Using Cellular Topology in a Unified Feature Modeling Scheme

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

Different computer-aided systems, which are used in specific product lifecycle stages, have different requirements on product geometry representation. Non-manifold and multi-dimensional geometries are also needed besides two-manifold solid models. Traditional geometric modeling systems, which usually use B-rep or CSG solid models, have limitations to accommodate these diversified requirements. In addition, in an integrated product development environment, to share data as well as to propagate changes across applications, a unified feature modeling scheme, which can support different geometric modeling requirements uniformly, is preferred. In this paper, a unified, cellular topology based feature modeling scheme is proposed. Its model structure and usage in integrating application models are described.

Keywords: Unified feature modeling; Cellular topology; Product lifecycle management

1. INTRODUCTION

Historically, computer-aided applications, such as CAD, CAPP and CAM systems (called "CAx systems" hereafter) are developed to support the corresponding product lifecycle stages. Because these stages are inter-related and mutually constraining each other, to maintain the consistency of the whole product information model, their corresponding application models must be managed coherently. A unified feature modeling scheme has been proposed to integrate the conceptual design, detail design and process planning applications [1]. It is characterized by a unified feature definition, data association mechanisms as well as the algorithms for propagating modifications. Two major extensions to the traditional feature technology are the embedded knowledge-based reasoning [2] and the unified, cellular topology based feature modeling mechanisms.

Traditional geometric modeling systems use boundary representation (B-rep) or constructive solid geometry (CSG) models for geometry representation. They have the following limitations with respect to the requirements of the unified feature modeling scheme:

(1) Only the final product geometry is stored and managed. Intermediate geometries, which do not belong to the final boundary, are usually not stored. This limitation makes feature modifications difficult [3]. It also results in a persistent naming problem [4] [5].
(2) CAx systems have different requirements on representing the product geometry.
(3) CAx systems need to represent the same product geometry in different ways. On the one hand, geometry may be represented in different abstraction levels [6]. For instance, a hole can be represented as a central line (plus a radius), a cylindrical face or a cylinder in different contexts [7]. On the other hand, product geometry may be represented in different ways [8]. For instance, two adjacent faces in one application might be represented as a single face in another application. In addition, it is important for the unified feature modeling scheme that higher level application features can use lower level topological entities to propagate changes and control the information consistency. Relationships or constraints in higher levels (*e.g.* feature level) may also be specified using lower level (*e.g.* topological entity level) relations.

A hybrid geometric modeling environment that can accommodate the associative wireframe, surface and solid models coherently is a natural outcome of the unified feature modeling scheme. Several researchers have proposed to use the multi-dimensional, non-manifold topology, especially the cellular topology, to solve these problems. Following the pioneer work of Weiler [9] [10], in which the radial edge structure was proposed to represent non-manifold geometries, Rossignac identified the potential usage of the cellular topology in representing objects of mixed dimensionality as well as with internal structures [11]. Crocker and Reinke pointed out that the major difference between a two-manifold and a non-manifold boundary evaluation lies in the execution order of topological construction and classification [12]. Two-manifold boundary evaluation process classifies the original geometries as in, out or on the boundary in the first place. "Useless" geometries (according to the types of Boolean operators) are then discarded before the part topology is reconstructed. Non-manifold boundary evaluation process reverses the order of these two steps. All original geometry are kept (but marked according to the types of Boolean operators) even they do not belong to the final real boundary. Masuda [13] proposed a mathematical framework with extended Euler operators for non-manifold geometric modeling. Sriram *et al.* developed a non-manifold geometric engine, which provides a unified representation for different application geometric modeling requirements [14]. Lee [6] and Deng *et al.* [15] suggested using multi-dimensional, non-manifold geometric model to integrate CAD and CAE applications because CAE analysis usually requires a higher abstraction level than the detailed product geometry. The idea of using two sub-models for CAD and CAE operations, and maintaining the consistency of these two sub-models via a CAD/CAE integrated model, was proposed. The above-mentioned research does not fully apply the multi-dimensional, non-manifold topology to the feature-based modeling processes. In particular, in a multiple-view feature-based modeling environment, an integrated geometry model is necessary for data sharing and change propagation. Bidarra *et al.* proposed using the cellular model to address this issue [3] [16]. Cells' nature consistency and "link" constraints (*i.e.* shared cell faces) are proposed to keep the consistency among views; but their research scope are confined to 3D features only. Some researchers have suggested other potential applications of the cellular topology. For example, Wu and Sarma proposed a dynamic segmentation and incremental editing methods to support collaborative design [17]. The cellular topology is used to localize the shape change in an editing process. Only the cells that embody shape changes, not the whole B-rep, are transferred between collaborators. Similarly, Lee *et al.* proposed progressively streaming and transmitting solid models over the network [18]. Cellular topology has also been used for more efficient machining feature recognition [19].

