In *Advances in Manufacturing Technology XI*,
Proceedings of the Thirteenth National Conference on Manufacturing Research,
Editor D K Harrison, Glasgow Caledonian University, Glasgow, September 9-11, pp64-68, 1997

Morphological and Volumetrical Feature-based Designer's Intents

M. S. Hounsell and K. Case
Manufacturing Engineering Department, Loughborough University
Lough borough, LEICS, LE II 3TU, UK

Features are claimed to be the carriers of Designer's Intents (DI's) which are seldom defined, identified and represented in Design-by-Features **(DbF)** systems. This paper presents an interpretation of Designer's Intents for the Feature-based Modelling **(FBM)** context and emphasis will be given to the Morphological Functional and Volumetrical Geometrical DI's which express the basic behaviour of a DbF system. DI's are also an important part of a validation system capable of reasoning about the semantics of using features in a particular design. If features' characterisations via DI's are well established and measurable the representation could be assessed as to its conformity with feature's meaning and their semantics could be validated. It is considered that the better Designer's Intents are understood and specified, the more useful Feature-based Modelling will become.

Introduction

Conventional CAD systems are recognised as being difficult to integrate with other activities such as engineering, manufacturing, process and production planning because they "are incapable of capturing non-geometric aspects of designer's intent such as tolerances, parts relationships, surface finish, etc." [1][M1]. Also, more abstract design activities such as conceptual design, generation of design alternatives, reuse and reasoning on design procedures and capturing the functionality of a product are just impossible [2][M2]. Design-by-Feature systems are considered one approach to overcome some of these drawbacks. However, although many FBM systems have explicitly claimed to capture and represent designer's intents to some extent, [M3] few attempts [2, 3] were found that effectively define, clarify and identify DI's in the feature-based design context.
No precise definition for designer's intents exists in the context of FBM and it has been acknowledged that “the information that constitutes intent, and how to capture and use intent are all research issues to be explored" [4][M4]. Thus, it is herein interpreted that "Feature-based Designer's Intents (DI's) are a wide variety of concerns that help decide on a specific geometric attribute or configuration to achieve (a set of) higher level and more abstract designer's intents (ADI's). Designer's Intents (DI's) act as a bridge between the (set of) abstract designer's intents (ADI's) and the geometric model. DI's represent information which should be verified and maintained throughout the Detailed Design Process and could be used as constraints to drive the decision-making process of a downstream application. To some extent, it could be said that Feature-based Designer's Intents represent means to implement abstract designer's intents.

Abstract designer's intents (ADI's) include morphological functional, theoretical functional and application-dependent ones. A taxonomy of DI's concerned with the feature-based geometric detail design phase for prismatic machining parts has been established and includes Volumetrical, Dimensional and Structural Geometrical DI's. This paper discusses Morphological Functional abstract designer's intents (ADI's) and Volumetrical Geometrical designer's intents (DI's). It is considered that with such a taxonomy and clear definition of DI, agents can be conceived to effectively capture, represent and maintain intents through feature-based modelling. DI's can be expressed by relationships between features themselves or elements of the feature-based model such as features' faces (and their attributes) and features' parameters.

Functional Designer's Intents

Features represent a good means to embed functional significance and this fact can be inferred by some definitions applied to features. Features have been defined as "the addition of functionality to geometric forms" [1, 4], "high-level morphological information with well-defined functional meaning" [6] and, "high-level functionally significant entities" [5]. These definitions clearly expose features as having a **morphological functional** ADI. For instance, "if an application considers only functional morphological information, then the term form-feature can be used" [7]. The morphological functional ADI identifies shape characteristics that a feature must comply with and imprint on the part. If such an aspect is changed, the feature is possibly no longer of the specified type (say a *slot*).

