



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/>

A TAXONOMY OF FEATURE-BASED DESIGNER'S INTENTS

M. S. Hounsell and K. Case

Manufacturing Engineering Department, Loughborough University.
Loughborough, LEICS, LE11 3TU, England.

ABSTRACT

Feature-based modelling is considered an improvement on existing CAD systems. Features are considered to be a medium that carries designer's intents, but neither features nor designer's intent have widely accepted definitions. Morphological functional designer's intents, defined as common-sense behaviours of the (form) feature's concept, have been defined and presented within a feature-based representation validation system [4].

The process of "feature elicitation", frequently implied to identify and categorise features, comprises "featurization" and "featurization validation" processes which help specify an appropriate feature library to be used in a particular application. In the research reported here a similar approach, called "intent elicitation", has been performed to identify and categorise meaningful and measurable designer's intents from the integrated CAD/CAM and Computer Aided Process Planning (CAPP) domains.

The resulting classification and taxonomy is presented in this paper. It can be observed that the classification encompasses morphological feature-based designer's intents (**FbDI's**), because of the feature's concept, and is application dependent. The identified FbDI's have been used in a feature-based reasoning system which has led to an intent-driven approach for feature-based modelling where designer's intents are an explicit and central aspect.

1. INTRODUCTION

Feature-based Modelling (**FBM**) is considered to be the underlying technology for the next generation of Computer Aided Design (**CAD**) systems and Concurrent Engineering environments in the same way as Geometric Solid Modelling (**GSM**) is considered to be the underlying technology for existing CAD systems. Some advantages of FBM are the use of a more friendly environment, meaningful entities and manipulations, the ability to store information beyond geometry and, the consequent possibility of integration with other engineering applications such as manufacturing, process planning, etc. In many respects, FBM is considered an evolution of GSM. However, one basic element that makes GSM so well established, important, popular and powerful, namely *Geometric Validation*, lacks a sibling in the FBM world. This is so because features add a layer of complex semantics, which are difficult to measure and subjective to implement. Feature-based representation validation is very important because it is the process responsible for guaranteeing the delivery of a valid representation (and therefore verified, useful and misrepresentation free) to a downstream application.

Designer's intents represent information that should be verified and maintained throughout the detailed design process and could be used as restrictions to drive the decision-making process of a downstream application. Because they are considered intrinsic to features, they are sometimes left out of the formal and explicit description of a design. Nevertheless, FbDI's are suitable to act as a verification medium to perform feature-based representation validation.

2. FEATURE'S EMBEDDED DESIGNER'S INTENTS

1 Definition

It has been acknowledged that "the information that constitutes intent, and how to capture and use intent are all research issues to be explored" [1]. Thus, it is herein defined that

Feature-based Designer's Intents (FbDI's) represent 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 themselves 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 that a feature provokes with the existing surrounding features in the model.

2 Volumetric Designer's Intents

Four Volumetric Designer's Intents (VDI's) have been observed in a feature-based model (detailed in [4] and briefly reproduced here for completeness, see Figure 1):

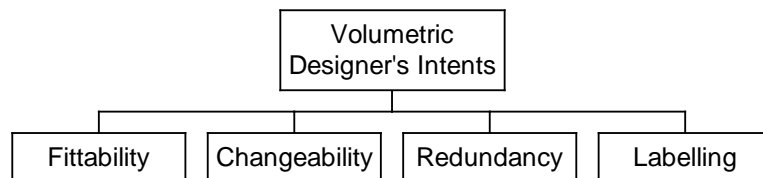


Figure 1: Volumetric Designer's Intents.

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 feature's requirement and ability 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.

3. TOWARDS AN INTENT-DRIVEN APPROACH

1 The Need for a Designer's Intent Taxonomy

A validation system need not be used solely for conceptual representation validation and the associated *volumetric intention* reasonings. It could also be extended to validate various other types of FbDI's and therefore become an intent-driven reasoning system. To identify and

understand these other types of FbDI's the process of "entities elicitation" which has been applied for features is adopted.

Depicting all sets of FbDI's present in the designer's mind is beyond the scope of this research and is a very cumbersome approach even in a limited domain. The objective of this research is 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.