It can be concluded that the multi-dimensional cellular topology provides the flexibility to tackle the integration or collaboration problems in product lifecycle modeling. However, how to propagate geometric modifications, and in turn, how to maintain geometric consistency among feature models are still issues to be solved. In this paper, a unified cellular model is proposed to support the multiple-view feature-based modeling processes. The traditional usage of the cellular topology in multiple-view feature modeling is extended in three aspects:

(1) 2D and 3D features are supported uniformly;
(2) The unified cellular model is used to share geometric data as well as to propagate geometric modifications (creating or deleting cells) among views through the cells' owning feature attributes;
(3) Relations in the cell level are generalized. These relations can be used as building blocks to establish higher-level feature relations.

This paper is organized as follows: the next section introduces the unified feature modeling scheme briefly to provide the context. The current application of cellular topology in feature-based modeling is described in Section 3. The multi-dimensional cellular topology as well as its relations with the unified feature modeling scheme is given in Section 4. In Section 5, generic relations in the cell level are identified. To illustrate the concept, two cases are given in Section 6.

2. UNIFIED FEATURE MODELING SCHEME

The unified feature concept is derived from the associative feature concept [7] [20]. Currently, the unified feature modeling scheme covers three stages: conceptual design, detail design and process planning. Two aspects of the product lifecycle are accentuated:

(1) The commonalities among the information models for different stages are identified. For example, they are all related to the product geometry, and they all have embedded semantics. These generic characteristics and methods are represented as *unified features*.
(2) These stages are associated to each other, *i.e.* inter-related and mutually constrained. These associations are maintained by the data association mechanisms.

The main purpose of the unified feature modeling scheme is to keep the validity, consistency and integrity of product models. It includes three major modules, *i.e.* an embedded expert system, a unified cellular model (which is the focus of this paper) and a change propagation mechanism.

3. THREE-DIMENSION CELLULAR TOPOLOGY USED IN FEATURE-BASED MODELING PROCESS

Fig. 1 illustrates a feature model established on the traditional two-manifold boundary representation. Because the intermediate geometries are not kept after Boolean operations, it is not easy to keep the relations between the feature and its corresponding topological entities in the final boundary representation. In the constraint-based, parametric design processes, this limitation makes the feature model history-based. It is also the major reason for the persistent naming problem.

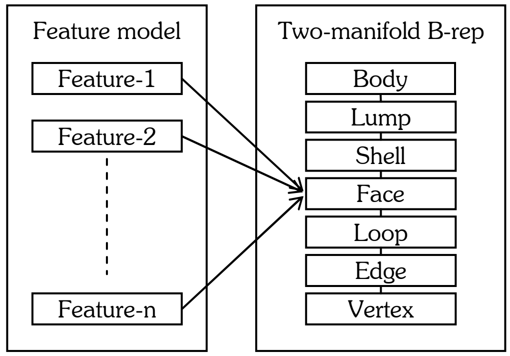


Fig. 1. Feature model based on the traditional two-manifold boundary representation.

As a solution, the cellular topology has been used to support feature-based modeling. The goal of using the cellular topology is to keep a complete description of all the input geometries without removing them after the Boolean operations regardless whether they belong to the final part boundary or not. Fig. 2 describes a feature model established on the cellular topology. Further more, the cellular model is established on the basis of non-manifold boundary representation.

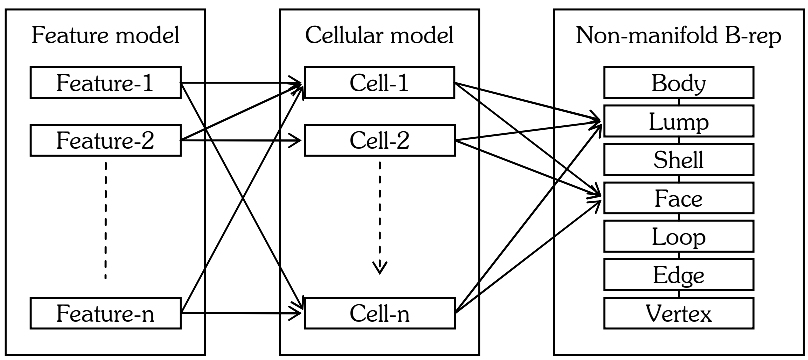


Fig. 2. Feature model, cellular model and non-manifold boundary representation.

The cellular model uses three mechanisms to fulfill this goal:

(1) *Attribute mechanism*. There are two kinds of attributes used in a cellular model:
   - *Cell nature*. A 3D cell is either additive or negative depending on whether it represents material of the product or not [16].
   - *Owner*. Each cell records its owing features because a cell may belong to several features due to feature interactions. The sequence of the owning features is kept to determine the cell nature.

(2) *Decomposition mechanism*. Two 3D cells do not overlap volumetrically. Whenever two cells overlap, new cells are generated to represent the overlap with the merged owning feature list.