In addition to the topological and geometrical analysis usually applied to identify features (as in feature Recognition approaches), extra functional analysis could be performed for a better specification of the elements of a feature family. For instance, a *cylindrical boss* family of features could be specialised into a *disk* if limited to a certain height-to-diameter ratio range, otherwise it becomes a *rod* [3] and, this specialisation could lead to a drastically different manufacturing approach. In this case it could be machining the *disk* or welding the *rod.*
Nevertheless, Functional ADI 's are also linked to the "function concept" itself which is defined as "the behaviour of an object, an operation of energy, material, information or signal that tells what the design does" [2] and, "include not only in-use purpose, but also manufacturing and life-cycle considerations" [4]. Although some research has addressed the relationship between form and function, it is not formally understood yet because of many difficulties. Firstly, the abstract nature and understanding of the *function* concept. Secondly, a function can be a composite result of many interacting sub-functions. Lastly but not least, a given function could be performed by several forms and one form could be used to perform many different functions [8]. This "functional concept" is sometimes implemented as physics-based or engineering-based laws or formulae depending on the subjects being considered (say heat propagation, transference of torque/force or stress analysis).

Thus, Functional ADI's are considered to have at least two branches: *Morphological* and *Theoretical.* The former is intrinsically related to FBM and the last is difficult to deal with in a broad and pragmatic perspective and is outside the scope of this research.

Volumetric Geometrical Designer's Intents

Volumetric Geometrical Designer's Intents (**VGDI's**) implement morphological functional ADI's within the geometrical realm and are concerned with the feature's expected behaviour. Volumetrical GDI's are to be considered specially when an interaction between feature's volumes occurs. Intermediate states, delete operations and editing manipulations have direct influence over the VGDI's in a design. To deal with VGDI the semantics of non-conflicting and conflicting interaction between features must be defined. These situations could result in normal, obsolete (redundant) or undesirable cases (such as *hollows* and *satellite* features).
Labelling VGDI identifies the relationships between all feature's faces and their attributes. Every feature has a set of labelling relationships that is kept as the feature's *label.* Labelling is basically implemented by defining a template of *virtual* and *real* faces that bound the produced volume of a feature type. *Virtual* faces basically identify tooling external access directions and *real* faces identify surfaces to be imprinted on the part. If a face of a given feature abuts and is completely inserted into another feature's *real* face then, the former must be a *virtual* face [9]. Using reasonings such as this the labelling aspect can be maintained. If the template and the realisation do not match, the feature's *label* is invalid, and a "revalidation process" [10] called <u>search label</u> will then search for the right match. The search_label process is responsible for keeping the label-to-shape relationship matching as defined by the template of every feature's type. A feature's *label* is dependent on the feature's positioning and thus, changing feature positioning could possibly lead to a change in the feature's *label.* In addition to establishing a label-to-shape relationship, features are usually expected to imply a

volumetrical behaviour, which is called the feature's *nature* by [11], of *adding* material (when it is said to have a *positive* volume) or *removing* material (when it is said to have a *negative* volume) from the stock. A feature's *nature* is identified by a volume description and a boolean operation (union for a *positive* volume and difference for a *negative* volume). The feature's *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 feature's requirement and ability to change the existing model is called the **changeability** VGDI. The changeability requirement invalidates *obsolete features* [8] that occur when a feature is completely inserted into another and has the same *nature*. However, it does not require that all the boundaries of the feature's produced volume should be shaped into the part.
A feature's *nature* (volume and boolean operation description) is closely and complementarily related to a feature's *label* (its positioning and template of *real* and *virtual* faces). The *label* is considered to be the link between the geometric modelling realisation and other non-geometrical information associated with a specific feature type. For instance, the same *nature* and same positioning but slightly different template description would result in a different feature *label* (such as for *pocket* and *hole* features).

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** VGDI. The fittability requirement invalidates *feature's parameters made obsolete* [8] where feature's parameters do not describe exactly the extent of what it imprints on the part. An example of fittability reasoning occurs when two features touch each other with a perfect face match. If such a situation happens, the "revalidation processes" of merging and search label should be performed or suggested by the FBM system. However, adequate parameters could also mean that part of the feature does not affect the component and hence, the *hole* in Figure 1.a is a valid representation as a single long *through-hole.*