4. FEATURIZATION

1 The Featurization Process

Some guidelines have been suggested to perform "feature elicitation", i.e.: to obtain a feature library for a given domain, also called "featurization" [5, 11]. These guidelines have intrinsic *elicitation criteria* that emphasise characteristics of the domain.

Classification schemes have been proposed to ease the task of *features elicitation* and to facilitate the understanding of a feature domain and its functionality. This has been done by categorising features using shared behaviours and characteristics. Various classifications have been proposed but it has been stated that "their differences emphasise the difference in feature views between researchers even when they share a similar interest in the same application" [6].

The subsequent step is to better identify and enumerate individual features (entities) for use in the particular application context. This produces "feature taxonomies". Feature taxonomies have been divided according to the pair of process-product type and/or their cross-section shapes [9]: Rotational Features; Prismatic Features; Thin-Walled Features, and others.

2 "Featurization" Validation

After the process of featurization, a set of feature candidates is produced and should be validated against the chosen application. *Featurization validation* is the process of selecting a reasonably small (or minimum) subset of all feature candidates raised from the elicitation phase, in a specific domain, that demonstrates the best properties (including expressiveness and flexibility) to suit an application. Examples of *featurization validation criteria* include [5, 11]:

- * *Completeness*; is the identified set capable of creating all parts of the chosen domain?
- * *Unambiguity*; do the proposed parameters unambiguously identify a feature type?
- * *Simplicity*; are properties (and parameters) only included if they are in use in some application?
- * *Uniqueness*; can a part be uniquely modelled using those features?

In the context of features, this final validation process has been relegated to a minor importance because: there is a close relationship between a feature's domain and its application, and therefore it is not easy to dissociate features themselves from their application semantics, and; the feature classification process has been organised in a way to emphasise application needs (see dotted arrow in Figure 2 showing the influence of an application over the definition of the classification and therefore, over the elicitation process)

3 The Complete Process

Figure 2 depicts the *entities elicitation* process. The feature classification and taxonomy are important products of this process. The following elements of the process need to be defined in order to obtain the resulting set of entities (e.g.: features, intents): *domain*, *elicitation criteria*, *classification*, *taxonomy*, *validation criteria* and *applications*.

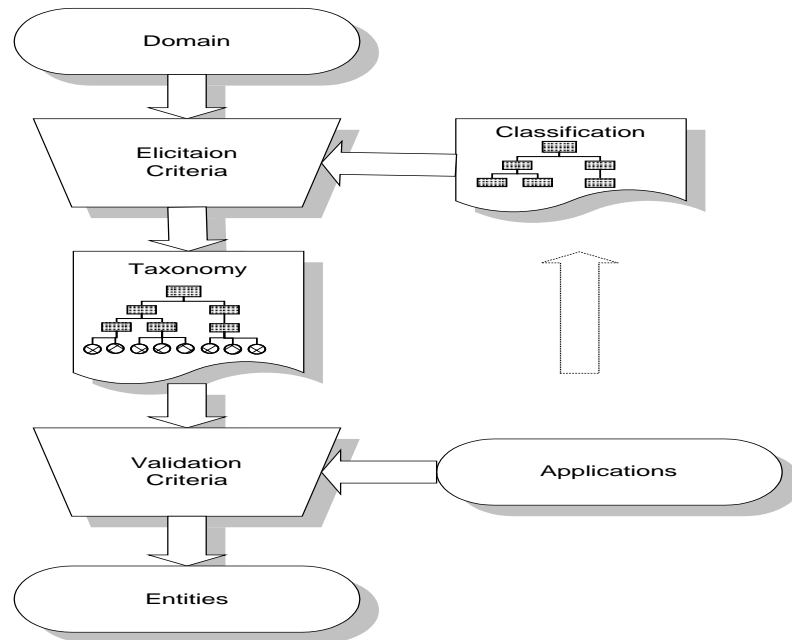


Figure 2: Entities Elicitation Process.

5. INTENTURIZATION

Although features are a proclaimed and accepted means of capturing and representing FbDI's, existing systems do not deal with FbDI's as their major concern.

The main reasons for this are threefold: firstly, there still is a lack of a formal well-accepted definition for features and their role as a geometric modelling technique; secondly, there is the same lack of understanding of what FbDI's are, especially in the context of FBM and; thirdly, identified intents are usually blended, immersed and diluted within the application under consideration.

