Unified Feature Modeling Scheme for the Integration of CAD and CAx

G. Chen[1], Y.-S. Ma[2], G. Thimm[3] and S.-H. Tang[4]

[1]Nanyang Technological University, pg02198079@ntu.edu.sg
[2]Nanyang Technological University, mysma@ntu.edu.sg
[3]Nanyang Technological University, mgeorg@ntu.edu.sg
[4]Nanyang Technological University, pg02104852@ntu.edu.sg

ABSTRACT

In concurrent engineering, it is difficult to organize product information in an interconnected and consistent way due to complicated interrelations and proprietary data formats. This paper proposes an information representation scheme which accentuates feature association and feature unification. Feature association establishes persistent relations among different features constituents while feature unification provides a generic format for different application features. A unified feature defines common attributes and methods of all the supported application features. Feature relations are identified in application, feature and feature constituent levels for controlling the consistency among different application feature models.

Keywords: Concurrent engineering; Feature modeling; Information consistency control.

1. INTRODUCTION

The development of a mechanical product undergoes a sequence of life cycle phases including design, process planning, manufacturing, and so on. Different life cycle phases view the same product through different application models, such as design models, process-planning models, etc. Each model comprises different information entities relevant to the particular phase of product life cycle. However, different application models refer to the common final product geometry and product parameters. To improve product quality while reducing time and cost, the closure of the information gaps among different phases, i.e. managing product information in a coherent and consistent way, is a crucial and challenging research topic.

Ideally, a product information model for supporting concurrent engineering should fulfill representation, sharing and consistency requirements. Product information entities have to be represented precisely and completely; information originated in one application has to be accessible in other applications; and different application models must be consistent.

Used as information objects, features are suitable for supporting applications based on a product information model that satisfies the above modeling requirements. In knowledge-oriented systems, such as product design or generative process planning, features can be used as information elements to connect high-level knowledge-based systems and low-level product information models and to bridge the gaps among different applications [15]. Fig. 1 illustrates this concept.

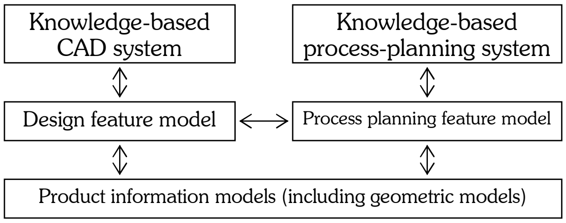

Fig. 1. Features as information elements

In this research, a feature is defined as a relationship object associating geometric entities. It has specific engineering meanings in a specific application [19]. Features can be used as:

(1) Carriers to represent product information,
(2) Building blocks to establish an application model in knowledge-level as well as geometric-level,
(3) Agents to execute validation and reasoning procedures.

This is in a stark contrast with current feature-based computer-aided systems, where four major problems exist:

(1) The lack of explicitly defined semantics for engineering features in their corresponding

applications. The feature ontology and feature relations are not well defined.

(2) Feature semantics, i.e. features' validation in the view of a specific application, are not well maintained during the product modeling process.

(3) Data interoperability between different CAx systems is difficult as they use different data formats [10]. Usually only low-level geometric information stored in the design models can be used for downstream applications directly [14], [7], which limits information sharing.

(4) The granularity of accessible information is different and makes across-domain information sharing and communication impossible.

In general, current computer-aided systems still can not effectively support consistent information flow through the entire product life cycle. To solve these problems, information sharing and consistency control among different CAx applications are necessary. In this paper, a unified feature modeling scheme is proposed as the first step to establish such an information infrastructure, in which different application features can be defined, communicated and used consistently. This unified feature-modeling scheme includes the definition of:

(1) Unified feature elements and their semantics;

(2) Generic geometric and non-geometric relation types in the unified feature model.