(3) *Topology construction mechanism*. In a cellular topology based, non-manifold boundary representation, an operation on volumes does not remove any input geometry. The cellular model constructs topology (generates new faces, edges and vertexes) before classifying the topological entities as *in*, *out* or *on the boundary*. All topological entities are marked and filtered for displaying according the type of the operation.

The cellular topology enables representing features as combinations of cells. The canonical definitions of features are persistently linked to the non-manifold boundary representation through their corresponding cells. In other words, all feature geometries, geometric relations between features as well as relations between a feature and its corresponding topological entities are kept in the product model. By using this approach, the traditional persistent naming problems can be avoided. It is also possible to modify a feature based on the dependency relations, not on the construction history, because the influence scope of a geometric modification can be localized by the inter-cell relations.

As mentioned by Weiler [9], non-manifold geometric modeling encompasses both the manifold and non-manifold domain. It also allows the unified representation of wireframe, surface, and solid modeling forms. It has been observed that the decomposition mechanism causes fragmentations of the cellular model whenever feature interactions occur during an editing operation [17]. The efficiency issues brought by fragmentations are discussed in [12] and [21]. They concluded that boundary evaluation for the cellular model is more efficient than boundary evaluation for a B-rep. The main reason is that the former is dependency-based and hence is proportional to the number of intersections with the edited entity while the latter is history-based and hence depends on the position of the edited entity in the product construction history as well as the model complexity.

4. UNIFIED CELLULAR MODEL

4.1 The Extended Use of Cellular Model

In real industry, distinct applications covered by the unified feature modeling scheme have their particular geometry representation requirements:

(1) During conceptual design, a designer is concerned about functions and behaviors. Only critical geometries and their relations are specified. These critical geometries may only be represented as abstracted lines, faces, curves or surfaces. Solid models, detailed topologies, and geometries are not specified in this stage.

(2) In the detail design stage, the product geometries or layouts are further materialized. Two-manifold solid model representation is usually preferred.

(3) In the process planning stage, features are usually defined as material removal or accessing volumes related to machining operations. Fixtures are also conceptualized in this stage. For theses type of features, solid representation with surface manipulation support is more appropriate because, other than the machined volumes, fixture design uses sub-area patches of the part, *e.g.* locating, clamping areas, *etc.*

(4) Similar requirements are applicable to the assembly design stage. In particular, the sub-areas of the part or assembly for interfacing or grasping are concerned.

The geometrical representations discussed above relate to each other. They represent different aspects or abstraction levels of a product. To meet these diversified geometry representation requirements, the current cellular topology based feature modeling needs to be further extended to support not only 3D solid features, but also non-solid features. A multi-dimensional cellular model, named as *unified cellular model*, is proposed here to integrate all these representations, manage their relations and hence support the multiple-view feature-based modeling processes. Such unification is demanded by concurrent and collaborative engineering. The geometry model of each application is a particular aspect (a sub-model) of the unified cellular model.

4.2 The Characteristics of the Unified Cellular Model

A unified cellular model (*UCM*) includes all geometries from different applications.

(1) It consists of a set of unified cells (*UCs)*:

$$UCM = \left(\bigcup_{i=1}^{q} UC_i^0\right) \cup \left(\bigcup_{j=1}^{r} UC_j^1\right) \cup \left(\bigcup_{k=1}^{s} UC_k^2\right) \cup \left(\bigcup_{l=1}^{t} UC_l^3\right), \quad (1)$$

in which UC^0, UC^1, UC^2 and UC^3 represent zero-dimension (0D) vertices, one-dimension (1D) edges, two-dimension (2D) faces and three-dimension (3D) solids, respectively. Similarly, *q*, *r*, *s* and *t* are the numbers of 0D, 1D, 2D and 3D cells, respectively, in the unified cellular model.

(2) Each cell (except 0D cells) is bounded by a set of cells of a dimensionality lowered by one. On the other hand, a cell may exist independently without bounding any higher-dimensional cell.

(3) Any two cells, regardless of the same or different dimensionalities, do not overlap:

$$UC_i^a \cap UC_j^b = \phi \;\; (0 \le a < b \le 3, \text{ or } (a = b) \wedge (i \ne j)). \quad (2)$$

In addition, a cell does not include its boundary, except for 0D cells.

(4) The cellular model obeys the Euler-Poincare formula for non-manifold geometric models [13]:

$$v - e + (f - r) - (V - V_h + V_c) = C - C_h + C_c , \quad (3)$$

where v, e, f, r, V, V_h, V_c, C, C_h and C_c are the numbers of vertices, edges, faces, rings, volumes, holes through volumes, cavities in volumes, cellular complexes, holes through cellular complexes and cavities in cellular complexes, respectively.

Each application feature model is derived from the unified feature models. The relations among these models are described as follows (see Fig. 3, arrows in the figure represent "map" or "consist" relations):

(1) An application feature model (*AFM)* consists of a set of application features (*AFs)* and non-geometric entities (*NGEs)*:

$$AFM = \bigcup_{i=1}^{m} AF_i \cup \bigcup_{j=1}^{n} NGE_j \,, \tag{4}$$

where m and n are numbers of application features and non-geometric entities in the application feature model.