Capturing FbDI's at early stages of the design in a more user-friendly interface that includes a vocabulary meaningful to the designer is a property of a design-by-features (**DbF**) system that allows more intelligent decisions and reasonings to be made and has been considered as "the only possible basis for Intelligent CAD" [1]. FbDI's are of "high importance to be preserved but their understanding has a complicated nature" [15]. The application of the *elicitation process* in the context of features produces the "featurization" and the "featurization

validation" processes and the "intenturization" and the "intenturization validation" processes, in the FbDI's context.

1 Designer's Intents Elicitation Criteria and Domain

A reasonable set of "manageable" FbDI's should be clearly identified and classified to match feature semantics and this is achieved via a suitable set of elicitation criteria. The *domain*

adopted for this research is the integration of feature-based CAD and CAPP and has been mainly gathered from related publications and therefore, FbDI's presented are the ones perceived from these systems.

Keeping a pragmatic awareness of the implications for DbF implementations, the following set of *elicitation criteria* were established to select objective, concrete and verifiable FbDI's:

- * They must have importance to the decision-making process of detailing a geometric design and hence, are *not* for *documentation* or illustrative purposes solely. In some systems, "designer's intents" can be found but they do not usually constitute a representation of the design knowledge to be used for subsequent verifications or to trigger reasoning.

- * They must have *geometric semantics* in a way that is suitable for association with features and for building a reasoning process.

- * They can be of a *hierarchical* nature, where high level abstract FbDI's can be defined but there are basic atomic FbDI's which are preferable.

- * Major attention should be paid to FbDI's that are *computable and inferred* during the design process rather than to those that can only be explicitly stated by the designer [1, 7, 10, 12, 146]. This does not mean that this process is easy or already available but does mean that it is conceivable.

- * FbDI's that build a hierarchy with tight dependency should be avoided or kept as a distinct class to maintain simplicity of the reasoning.

2 Designer's Intents Classification

A feature model is considered invalid if it does not meet its functions. Functions reflected into FbDI's can be characterised as Morphological, Parametric and Relational FbDI's. Therefore, three types of FbDI's have been identified (Figure 3):

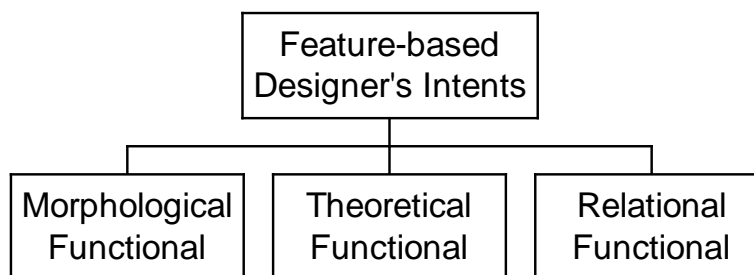


Figure 3: Feature-based Designer's Intents (FbDI's) Classification.

3 Designer's Intents Taxonomy

Each FbDI type has a set of objectives and a tangible set of properties at a "pragmatic" level, which helps to implement FbDI's within the geometric realm. FbDI types specify general engineering concepts or behaviours while the actual 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.

.1 Morphological Functional FbDI (MFI's)

Features represent a good means to embed functional significance into the geometric detailed design phase and this fact can be inferred by some definitions applied to features.

In addition to topological and geometrical analysis that is usually applied to identify features (as in Feature Recognition approaches), extra functional factors have been added to better specify 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* [8].

These considerations clearly expose features as having a *morphological function*. Volumetric Designer's Intents (**VDI's**) implement *morphological functional* FbDI's (**MFI's**) within the geometrical realm and are concerned with the feature's expected geometric behaviour FbDI (a detailed discussion on this can be found in [4]).

.2 Theoretical Functional FbDI (TDI's)

Features are also linked to the concept of *functions* themselves which is defined as "the behaviour of an object, an operation of energy, material, information or signal that tells what the design does" [13] and, "include not only in-use purpose, but also manufacturing and life-cycle considerations" [1].

Although some researchers have addressed the relationship between *form* and *function*, it is not formally understood yet because of many difficulties [10]: firstly, the abstract nature and understanding of the *function* concept; secondly, functionality can be a composite result of many interacting sub-functions; thirdly, a given *function* could be performed by several forms and one form could be used to perform 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 features theoretical functional behaviour and its form. This is possible by manipulating and controlling 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 thoroughly discussed in this text.

