.J.** Representing and manipulating interacting interfeature relationships in engineering design for manufacture. *Trans. ASME, Adv. in Des. Automn*, 1990, **DE-32-1**, 1–8.
- 11 **Gadh, R.** and **Prinz, F.B.** Automatic determination of feature interaction in design-for-manufacturing analysis. *J. Mech. Des.*, 1995, **117**(March), 2 – 9.
- 12 **Kumara, S.R.T. et al.** 3D interacting manufacturing feature recognition. *Ann. CIRP*, 1994, **43**(1), 133 – 136.
- 13 **Pratt, M.J.** Synthesis of an optimal approach to form feature modelling. In Proceedings of Computers and Engineering Conference, 1 – 3 August 1988, Vol. 1, pp.263 – 274.
- 14 **Li, R.-K.** and **Yu, M.-H.** A framework for prismatic part-data generation — unit-machined loop concept. *Int. J. Computer Integrated Mfg*, 1990, **3**(2), 96 – 111.
- 15 **Lenau, T.** and **Mu, L.** Features in integrated modelling of products and their production. *Int. J. Computer Integrated Mfg*, 1993, **6**(1 – 2), 65 – 73.
- 16 **Talwar, R.** and **Manoochchri, S.** Algorithms to detect geometric interactions in feature-based design system. *Trans. ASME, Adv. in Des. Automn*, 1994, **DE-69-1**, 307 – 314.
- 17 **Lee, R.J.V., Al-Ashaab, A.H.S.** and **Young, R.I.M.** Resolving feature interactions in design for injection moulding. *Int. J. Adv. Mfg Technol.*, 1994, **10**, 274 – 279.
- 18 **Zeid, I.** *CAD /CAM: Theory and Practice*. Computer Science Series, 1991, Vol. 1 (McGraw-Hill International Editions).
- 19 **Shah, J.J.** and **Rogers, M.T.** Expert form feature modelling shell. *Computer Aided Des.*, 1988, **20**(9), 515 – 524.
- 20 **Hounsell, M.S.** and **Case, K.** Representation validation in feature-based modelling: a framework for design correctness analysis and assurance. In Proceedings of NCMR 12th National Conference on *Manufacturing Research*, Bath, 1996, Vol. 1, pp. 256 – 261.

- 21 Hounsell, M.S.** and **Case, K.** Morphological and volumetrical feature-based designer's intents. In *Advances in Manufacturing Technology XI* (Ed. D.K. Harrison), 1997, pp.64 – 68 (Glasgow Caledonian University). Proceedings of 13th National Conference on *Manufacturing Research*, Glasgow, 1997.
- 22 Hounsell, M.S.** Feature-based validation reasoning for intent-driven engineering design. PhD thesis, Loughborough University, Loughborough, 1998.
- 23 Chang, T.-C.** *Expert Process Planning for Manufacturing*, 1990 (Addison-Wesley, Reading, Massachusetts).
- 24 Perng, D.-B.** and **Chang, C.-F.** Resolving feature interactions in 3D part editing. *Computer Aided Des.*, 1997, **29**(1), 687 – 699.
- 25 Mantyla, M., Opas, J.** and **Puhakka, J.** Generative process planning of prismatic parts by feature relaxation. In Proceedings of ASME-DE 15th Design Automation Conference, Montreal, Canada, 1989, Vol. 19-1, pp. 49 – 60.