(2) An application cellular model (*ACM)* is created at runtime, which consists of a set of application cells (*ACs)*:

$$ACM = \bigcup_{i=1}^{u} AC_i \,, \tag{5}$$

where u is the number of application cells in this application cellular model.

(3) Each application feature refers to a set of application cells. An application cell may belong to several application features, *i.e.* it records several features in its owning feature list. The geometries of an application feature correspond to 1D, 2D or 3D cells.

(4) The cells of an application feature, after being inserted into the unified cellular model, could be further split by other applications. Hence, an application cell can be mapped to one or more cells in the unified cellular model. On the one hand, for a particular application, one related cell in the unified cellular model is mapped to only one of its ACs because each AC is unique in the application. On the other hand, each cell in the unified cellular model is mapped to at least one AC (and therefore at least one AF). This mapping is realized through the owning feature attribute mechanism.

(5) The rule for determining cell nature applies to the unified cellular model, *i.e.* the nature of the latest feature in the owner list determines the nature of the cell.

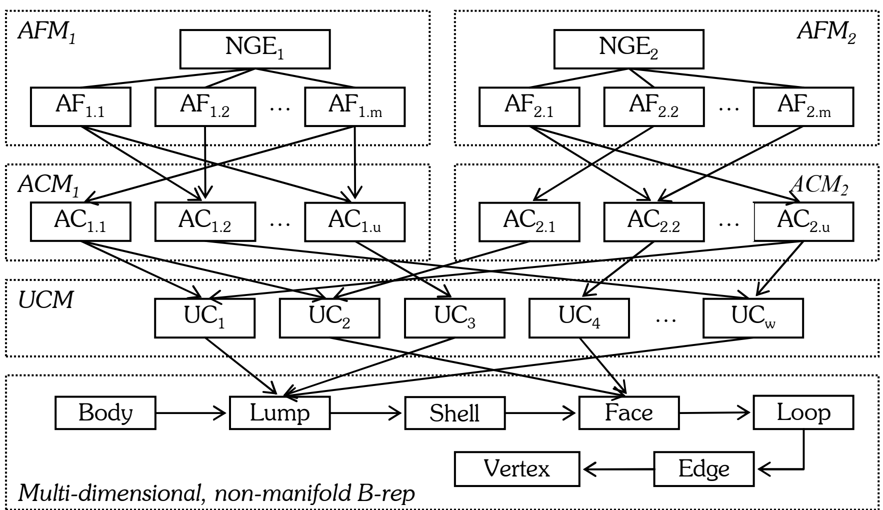


Fig. 3. Hierarchical structure of the unified cellular model.

As shown in Fig. 3, all applications use this unified, multi-dimensional cellular model. The geometry of each application feature model is one particular aspect of the unified cellular model.

4.3 Two-Dimension Features and Their Characteristics

The idea of this work is that unified cellular modeling scheme represents 1D, 2D and 3D cells uniformly (they are referred to as edge, face and solid cells respectively hereafter). Currently, the prototype system only handles face and solid cells (corresponding to 2D and 3D features respectively). 1D features are mentioned here but more research will be done in the future. Examples for 2D application features in the current implementation are:

(1) Conceptual design features, which represent functional areas in the product. The geometries of these features are usually abstracted as pairs of interacting faces. These faces are mapped to the entity faces in the detail design.
(2) Assembly features, which represent grasping or mating areas of a part in assembly processes.
(3) Locating or clamping features, which represent locating or clamping faces during a machining operation.

Similar to solid cells, the major advantage of using face cells instead of geometric faces is that operations on face cells do not remove faces (or parts of faces) even they do not belong to the final boundary. For example, the non-regular operations on surfaces and decomposition mechanism upon overlapping detection are available to face cells. A 2D feature is represented as a group of associated face cells with engineering semantics. For example, a locating feature is defined as a pair of faces associated with the constraints on accessibility, machining accuracy, non-interference and minimizing setup changes. The geometry of a 1D feature comprises one or more curves, while the geometry of a 2D feature comprises one or more surfaces. Two characteristics of 2D features are:

(1) A 2D feature has a nature attribute (additive or negative) that can be changed by feature interactions. A change of cell nature (from additive to negative or vice versa) requires the corresponding features to be validated. For example, a clamping feature represents a local area on a part that is used for clamping. When a clamping feature is altered, its face cells may be split and the natures of some of the resulting face cells inverted. This may jeopardize the clamping feature's stability (sufficient area for clamping). Similar situations are encountered for functional, assembly and locating features.
(2) Face cells corresponding to functional, assembly, locating and clamping features have the same surface definitions as existing face cell(s). Hence, to simplify the implementation, it is assumed that newly inserted face cells and existing solid cells do not intersect. However, this is not valid for some CAE analysis applications, in which middle faces are commonly used.