2. THE CONTEXT OF UNIFIED FEATURE MODELS

This research is a part of an encompassing research project, which explores web-enabled, feature-oriented database technology that supports scalable information modeling and sharing for concurrent engineering. This research project targets the definition and implementation of a multi-application oriented feature-modeling framework. It will provide a layer above the kernel for integrating existing CAx software packages. It will also lay down the foundation for a new generation of generic Application Service Provider (ASP) portal-based engineering services. Fig. 2 illustrates the overall architecture. In the figure, PIS represents product information service; ASP represents application service provider; PP Ap represents process-planning application; Design Ap represents design application; Assem Ap represents assembly application; PP KB represents process-planning knowledgebase; Design KB represents design knowledgebase; Assem KB represents assembly knowledgebase; PF represents process-planning feature; DF represents design feature; AssemF represents assembly feature; UF represents unified feature; AF represents associative feature; SM represents solid modeler.

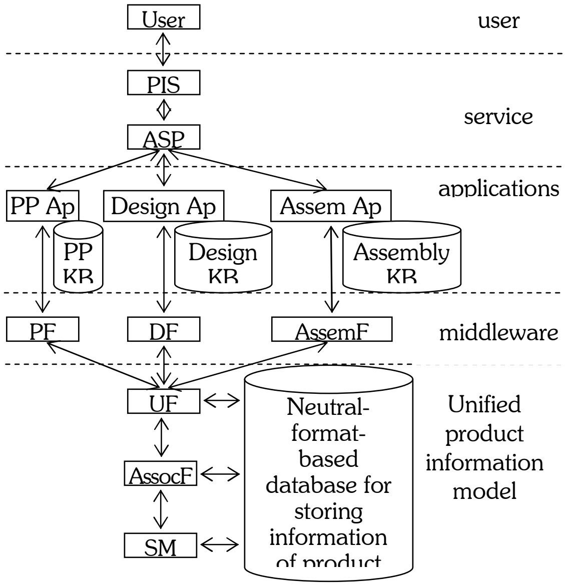

Fig. 2. Multi-application-oriented feature-modeling framework

3. RELATED WORK

Two approaches were explored in the development history of feature technology: design-by-feature and feature recognition. In the design-by-feature approach, predefined features are used to create the product model. Originally, it was expected that such features could be used by downstream applications directly. However, features are very application-specific. For example, design features cannot be used for process planning. At the same time, requiring designers to create product models using features defined for other applications is unreasonable because it limits a designer's creativity. Many researchers used feature recognition techniques to extract information from design models [11], [22], [9]. Due to feature interactions and multiple interpretations, a general and robust solution for feature recognition has not been found yet.

On the other hand, during the product modeling process, feature definitions may become inconsistent with the real product geometry due to feature interactions or inconsistent changes. Several solutions were proposed to solve this problem [12], [23], [14], [15]. Keeping feature models consistent with geometric models is crucial to connect knowledge-based reasoning and solid modeling process. However, besides feature geometric validation from the viewpoint of solid modeling, engineering validation, i.e. feature semantics checking is equally important and these consistencies should be checked and maintained in the context of the specific application.

To integrate different computer-aided applications, technologies to achieve a common information infrastructure were studied, such as in [4], [6], [10] where blackboard mechanisms were used. However, the mutual dependency relations among different application features have not been established yet, especially relations between non-geometric attributes of different features. They are necessary for information consistency control between different application models. However, they are also very complicated. For example, process planning on its own is a very complicated application; incrementally modifying process plans according to design changes is even more challenging. Such relations make across-domain consistency control very difficult.

Recently, some researchers concentrated on feature semantics [1-2], feature validity control [14-15] and feature relation maintenance [5], [13]. In addition, the ISO 10303-224 STEP standard [20] contributes in a great way for formally defining machining features. However, their machining feature definitions do not include (or provide mechanisms to link to) sufficient machining information, such as tolerances or machining methods, which are necessary for multi-application-oriented information. Furthermore, they do not provide implementation methods for defining new types of machining features.

