

Intelligent Form Feature Interaction Management in a Cellular Modeling Scheme

Rafael Bidarra and José Carlos Teixeira

Grupo de Métodos e Sistemas Gráficos
Departamento de Matemática, Universidade de Coimbra
Largo D. Dinis - Apartado 3008
3000 COIMBRA - PORTUGAL
Tel: (+351) 39 - 28097/8/9
Fax: (+351) 39 - 32568
E-mail: rafa/teixeira@ciuc2.uc.pt

Abstract

Form features present a rather attractive building block in computer-aided design environments for a variety of applications, providing the embodiment of engineering semantics in specific part shape.

In this paper we address the issue of volume feature behaviour throughout interaction phenomena. First, the fundamental concept of form feature as expression of the morphology of a model, is presented. A description of the feature properties, attributes and general constraints that are relevant for our purpose is attempted. In order to develop sound validity conditions for the various classes of form features, feature definitional entities are also introduced. A comprehensive definition of interaction among features is given that encompasses both adjacent and intersecting features.

We use a structured cellular modeling scheme to capture both the morphology of features and the interactions among them. Within this framework, a thorough analysis of feature interactions is performed that explores the accessibility of feature definitional entities in order to assist feature-based model editing and validation.

Operations that create or modify features often interfere with pre-existing ones, producing unanticipated effects that can corrupt or, at least, modify desired feature semantics and/or morphology. On the other hand, valid features can be obtained that exhibit non-standard or disconnected topology as a result of interactions caused by such operations. Systematic management of feature interactions is explored in both cases, from an object-oriented point of view, encapsulating interaction detection methods in feature class definition. This approach is shown to be quite adequate to handle complex interactions among several features. Reasoning mechanisms may, thereafter, be required to handle each situation identified.

Keywords: Feature-based modeling, Geometric reasoning, Intelligent CAD

1. Introduction

Design by features has been given an increasing attention, due to its expressive power as a promising high-level building block for the next generation of CAD systems. These are expected to further automate downstream applications, as well as providing a richer assistance to the design task. With form feature technology, higher level information becomes available, as the engineering significance associated with generic shapes, making it possible for CAD systems to more accurately capture the designer intent [Shah 91a]. Significant improvements in product quality and manufacturability can be expected, if reasoning mechanisms are incorporated in such systems in order to make engineering knowledge available for analysing product designs.

Several solid modelers that implement a high-level feature-based interface, both prototypes and commercial systems, have so far been proposed [Miner 85] [Chung *et al.* 88] [Cutkosky *et al.* 88] [Shah and Rogers 88]. Considerable progress has been achieved in clarifying the concept of form features [Pratt 88] [Shah 91b], as well as in developing several representation schemes for them [Gossard *et al.* 88] [Roy and Liu 88] [Rossignac and O'Connor 89] [Bruzzone and De Floriani 91] [Gomes *et al.* 92]. However, there have not been many practical results in investigating the nature of form feature interactions and their consequences on the validity of features, since the early informal identification of the problem, with [Pratt 87]. This is so, in part, because designing by features begins, indeed, by creating solid models using feature's vocabulary, and that offers no special difficulties when they are inserted in the model disjoint from each other, provided that validity is enforced by insertion methods. Unfortunately, such simple models are of little practical use and, therefore, expansion of our modeling space is required, if we are to model parts of real world solids. On the other hand, the degrees of freedom available in overlapping several features of different types substantially raise the complexity of models and of their manipulation, making it quite hard to systematically approach interaction phenomena.

In this paper we are mainly concerned with creating a semantic framework powerful enough to support the operation of reasoning mechanisms that handle interactions among volumetric form features. We first introduce the essential aspects of form features required for this purpose (section 2), and elaborate on the concept of form feature interaction (section 3). An overview of the structured cellular representation scheme that we are using is given in

section 4, emphasizing its ability to capture both the morphology of feature-based models and the interactions among form features. We, then, elaborate a thorough analysis of the various possible feature interaction situations, exploring their consequences in order to assist feature-based editing and validation.