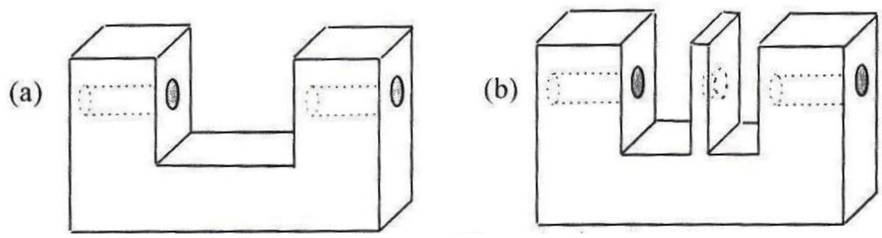

Figure 1: Redundant volumetric intents analysis

Capturing and maintaining VGDI's is a subjective problem. For instance, simple operations such as creating a *boss* inside a *slot* could be interpreted as an acceptable situation (when it leads to the formation of a protrusion, see Figure 1.b) or could be interpreted as a VGDI conflict because a material *removal* intent existed and then a contradictory requirement, *addition* of material, is superimposed. Conversely, inverse operations (superimpose a material *removal* over an *addition* intent) can cause hollows (usually undesirable) or, the vanishing of the initial *addition* intent (deleting the previous feature at that same location). Furthermore, interesting and difficult situations arise when redundant intents are found. Suppose a part composed by a *slot* and a *hole* (Figure 1.a), is modelled as a single long cylinder. There is a **redundancy** of VGDI's where the *hole* intersects the *slot.* This is a feature interaction problem that has been receiving much attention in the literature as being of special difficulty to handle (see [12] and *The Contiguity Problem* in [8]).

Thus, whether to accept the redundant *removal* intent in the representation or to split the *long hole* into two *short hole* features has consequent implications: If the redundant portion is allowed, care should be taken during further analysis to avoid redundant manufacturing operations. On the other hand, if two separate *short hole* features were created, the redundant geometric part must be deleted from the model and other types of DI's (such as the *equal radius* and *concentricity* intents between the two *short holes)* must be added to the representation .

However, eliminating the redundant part also eliminates the *removal* intent at that location. For instance, consider adding a *boss* inside a *slot* (Figure 1.b) after deleting the redundant *removal* intent. Should this *boss* have the former *hole* intent as well? Another possible reasoning includes how to perform representation optimisation from the application's point-of-view without affecting on-going volumetric intents: i.e. how to split the volumetrical intent, based on process planning cost criteria, for example, and merge/label them afterwards.

Similar DI management problems appear when simple delete operations are required during an editing session. A non-procedural volumetric implementation of features assigns a volume to be used as a shape builder. Thus, imagine the primitive volumes used to produce Figure 2.a, a component formed by a *step*, a *slot* and a *hole* feature. Now imagine re-adding the primitive volume that produced the *slot* (Figure 2.b) as a way to delete the *slot.* Not only is the *slot* deleted but the *step* is turned into two *notches* and, the *hole's* shape is also affected . A clear unwanted volumetric DI scenario has emerged instead of the simple deletion of the *slot* (Figure 2.c). This is a matter of managing redundant VGDI's. If the system was able to identify those portions of the feature represented by a redundant material removal (like identifying *required* and *optional* portions suggested by [13]) then, it would have known that the deletion of that feature should not be made by just re-adding the feature's primitive volume, although for deletion cases with non-redundant intents re-adding the feature's primitive volume suffices.

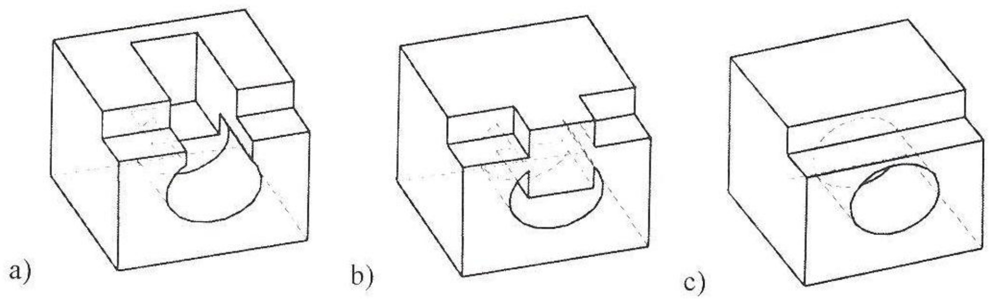

Figure 2: Editing problems and volumetrical intents

Intent-driven Feature-based Design

A prototype system called **FRIEND**, short for **F**eature-based **R**easoning system for **I**ntent-driven **EN**gineering **D**esign, has been implemented with special concern for the validation of feature-based geometric design representations [10]. A clearer definition of feature's semantics within FRIEND was achieved with the help of the Morphological Functional and Volumetrical Geometrical DI’s presented here. The verification reasoning is based on the spatial geometrical feature interaction (such as those exemplified here: abuts, touches, inserted) applied at various levels, such as the feature’s volume and feature's face levels. Taken together, the feature's interaction and feature's data structure (which includes feature's *nature* and *label)* offer a vocabulary that permits a knowledge-based system to reason and validate the feature-based model. **FRIEND** represents all the relationships mentioned above and uses a knowledge-based environment (rule-based system) to reason with them.