.3 Relational Functional FbDI's (RDI's)

While theoretical functional intents are usually expressed by formulae, other engineering constraints are expressed in the form of relationships between entities. Thus, they are called *relational functional* FbDI's.

Relational functional FbDI's (**RDI's**) comprise different disciplines and are dependent on the application of the feature-based model. RDI's are mostly 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 CAPP system to establish machining precedence among features

RDI's describe physical and/or spatial relationships between features and are divided into two categories:

- application-dependent but mostly geometry-dependent, called Geometric RDI's (**GDI's**).
- geometry-dependent but mostly application-oriented, called Application-Oriented RDI's (**AOI's**).

.1 Geometrical RDI's

Despite the fact that “it is almost impossible to pre-define all (geometric) feature relationships” [3], the importance of GDI's has been recognised by many systems that incorporate this spatial information in various ways [8, 12, 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. For instance, a hierarchical GDI is needed in order to define machining precedence but other geometrical reasonings such as “supporting walls” and “tool accessibility” must be considered as well.

.2 Application-Oriented RDI's (AOI's)

GDI's are defined in the detail geometric designer's domain but there are also “process planning engineer's intents” [14] as well as “manufacturing engineer's intents”, “production engineer's intents”, and so on. The intents from all these other application domains are called Application-Oriented RDI's (AOI's). 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.

AOI's exist to establish a more definite interpretation from the application's point-of-view. In contrast to GDI's, these intents consider information beyond geometrical relational facts. This extra information includes tool availability, process optimisation and, precedence constraints. Thus, different applications could interpret the same factual GDI's differently.

4 “Intenturization” Validation

The following are some of the designer's intents *validation criteria*. They help to identify a minimum set of FbDI's most suitable for an application. Because this process has been called “featurization validation” for features, it is called “intenturization validation” for FbDI's:

- * Selectable FbDI's can be conflicting, and hence care should be taken to select only *non-conflicting* FbDI's for a specific design application. In this way, reasonings will not interfere destructively with each other and loops will be avoided.

- * Because there are partially redundant intents such as those used to define abstract hierarchical FbDI's, atomic intents that have *non-overlapping* concepts/definition should be preferred. Thereafter, tricky situations with redundant FbDI's can also be avoided.

As the intents were mainly gathered from CAD/CAM and CAPP FBM systems, they are consequently all valid candidates for these applications. The intenturization validation process is relegated because no other domain was considered and because no specific application was considered. Therefore, this step of the elicitation process has not been fully applied.

5 The Complete Taxonomy

Figure 4 presents the resulting taxonomy of FbDI's. In addition to the Volumetric FbDI's detailed before [4], a myriad of other FbDI's were identified. This taxonomy is not intended to be complete but it highlights important categories and relationships that are found in previously mentioned feature-based systems.

Positional GDI's found include concentric, opposite, planar, coplanar and concentric intents between features. Orientational GDI's detected include parallel, perpendicular, angularity, against, collinearity and, coEAD (same External Access Direction) intents. Hierarchical GDI's identified were *nested_at_the_bottom* (*nested@bot*) and *nested_at_the_side* (*nested@side*). Structural GDI's perceived were patterns with Linear, Circular, Planar or Spatial distribution; radial, axial or mirror-like symmetry and, coradius intention.

Among application-oriented FbDI's noticed there are: same or different setup AOI's; parent-child and precede precedence intentional relationships; T-slot, Cross feature (x_feat), Enter feature (e_feat), counter-bore, counter-sink, and cut-out compound intentions obtained between features and; thin-wall proximity intentions.

6. DISCUSSION

A prototype system called **FRIEND**, short for **Feature-based validation Reasoning for Intent-driven ENgineering Design**, has been implemented and is capable of performing *morphological functional* and some *relational functional* FbDI validation.

Feature-based Designer's Intents were divided into three areas: related to individual features (MFI's); related to groups of features (Geometrical RDI's and TDI's), and; dependent on applications (Application-Oriented RDI's).