4. FUNDAMENTALS OF UNIFIED FEATURE MODELING

In order to allow the description and manipulation of product information, explicit and concise definitions of information entities and relations among these entities are important. Another important issue is which information entities should be included in feature definitions. This section defines the basic concepts of unified feature modeling, which include definition of unified feature and generic relations. Constituents of two application features, design and process-planning features, are discussed.

Unified feature modeling scheme is an extension of the associative feature concept which was proposed for representing and manipulating complicated geometric relations with respect to mutual dependency [13]. These relations are difficult to represent by traditional feature technologies because traditional definitions are usually two-manifold and shell-based. The associative feature definition is not confined to the part boundary and emphasis on the geometric relations between entities within a single feature. It also highlights that an ideal data structure of a feature definition must be flexible and self-contained. In this paper, the associative feature concept is further extended to include non-geometric relations as well as associate entities of different features.

Another basic element of the unified feature modeling scheme is feature unification. It is proposed for providing a layer of generic definitions for different application features. The main characteristics of the proposed unified feature modeling scheme are listed as follows:

(1) Unified feature scheme defines the generic common characteristics (attributes and methods) of all supported application features;

(2) Unified feature attributes may be used to give feature specifications during the initialization of an application feature while unified constraint types can be used to establish relations between feature constituents for dynamic modifications;

(3) Constraint definitions in unified feature models provide an interface between feature geometric definitions and application's reasoning mechanism. These linkages ensure the validity of feature semantics from the viewpoint of a specific application;

(4) Due to the common information infrastructure, unified feature models can transcend application boundaries and therefore establish and maintain mutual dependency relations (geometric or non-geometric) between different application features through association types provided at the lower associative feature level;

(5) Unified feature models make use of associative geometric references to name, index, identify and query geometric elements supported with a solid model.

In general, the unified feature and associative feature concepts together provide a basis for establishing an information modeling scheme for collaborative and concurrent engineering. It supports an environment for consistency control among different application feature models.

4.1 Unified Feature Type

In accordance with the above-mentioned fundamental concepts, the unified features are modeled using a UML class diagram [3] (Fig. 3). In this figure, *GeoElement* represents geometric element; *PPFeature* represents process-planning feature while *OAFeature* represents other application feature.

For the reader's convenience, some UML symbols used in the above figures are explained here. Rectangles represent classes (such as the *UnifiedFeature* class), including class names, attributes and operations. Dashed and directed lines represent dependency relations. The lines are directed from the depending class to the class it depends on. Solid and directed lines with triangular, open arrowheads represent generalization relationships, pointing to the more general class defining basic properties. Solid and directed lines with open diamonds represent aggregation relationships, pointing from the "parts" to the "whole", aggregated object. The ranges aside the origin and target of an aggregation arrow indicate how many "parts" can or must be in a "whole" [3]. For example, a unified feature can include none or many other unified features.

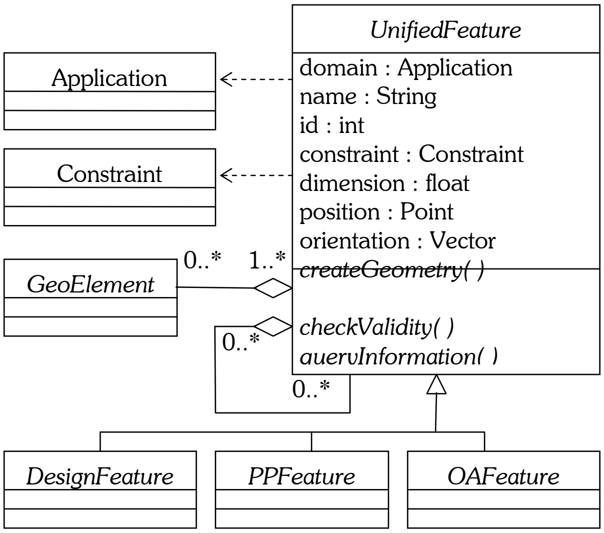

Fig. 3. Unified feature

In concept, the general unified feature type defines properties, which are inherited by all application features. As mentioned above, a feature is a relationship object associating a set of geometric entities, which have specific engineering meanings in a specific application.