When a 2D feature is generated, the corresponding face cell is also generated and inserted into the application and the unified cellular models.

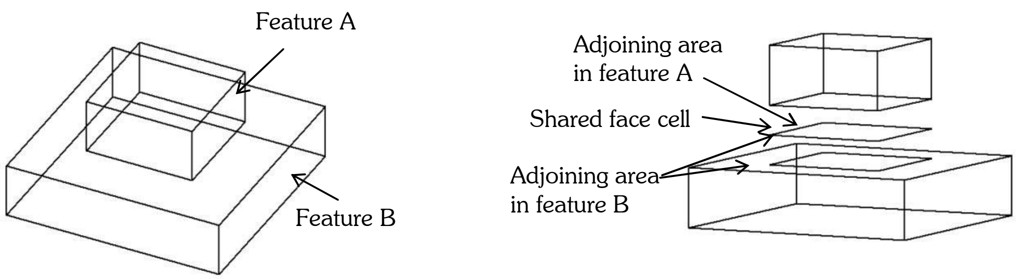
5. RELATION HEIRARCHY IN THE UNIFIED CELLULAR MODEL

Relations can be established on the cell level, the feature geometry level and the feature semantic level respectively. Higher level relations are established on the basis of lower level relations. This relation hierarchy is as follows:

(1) The lowest level of relations is between two cells, which covers the following cases:
   - The bounding relations among cells.
   - The bounding cells inherit the owner attributes of the bounded cell.
   - Two 3D cells are adjacent if they are bounded by one or more common 2D cells. Two 2D cells are adjacent if they are bounded by one or more common 1D cells.
   - Two adjacent edge or face cells may be part of the same curve or surface.
(2) Second level relations are topological relations between the geometries of application features. Note that a feature's dimensionality can be diversified depending on the application nature. Some topological relations between two application features are identified as follows:
   - After cellular splitting, two n-dimensional features are said to be *overlap* with each other if they use same n-dimensional cell(s). A n- and a (n-1)-dimensional features are also said to be *overlap* with each other if they use the same (n-1)-dimensional cell(s);
   - Two different n-dimension features are defined as *adjacent* ones if they share (n-1)-dimensional cell(s) but do not overlap;
   - In a 3D feature, *adjoining area* refers to one or more faces (represented by the 2D cells), which are mathematically connected and defined on the same surface. For two 3D features A and B, feature A is said to be *completely adjacent* to feature B, if a feature A's adjoining area is fully enclosed by any of feature B's adjoining area. Fig. 4 illustrates the complete adjacency of two 3D features. Other examples are:
      a. A single face in the detail design often corresponds to several functional faces in the conceptual design.
      b. A face in the process planning model corresponds to one or more faces in the detail design.
      c. In plastic injection mold design, completely adjacent relations can be used to represent maps from the plastic part to core or cavity inserts as well as electrode geometry. Such maps are commonly encountered in die casting, forging tooling, fixture design as well.
   - *Separated*: the geometries of two features do not spatially contact (overlap or are adjacent).
(3) Third level relations are semantic relations between application features. Relation types in this level are application specific. Examples are:
   - *Splitting*. Fig. 5(a) to (c) show a base block with a hole feature. The hole feature is further split by a vertical through slot feature. A similar situation for 2D features is shown in Fig. 5(d) and (e), in which the original

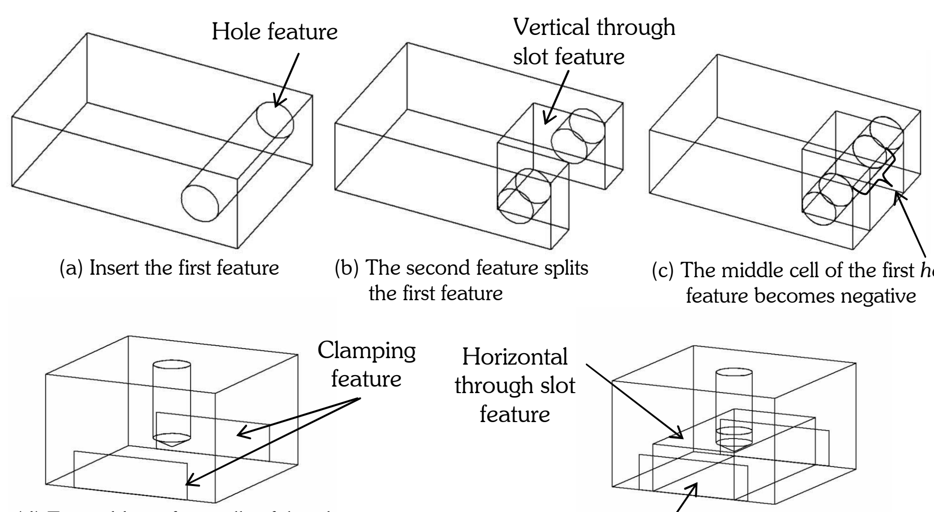
clamping feature is split by a newly inserted through slot feature. The middle face cell of the clamping feature becomes negative. The clamping feature must hence be checked for validity. This kind of relation between two interacting features is defined as a splitting relation [3]. Using the above-mentioned two lower levels of relations, the splitting relation can be described as:

a. The nature of the second feature is negative.
b. The two features overlap.
c. The insertion of the second feature splits the original single cell (additive or negative) of the first feature into several (at least three) cells, where the nature of at least one of the middle cell(s) is negative.