Although the use of a geometric constraining approach has already been introduced into FBM [2, 8] to either represent features as basic relational elements or to establish relationships between geometrical constituent elements of different features, their use is still based on conventional CAD's parametric or variational constraining approaches using mainly low-level geometric entities such as points, edges and faces. Thus, TDI's have not been detailed in the current FbDI's taxonomy.

7. CONCLUSIONS

The application of the *entity elicitation* process formalism (Figure 2), contributed to the detailing of the information required for an intent-driven validation system. As a product of applying the elicitation process, it has been possible to classify and give names to the various items. This produced a classification and a taxonomy of feature-based designer's intents (**FbDI's**, Figure 4). Classifications are important because they help group properties and highlight differences and, in particular, help emphasise the completeness of the subject and therefore, help identify the absence of elements.

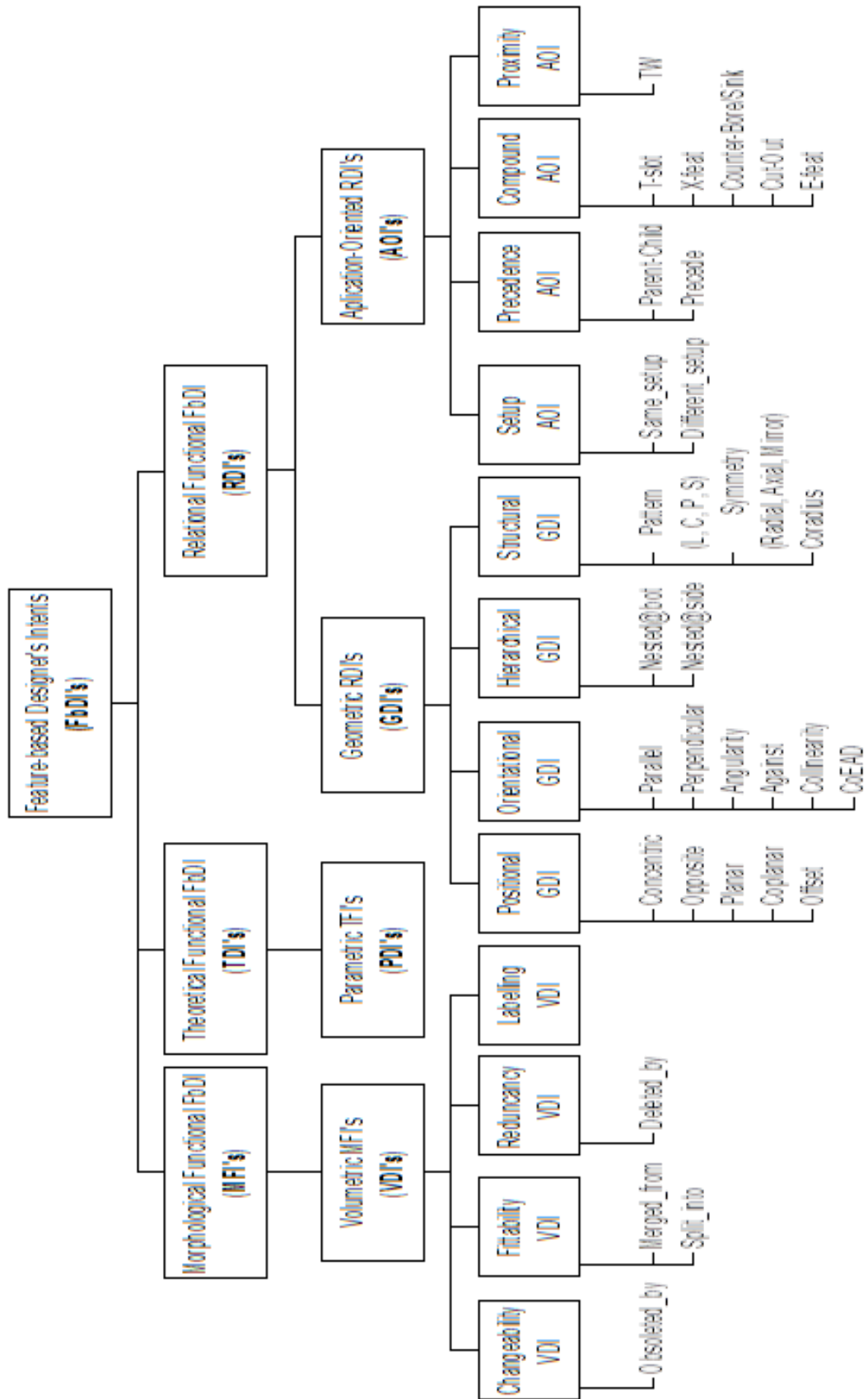


Figure 4: A Feature-based Designer's Intent Taxonomy.