(1) Constituents of a unified feature include geometric entities and attributes. Geometric entities may be primitive geometric elements, such as faces, edges, vertices; or other derived or constructive entities associated to the product and certain applications. Geometric entities may be part of several unified features. This allows the definition of combined features. Different application features' geometries have different natures and abstraction levels. For example, conceptual design feature's geometry may only include critical geometric entities which are indispensable to realize the required sub-functions. Detailed design feature's geometry include critical and other supplementary geometric entities while machining feature's geometry are derived from machining faces according to the chosen machining operations.

(2) Feature semantics are represented by persistent attributes and relations among different feature constituents.
   - Generic attributes can include *names*, *dimensions*, *positions* and *orientations*, etc. Specific application features can define their specific attributes.
   - Constraints specify relations among feature constituents. Constituent-level relations include relations defined on one or among a few primitive geometric elements, e.g. geometric constraints, such as radius, distance, parallel, incidence constraints; and those among inter-feature attributes and geometric elements. Relations among feature attributes and geometric elements are bidirectional. For example, changing the value of a feature's attributes can affect the corresponding geometric elements and vice versa. Specific application features can define and implement their specific constituent-level relations. Ontological relations can also be modeled to map semantic relations.

(3) Common methods include:
   - *createGeometry* method uses functions provided by solid modeler to create a feature's geometry;
   - *checkValidity* method checks the validity of constituent-level relations; In the case of combined features, constituent-level relations associate different features.
   - *queryInformation* method acquires object properties;

   Specific application feature classes need to materialize these methods.

4.2 Generic Relations

A generic relation *R (A, B, C,…),* where *A*, *B, C,…* are unified features, constrains geometric or non-geometric constituents of features *A, B, C,* etc.

According to the entities associated by the relations, generic relations can be defined on application, feature and feature constituent level. On different levels, different mechanisms or methods solve or enforce the constraints. Constraint relations in a unified feature model can be classified as geometric and non-geometric relations (Fig. 4).

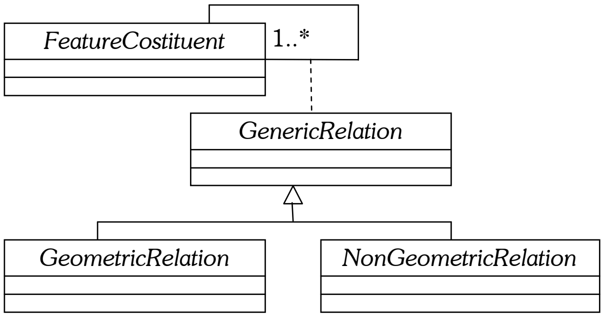

Fig. 4. Generic relation

Geometric relations are defined between two geometric elements. They include common geometric constraints, such as distance, parallel, radius, incidence, coaxial, and other geometric relations, such as those mentioned in [13]. It should be noted here that this research covers the definitions of geometric constraints specified in ISO 10303-108 [16]. Geometric elements involved in a geometric relation may belong to the same feature, different features from the same application, or even different features from different applications. Non-geometric relations may exist among:

(1) Feature attributes and the corresponding geometric entities; they are part of the consistency relations between feature models and geometric models.

(2) Feature attributes of the same feature or different features of the same application; these relations represent feature's engineering meanings.

(3) Feature attributes from different applications; these relations represent consistency relations and can be used to propagate modifications among associated application feature models.

A brief analysis of relations in design and process-planning stages are given in the next two sections. Major information entities and relations among them are discussed. These identified entities and relations can be used for further developing design feature models and process-planning feature models.