Conclusion

It can be said that Feature Recognition approaches represent an attempt to identify Morphological Functional ADI's *(label, nature,* etc.) from the geometrical data but a Design-by-Feature approach provides a richer, more powerful and more specific way to capture, represent and reason with DI's as defined and presented here.

The definition and identification of Designer's Intents presented here allowed the implementation of a prototype system capable of reasoning and validating with a feature-based model. A better understanding and categorisation of the meaning of Designers Intent within a design-by-feature CAD system is a necessary step in order to foresee how feature-based reasoning could be embedded in future Intelligent CAD systems that really support detailed designs and allow its integration with other engineering activities.

Special agents were identified that allows **FRIEND** not only to verify the model, but also maintain the model validity by operating on it. These "revalidation operations" appear underlined throughout this text and include split, merge, delete and search_label operations. The selective firing of these operations guarantee that **FRIEND** delivers valid representations from the Morphological Functional and Volumetrical Geometrical perspectives.

References

1. Nnaji, B. O., H.-C. Liu, and U. Rembold. "A product modeller for discrete components". *International Journal of Production Research,* Vol. 31(9), pp. 2017-2044. 1993.
2. Henderson, M. R. "Representing Functionality and Design Intent in Product Models". *Second Symposium on Solid Modeling and Applications,* Vol. I, pp. 387-396. 1993.
3. Nielsen, E. H., J. R. Dixon, and E. E. Zinsmeister. "Capturing and using designer intent in a design-with-features system". *DTM'91: 3rd. Int. Conf on Design Theory and Methodology,* Vol. 31, pp. 95-102. 1991.
4. Dixon, J. R., E. C. Libardi Jr, and E. H . Nielsen. "Unresolved Research Issues in Development of Design-with-Features Systems". *IFIP WG 5.2/NSF Working Conf on Geometric: Modelling,* Elsevier SCI. Pub., Vol. I, pp. 183-196. 1990.
5. Laakko, T. and M. Mantylä. "Feature modelling by incremental feature recognition". *Computer Aided Design,* Vol. 25(8), pp. 479-492. 1993.
6. Gomes, A.J.P. and J.C.G. Teixeira. "Form Feature Modelling in a Hybrid CSG/Brep Scheme". *Computers and Graphics* (Pergamon Press), Vol. 15(2), pp. 217-229. 1991.
7. Dohmen, M. "Constraint techniques in interactive feature modeling". *TUDelft Report 94-16. Delft University of Technology, Faculty of Technical Mathematics and Informatics,* 1994.
8. Shah, J. J. and M. Mantylä. *"Parametric and Feature-based CAD/CAM".* John Wiley and Sons, Inc. New York. USA. 1995.
9. Silva, R. E. d., K. L. Wood, and J. J. Bearman, "Representing and Manipulating Interacting lnterfeature Relationships in Engineering Design for Manufacture". *(ASME) Advances in Design Automation,* Vol. DE-32-1, pp. l-8. 1990.
10. Hounsell, M. S. and K. Case. "Representation Validation in Feature-based Modelling: A Framework for Design Correctness Analysis and Assurance". 12th *National Conference on Manufacturing Research,* UK. Vol. 10, pp. 156-161. 1996.
11. Lenau, T. and L. Mu . "Features in integrated modelling of products and their production". *International Journal of Computer Integrated Manufacturing,* Vol. 6(1-2), pp. 65-73. 1993.
12. Salomons, O. W., F. J. A.M. van Houten, and H. J. J. Kals. "Review of Research in Feature-Based Design". *Journal of Manufacturing Systems,* Vol. 12(2), pp. 113-132. 1993.
13. Vandenbrande, J. H. and A. A. G. Requicha. "Spatial Reasoning for Automatic Recognition of Machinable Features in Solid Modeling". *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 15(2), pp. 1269-1285. 1993.