2. Form Feature Description

A comprehensive definition of form features is still an open issue, and will probably remain so for long, although there is an increasing agreement, in the research community, on a few concepts that are likely to be among the fundamental elements of features. Sometimes, mainly among researchers working within the feature recognition approach, form features have been considered as special collections of faces of the boundary of a solid model, useful for abstraction modeling purposes. Although not incorrect, such a definition is rather incomplete, as well as too vague and imprecise. First, it should be stressed that the criteria for grouping together those topological entities of the model must be taken from morphological considerations, so that they reflect a local shape with specified and predictable properties. This morphological content seems to favour the volumetric approach to form features. On the other hand, the usefulness of features should be found in the vocabulary used in creating and analyzing a design [Shah 91b]. That means the reason for the use of features lies in the association of engineering semantics to some generic shapes intentionally introduced by the designer.

In this section, we attempt the unification of these two aspects, bringing together the morphological expressive power of form features and their semantic composition.

2.1. Form Features and Morphology

Recent theoretic research work on the morphology of solids has shown that the notion of morphology, rather than just topology or geometry, is crucial in the development of solid models, if they are intended to be the basis for complete and sound form feature models [Gomes 92]. Indeed, we believe that most difficulties presented so far in the creation, editing and validation of feature-based models are due to an imprecise and vague understanding of morphology's role in solid models and, particularly, its expression through form features.

We propose that the local morphology captured by a form feature comprises both its additive or subtractive volumetric nature, the topology of this volume, and the characteristics of its boundary, namely the specific subsets of the boundary that may or may not actually belong to the model's boundary.

2.2. Form Feature Semantic Components

In order to obtain a complete description of form features, it is necessary to identify those semantic components that are associated with their geometric, topological and morphological properties. For the purposes of the research work being discussed, we have derived the following classification for such components:

2.2.1. Dimension parameters

Dimension parameters represent the quantitative geometric properties of the feature, e.g., slot width or pocket depth. Their values may be either directly obtained from the designer or inherited from those of other dimensional parameters, eventually related through some specified function.

2.2.2. Location and orientation parameters

Location and orientation parameters describe the actual positioning of the feature. Most commonly, they are specified relative to some selected entities of the model, thus providing a natural method for encapsulating references to positioning entities of other features. The relationships involving these dependent parameters can be algebraic (e.g., insert a slot so as to split a face in two halves) or geometric (e.g., impose parallelism between two faces), in which case they are regarded as associative geometric constraints [Arbab and Wang 88] [Teixeira and Kilian 89]; when they are explicitly introduced by the designer, we call them *geometric design constraints* [Suzuki *et al.* 90].

2.2.3. Definitional entities

Several taxonomies for form features have been proposed so far [Wilson and Pratt 88] [ISO 92] [Ovtcharova *et al.* 92]. In all of them, a classification is made that captures, explicitly or not, for each feature class, some generic morphology with a definite semantic content. Besides the additive or subtractive nature intrinsic to each class, the morphology expressed by its instance features can be distinguished from that of the other classes by comparing the specific behaviour of the feature's boundary on the model. This

lead us to identify for each feature class the essential collections of topological entities on its boundary: we call each of these collections a *definitional entity*; together, they make up the feature *definitional entities set*. Let us illustrate this point with an example: we can devise several volumetric form features whose associated volumes are congruent, like the slot, the pocket and the through hole in the model of Figure 1.(a).

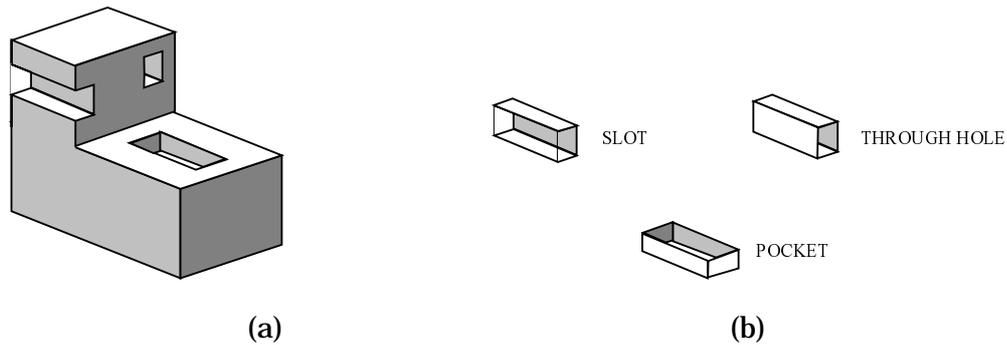


Figure 1. Distinction of form features by their definitional entities

Once they are all subtractive, the distinguishing characteristics of each one can only be found on the different roles played by the arrangement of the various faces on the respective boundary, as depicted in Figure 1.(b).