4.2.1 Design relations

A product design process can be roughly divided into conceptual design stage and detailed design stage. During the conceptual design stage, a designer translates given design requirements (from market or customers) into product functions. Following function decomposition, suitable behaviors (usually with multiple possibilities) and the corresponding supporting geometric structures (assembly or part) are generated. Generally, function-behavior-geometry correspondences are not one-to-one, i.e. many possible geometric solutions for a given function exist. In the detailed design stage, the chosen geometric structures are transformed into real geometries. Similarly, multiple possibilities exist in this mapping process. Detailed product parameters, such as dimensions, tolerances, surface finishes or materials, are specified in this stage. Some intermediate or more determinant linkages are proposed to control the mapping [17], [8], [18]. These approaches depend on predefined structure libraries to limit the search space. Reasoning among different solutions is still an unsolved issue. However, no matter interactively or manually, once the detailed product geometries and parameters are generated, the relations between product's geometries, parameters and its corresponding functions should be established and maintained for later consistency control.

Each part in a product assembly has its geometry and characteristics (shape, size, etc.). Product behaviors are generated from part-part interactions which may be geometric or physical interactions. It is these static geometric structures and dynamic behaviors that realize the product functions. On the other hand, product functions determine the necessary behaviors, i.e. interactions. Part-part interactions further require the existence of critical geometric entities and their corresponding attributes. These critical geometric entities/attributes as well as relations (including non-geometric relations) between them should be kept in the detailed design model.

Three types of design relations are generalized:

(1) Function/behavior relations. Geometric structures' behaviors collectively determine the product functionality.

(2) Behavior/geometric structure relations. Geometric structures and their interactions can be determined by behavior requirements. These structures include part geometries, dimensions, tolerances, surface finishes, materials, etc.

(3) Geometric structure/geometric structure relations. Reference relations and assembly relations among different feature constituents can be derived from the required interactions.

Generally, in the existing feature modeling systems, features are only used as geometric macros. Feature parameters are usually not checked and maintained from the viewpoint of their engineering validity. In this proposed unified feature modeling scheme, besides geometric relations, non-geometric relations should also be embedded into feature definitions. These two kinds of relations are associated to functional reasoning processes, i.e. function-behavior-geometric structure mapping processes.

4.2.2 Process planning relations

Feature-based process planning, generally includes tasks of macro machining feature generation (design surface-machining operations mapping), machining operation aggregation, sequencing (micro machining feature generation) and machining parameters determinations. A predefined and workshop-specific machining feature library is necessary.

In different machining environments, a design model can be interpreted as different macro machining features. Furthermore, in a specific machining environment, according to different priorities (machining time, cost or product quality), a design model may correspond to different process plans, i.e. different micro machining features [21].

A macro machining feature represents a stock removal set [21] which in turn consists of a set of micro machining features. A micro machining feature is generally defined as the surface generated by a primitive machining operation. Its attributes include surface type, operation tolerance, cutting depth, datum face, pre-operation face and post-operation face. Major relations in process-planning include:

(1) The dimension, tolerance and surface finish of the last workpiece face, i.e. the post-operation face of

the last micro machining operation, should fulfill the corresponding design face specifications.

(2) The sum of cutting depths of all micro machining features of a macro machining feature should be equal to the overall allowance.

(3) The post-operation face of the previous micro machining operation is the same as the pre-operation face of the current micro machining operation.

(4) A micro machining operation's cutting depth is determined by the cutter specifications of the current micro machining operation and those of the next micro machining operation.

(5) Precedence constraints among machining operations, e.g. finishing operations can not be arranged before roughing operations.

Traditionally, design by machining feature systems only use macro machining features to create product's geometric model. In this way, only limited information can be represented from the viewpoint of process-planning. Micro machining features are more suitable to link design and process-planning. The parameters of a micro machining feature and the relations between different machining features are manipulated by higher-level process-planning reasoning processes, i.e. design surface-machining operations mapping, machining operations aggregation and sequencing.