(a) Two adjacent features in position (b) Separated cells

Fig. 4. Completely-adjacent relation between two 3D features.



(a) Insert the first feature (b) The second feature splits the first feature (c) The middle cell of the first *hole* feature becomes negative

(d) Two additive face cells of the *clamping* feature are inserted (e) The middle face cell of the *clamping* feature becomes negative

Fig. 5. Splitting relation.

– *Non-interference*. This relation specifies that two features cannot overlap with or are adjacent to each other. This constraint is satisfied if no n-dimensional cell in the unified cellular model has both of these two features in its owner list. This constraint is commonly used in product design or manufacturing activities. For example, a process planning feature cannot interfere with the corresponding clamping features.

6. CASE STUDY

The unified cellular module is developed using ACIS'a cellular topology component [22]. MySQL [23] is used to develop a database, where the unified cellular model is stored and accessed by different applications. Two examples are given here to illustrate the proposed approach.

Case-1: Linking conceptual and detail design via UCM

Fig. 6 (a) and (b) illustrate the conceptual design while Fig. 6 (c) to (e) show the detail design of the of ejector pins in plastic injection mould design. For each pin, three conceptual features, “imprint”, “propel” and “guide” are derived. The geometries of the “guide” feature are a pair of co-axial cylindrical faces. They are mapped to the sliding hole face of core insert and the cylindrical pin face in detail design respectively. The “propel” feature are a pair of acting and receiving faces. A “complete adjacency” constraint with shrinkage factor is specified between these two faces when “imprint” is considered. This constraint associates the pin tip face and the moulding concave face in detail design.

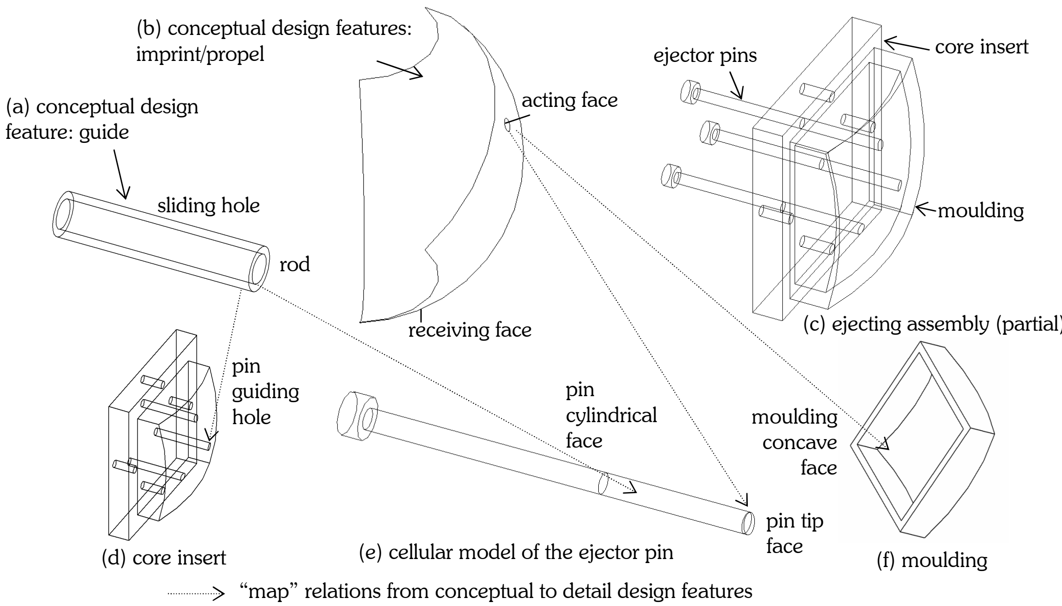


Fig. 6. Using face cells to propagate modifications between conceptual and detail designs.

These mapping associations are established via the unified cellular model, *i.e.* these shared face cells record the corresponding features as owners. For example, the smaller curved face cell in Fig. 6(b) records the “imprint”, “propel”, and “pin” as its owning features in the unified cellular model. When the position of the pin is changed in detail design (*e.g.* due to the change of pin pattern in response to the moulding size change), the unified cellular model is notified for updating. Consequently, the “imprint” and “propel” features in the conceptual design are found to be the associative features and are validated accordingly. Due to the “complete adjacency” constraint the shape of the pin tip face cell is modified too.