It is expected that a FBM system driven by FbDI's will help to preserve the reasons for a particular decision in a design. For instance, the reason for a feature to be located at a specific position could be the achievement of an *axial symmetry* structural GDI. It can be beneficial to future FBM systems if featurization analysis and features taxonomy come accompanied by similar intenturization analysis and intents taxonomy to help define the implementation boundaries and capabilities of a particular feature-based application.

8. REFERENCES

1. Dixon, J. R., E. C. Libardi JR, and E. H. Nielsen. (1990) "Unresolved Research Issues in Development of Design-with-Features Systems". IFIP WG 5.2/ NSF Working Conf. on Geometric Modelling, RensselaerVille - USA, Elsevier Sc. Pub., Vol. 1, 183-196.
2. Dohmen, M. (1994) "Constraint Techniques in Interactive Feature Modeling". TUDelft - Delft University of Technology, Faculty of Technical Mathematics and Informatics.
3. Gindy, N.N.Z., X. Huang, and T.M. Ratchev. (1993) "Feature-based component model for computer-aided process planning". International Journal of Computer Integrated Manufacturing, Vol. 6(1-2), 20-26.
4. Hounsell, M. S. & K. Case. (1997) "Morphological and Volumetrical Feature-based Designer's Intents". NCMR'97 - 13th National Conference on Manufacturing Research (ISBN 1 9012 4811 9), Glasgow, Scotland, Vol. 1, 64-68.
5. Mantyla, M., D. Nau, and J. Shah. (1996) "Challenges in Feature-based Manufacturing Research". Communications of the ACM, Vol. 39(2), 77-85.
6. Mitchell, S.R. (1996) "A Feature-based Approach to the Computer Aided Design of Sculptured Products". Loughborough University.
7. Mill, F. G., J. C. Salmon, and A. G. Pedley. (1993) "Representation Problems in Feature-based Approaches to Design and Process Planning". International Journal of Computer Integrated Manufacturing, Vol. 6(1-2), 27-33.
8. Nielsen, E.H., J.R. Dixon, and E.E. Zinsmeister. (1991) "Capturing and using designer intent in a design-with-features system". DTM'91: 3rd. Int. Conf. on Design Theory and Methodology (ASME-DE), Miami, Florida, USA. Sept 22-25, Vol. 31, 95-102.
9. Pratt, M.J. and P.R. Wilson. (1985) "Requirements for support of form features in a solid modelling system". CAM-I Inc., Arlington, Texas, USA.
10. Salomons, O.W., F.J.A.M. van Houten, and H.J.J. Kals. (1993) "Review of Research in Feature-Based Design". Journal of Manufacturing Systems, Vol. 12(2), 113-132.
11. Shah, J.J. and M. Mantyla. (1995) "Parametric and Feature-Based CAD/CAM: Concepts, Techniques and Applications". John Wiley and Sons Inc.
12. Silva, R. E. d., K. L. Wood, and J. J. Bearman. (1990) "Representing and Manipulating Interacting Interfeature Relationships in Engineering Design for Manufacture". (ASME) Advances in Design Automation, Vol. DE-32-1, 1-8.
13. Tomiyama, T., Y. Umeda, and H. Yoshikawa. (1993) "A CAD for Functional Design". Annals of the CIRP, Vol. 42/1, 143-146.
14. Vancza, J. and A. Markus. (1993) "Features and the principle of locality in process planning". International Journal of Computer Integrated Manufacturing, Vol. 6(1-2), 126-136.
15. Yoshikawa, H. and K. Ando. (1987) "Intelligent CAD in Manufacturing". Annals of the CIRP, Vol. 36/1, 77-80.
16. Zhang, K.F. and H.A. Elmaraghy. (1993) "Validity Check for a Functional-oriented Modeler". (ASME) Advances in Design Automation, Vol. DE-65-2, 293-300.