4.3 Unified Feature Model

A product information model consists of several application feature models. Each application feature model consists of a set of application features, which are instances of subclasses of unified feature. Therefore, a product information model may also be regarded as the combination of a set of feature constituents (geometric and non-geometric) and relations specified among them. All these different application feature models refer to the same final product geometry and product parameters. From the above analysis, we can see final product geometry and product parameters are the common basis for linking design and process-planning. The final product geometry and product parameters are also the pivot to trade off the conflicted design and manufacturing requirements.

According to the elements involved, relations can be classified on application-level, feature-level and feature-constituent-level. A product information model is responsible for maintaining application-level relations, i.e. keeping consistency relations or trading off between different applications. Each application is responsible for maintaining corresponding feature-level relations, i.e. keeping consistency relations between different features within a specific application. Each application feature is responsible for maintaining feature-constituent-level relations, i.e. keeping validity of an application feature.

A product information model is regarded as a valid unified feature model (Fig. 5), if:

(1) All application feature models are created using the unified feature concept,

(2) Three-levels (application, feature and feature-constituent levels) of geometric and non-geometric mutual dependency relations are established and maintained.

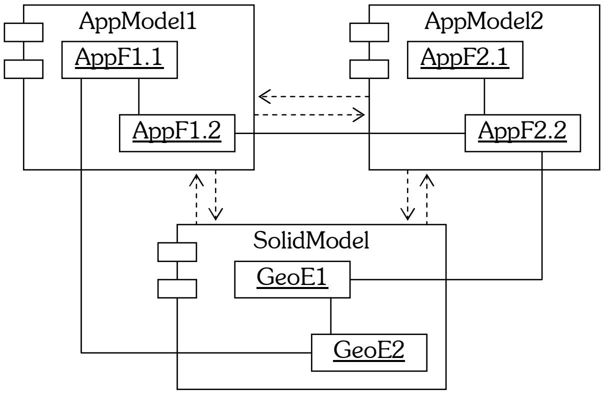

Fig. 5. Unified feature model

A rectangle with two tabs in Fig. 5 represents a component (such as application model *AppModel1* or solid model *SolidModel*), which is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Rectangles with their names underlined (such as geometric element *GeoE1*) represent objects to which a set of operations can be applied and which has a state that stores the effects of the operations [3]. In the figure, *AppModel* means application model. *AppF* means application feature and *GeoE* means geometric element. Note that the requirements from different applications in a real product information model may conflict with each other. A conflict resolution mechanism for all supported applications is necessary in such cases to solve the conflicts and keep the consistency relations in the product information model. This problem can be solved eventually if a common unified feature model is established and accepted by the engineering community.

5. CONCLUSION AND FUTURE WORK

This paper proposes a new unified feature-modeling scheme for information sharing and consistency control among different application feature models. This scheme is based on feature association and unification concepts, covering three-level geometric and non-geometric relations. Such relations are established and dynamically maintained in the proposed unified feature model. This scheme is intended as the foundation for a future multi-application and feature-object-oriented information infrastructure for collaborative and concurrent engineering.

6. REFERENCES

[1] Au, C. K. and Yuen, M. M. F., A semantic feature language for sculptured object modeling, *Computer-Aided Design*, Vol. 32, No. 1, 2000, pp 63-74.

[2] Bidarra, R. and Bronsvoort, W. F., Semantic feature modeling, *Computer-Aided Design*, Vol. 32, No. 3, 2000, pp 201-225.

[3] Booch, G., Rumbaugh, J. and Jacobson, I., *The Unified Modeling Language User Guide*, Addison Wesley, 1999.

[4] Bronsvoort, W. F., Bidarra, R. and Noort, A., Semantic and multiple-view feature modeling: towards more meaningful product modeling, in *Geometric Modeling: Theoretical and Computational Basis towards Advanced CAD Applications*, ed. Kimura, F., Kluwer Academic Publishers, Dordrecht, 2001, pp 69-84.