Case-2: Linking detail design and process planning via UCM

Fig. 7 illustrates a blind hole in a base block which needs to be drilled. According to the process plan specification, the bottom face is used as the supporting and locating face for the drilling operation. A process planning feature (PP_drilling) and its corresponding supporting feature (PP_supporting) are generated and inserted into the process planning feature model. The corresponding solid and face cells are identified as the run time application cells and also recorded into the unified cellular model. In addition, a non-interference constraint is generated to specify that a process planning feature cannot interfere with its supporting feature. In Fig. 7, “D” represents detail design feature while “PP” represents process planning feature.

Later, the designer decides to change the “*blind hole*” into a “*through hole*” (see Fig. 8). The records (saved in the database) corresponding to the design *hole* feature and the cells in the cellular model are changed and marked with the description of the modification. When the process planning module checks the database and detects the modification, the original drilling feature is deleted first and then a new one inserted into cellular model. This makes the new drilling feature intersect with the supporting feature, *i.e.* cell_5, which carries the identifiers of both drilling and supporting features in its owning feature list. The modification invokes the feature validation process and the violation of the specified non-interference constraint is identified. The process planner may choose another candidate supporting feature if feasible, or here in this simple case, to modify the supporting feature in order to allow the drilling.

7. CONCLUSION

In this paper, a cellular topology based feature modeling scheme is introduced to accommodate different geometry representation requirements for different applications. This unified cellular model can be used to maintain the feature consistency. The main contribution is the novel application of face cells together with solid cells in feature-based modeling. More detailed application feature models (including 1D features) and their user-interfaces based on the unified cellular model, need to be further studied.

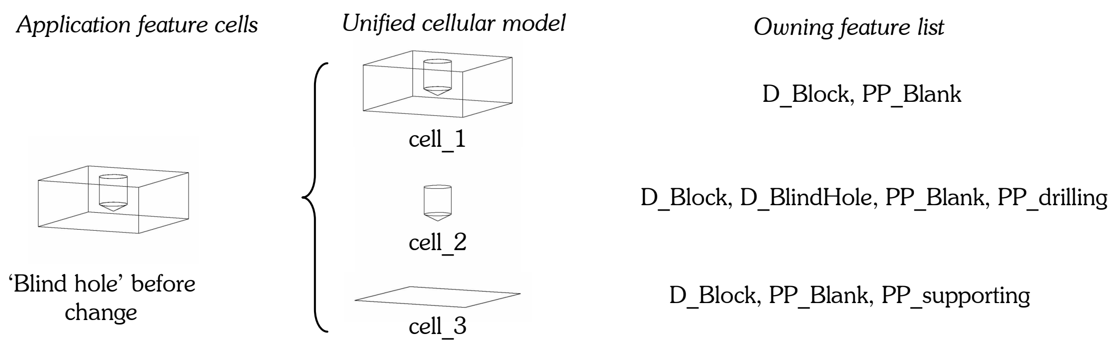


Fig. 7. Identical design and process planning feature cells.

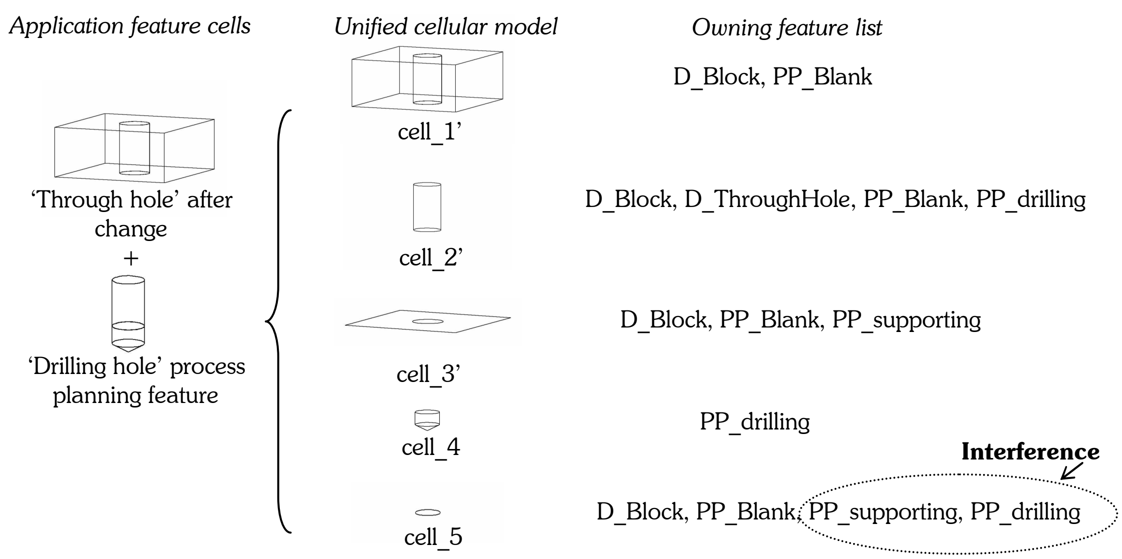


Fig. 8. Propagating modifications using face cells.