[5] Brunetti, G. and Golob, B., A feature-based approach towards an integrated product model including conceptual design information, *Computer-Aided Design*, Vol. 32, No. 14, 2000, pp 877-887.

[6] De Martino, T., Falcidieno, B. and Habinger, S., Design and engineering process integration through a multiple view intermediate modeler in a distributed object-oriented system environment, *Computer-Aided Design*, Vol. 30, No. 6, 1998, pp 437-452.

[7] Dereli, T. and Filiz, H., A note on the use of STEP for interfacing design to process planning, *Computer-Aided Design*, Vol. 34, No. 14, 2002, pp 1075-1085.

[8] Gorti, S. R. and Sriram, R., From symbol to form: a framework for conceptual design, *Computer-Aided Design*, Vol. 28, No. 11, 1996, pp 853-870.

[9] Gupta, S. K., *Automated manufacturability analysis of machined parts*, Ph.D. Thesis, The University of Maryland, College Park, MD, USA, 1994.

[10] Hoffman, C. M. and Joan-Arinyo, R., CAD and the product master model, *Computer-Aided Design*, Vol. 30, No. 11, 1998, pp 905-918.

[11] Joshi, S. and Chang, T.C., Graph-based heuristics for recognition of machined features from a 3D solid model, *Computer-Aided Design*, Vol. 20, No. 2, 1988, pp 58-66.

[12] Laakko, T. and Mantyla, M., Incremental feature recognition, in *Advances in Feature Based Manufacturing*, Manufacturing Research and Technology, ed. Shah, J. J., Mantyla, M. and Nau, D. S., Elsevier Science, 1994, pp 455-479.

[13] Ma, Y. -S. and Tong, T., Associative feature modeling for concurrent engineering integration, *Computers in Industry*, Vol. 51, No. 1, 2003, pp 51-71.

[14] Mandorli, F., Cugini, U., Otto, H. E. and Kimura, F., Modeling with self validation features, in *Proceedings of ACM/IEEE Symposium on Solid Modeling and Applications '97*, 1997, pp 88-96.

[15] Otto, H. E., From concepts to consistent object specifications: translation of a domain-oriented feature framework into practice, *Journal of computer science & technology*, Vol. 16, No. 3, 2001, pp 208-230.

[16] Pratt, M. J., Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models, ISO TC184/SC4/WG12 N940 (ISO/CD 10303-108), 2001-08-01.

[17] Ranta, M., Mantyla, M., Umeda, Y. and Tomiyam, T., Integration of functional and feature-based product modeling – the IMS/GNOSIS experience, *Computer-Aided Design*, Vol. 28, No. 5, 1996, pp 371-381.

[18] Roy, U. and Bharadwaj, B., Design with part behavior model, representation and applications, *Computer-Aided Design*, Vol. 34, No. 9, 2002, pp 613-636.

[19] Shah, J. J. and Mantyla, M., *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*, John Wiley & Sons, Inc., 1995.

[20] Slovensky, L., Product data representation and exchange: Application protocol: Mechanical product definition for process planning using machining features, ISO TC184/SC4/WG3 N988 (ISO/IS 10303-224.2), 2000-12-01.

[21] Thimm, G., Britton, G. A. and Fok, S. C., A graph theoretic approach linking design dimensioning and process planning Part 1: design to process planning, *International Journal of Advanced Manufacturing Technology,* 2004, to appear.

[22] Vandenbrande, J. H. and Requicha, A. A. G., Spatial reasoning for the automatic recognition of machinable features in solid models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 12, 1993, pp 1269-1285.

[23] Venkataraman, S., Shah, J. J. and Summers, J. D., An investigation of integrating design by features and feature recognition, in *IFIP Conference, FEATS 2001*, Valenciences, France, 2001.