8. REFERENCES

[1] Chen, G., Ma, Y.-S., Thimm, G. and Tang, S.-H., Unified feature modeling scheme for the integration of CAD and CAx, *Computer-Aided Design & Applications*, Vol. 1, Nos. 1-4, *CAD'04*, 2004, pp 595-602.

[2] Chen, G., Ma, Y.-S., Thimm, G. and Tang, S.-H., Knowledge-based reasoning in a unified feature modeling scheme, *Computer-Aided Design & Applications*, Vol. 2, Nos. 1-4, *CAD'05*, 2005, pp 173-182.

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

[4] Capoyleas, V., Chen, X.-P. and Hoffmann, C. M., Generic naming in generative, constraint-based design, *Computer-Aided Design*, Vol. 28, No. 1, 1996, pp 17-26.

[5] Kripac, J., A mechanism for persistently naming topological entities in history-based parametric solid models, *Computer-Aided Design*, Vol. 29, No. 2, 1997, pp 113-122.

[6] Lee, S. H., A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, *Computer-Aided Design*, Vol. 37, No. 9, 2005, pp 941-955.
[7] Ma, Y.-S. and Tong, T., Associative feature modeling for concurrent engineering integration, *Computers in Industry*, Vol. 51, No. 1, 2003, pp 51-71.
[8] Rosenman, M. A. and Gero, J. S., Modelling multiple views of design objects in a collaborative CAD environment, *Computer-Aided Design*, Vol. 28, No. 3, 1996, pp 193-205.
[9] Weiler, K., The radial edge structure: a topological representation for non-manifold geometric boundary modeling, In: Wozny, M. J., McLaughlin, H. W. and Encarnacao, J. L., (eds.) *Geometric modeling for CAD applications*, North-Holland, 1988, pp 3-36.
[10] Weiler, K., Boundary graph operators for non-manifold geometric modeling topology representations, In: Wozny, M. J., McLaughlin, H. W. and Encarnacao, J. L., (eds.) *Geometric modeling for CAD applications*, North-Holland, 1988, pp 37-66.
[11] Rossignac, J. R., Issues of feature-based editing and interrogation of solid models, *Computers & Graphics*, Vol. 14, No. 2, 1990, pp 149-172.
[12] Crocker, G. A. and Reinke, W. F., An editable nonmanifold boundary representation, *IEEE Computer Graphics & Applications*, Vol. 11, No. 2, 1991, pp 39-51.
[13] Masuda, H., Topological operators and Boolean operations for complex-based nonmanifold geometric models, *Computer-Aided Design*, Vol. 25, No. 2, 1993, pp 119-129.
[14] Sriram, R. D., Wong, A. and He, L.-X., GNOMES: an object-oriented nonmanifold geometric engine, *Computer-Aided Design*, Vol. 27, No. 11, 1995, pp 853-868.
[15] Deng, Y.-M., Britton, G.A., Lam, Y.C., Tor, S.B. and Ma, Y.-S., A feature-based cad-cae integration model for injection moulded product design, *International Journal of Production Research*, Vol. 40, No. 15, 2002, pp. 3737-3750.
[16] Bidarra, R., de Kraker, K. J. and Bronsvoort, W. F., Representation and management of feature information in a cellular model, *Computer-Aided Design*, Vol. 30, No. 4, 1998, pp 301-313.
[17] Wu, D. and Sarma, R., Dynamic segmentation and incremental editing of boundary representations in a collaborative design environment, In: *Proceedings of the sixth ACM symposium on Solid modeling and applications*, Ann Arbor, Michigan, United States, 2001, pp 289-300.
[18] Lee, J. Y., Lee, J.-H., Kim, H. and Kim, H.-S., A cellular topology-based approach to generating progressive solid models from feature-centric models. *Computer-Aided Design*, Vol. 36, No. 3, 2004, pp 217-229.
[19] Woo, Y., Fast cell-based decomposition and applications to solid modeling. *Computer-Aided Design*, Vol. 35, No. 11, 2003, pp 969-977.
[20] Ma, Y.-S., Britton, G. A., Tor, S. B. and Jin, L.-Y., Associative assembly design features: concept, implementation and application, Accepted by *International Journal of Advanced Manufacturing Technology*.
[21] Bidarra, R., Madeira. J., Neels, W. J. and Bronsvoort, W. F., Efficiency of boundary evaluation for a cellular model, *Computer-Aided Design*, Vol. 37, No. 12, 2005, pp 1266-1284.
[22] Spatial Corp., ACIS 3D Geometric Modeler, Version 7.0, http://www.spatial.com, 2001.
[23] MySQL AB, MySQL Reference Manual for Version 5.0.1-alpha, http://dev.mysql.com/doc/mysql.