Definitional entities have a status attribute that may be set either to positive, in which case the elements of the respective collection are part of the model's boundary, or negative, i.e., those elements do not belong to that boundary. For instance, a slot should have methods for accessing its positive *floor* or its negative *roof* in the model, as exemplified in the SLOT1 of Figure 2.

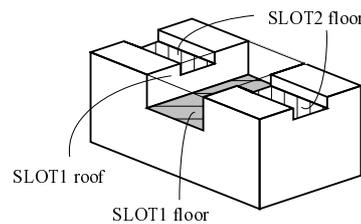


Figure 2. Example of definitional entities of slots in a model

One of the main advantages of definitional entities definition is that it makes no assumption on the adjacency among the topological entities it comprises, provided that they are all coplanar. In this way, for example, the definitional entity *floor* of the SLOT2 in Figure 2 is composed of two faces of the model.

2.2.4. Semantic constraints

As stated before, each feature class is expected to exhibit a specific behaviour on the local morphology of the model. Therefore, every feature should be required to satisfy a set of constraints imposed by its own semantic nature; this kind of feature class-dependent constraints will be here called *semantic constraints*. Desired feature specific behaviour can be achieved by imposing the status of selected definitional entities. When all constrained definitional entities of a particular feature are non-empty, we say it presents semantic completeness and the feature is, therefore, valid. For instance, a slot should always have a non-empty positive *floor*, no matter if it is composed of two disjoint faces of the model, as shown in Figure 2 for SLOT2.

Another kind of semantic constraints provides the ability to express direct relationships with entities or properties of other features. Examples of these include parameter inheritance (e.g., the depth of a through hole being equal to the height of a block), position or orientation dependency (e.g., between elementary features in a compound feature) and insertion face coplanarity (e.g., forcing a slot *roof* to lie on a particular insertion face of the model).

2.2.5. Validity conditions

Invalid features diverge, in some way, from the original morphology desired with their insertion in the model. With definitional entities established for elementary form feature classes and a set of semantic constraints imposed on them, it becomes possible to assess the validity of each feature instance present in the model. With this aim, it is necessary to settle the appropriate validity conditions for each feature class. As a general rule, when a semantic constraint of a form feature is infringed, the feature becomes invalid. This can occur due to some feature insertions or manipulations, as will be dealt with extensively in section 5. Furthermore, additional validity conditions can be set for a particular feature class, or even for a single feature instance, for example, confining parameter values to a limited range. When required, validity conditions can provide an explanation on the reasons that make a particular feature invalid.

Invalid features can be maintained in the model, if not otherwise stated, although the respective desired morphology is, perhaps temporarily, overridden; in such situations they are considered as intentional features, just reflecting some intention of the designer on the corresponding functionality [Rossignac 90].

3. Definition of Form Feature Interaction

As outlined in the introduction, creation of solid models using feature's vocabulary can be a straightforward task, provided that features are inserted in the model disjoint from each other. An example of such a model is shown in Figure 3.

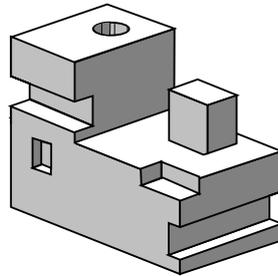


Figure 3. Model with disjoint features

Actually, however, machining paths in a real object often interfere, expressing some more complex functionality intended by the designer. In the corresponding feature model, such interference is called *feature interaction*, and it can take place both at feature insertion stage, due to some common geometry or location introduced, or in post-insertion stages, by modification of any of these parameters. As an example of the latter, Figure 4 presents the same model of Figure 3 after a few modifications on some dimension and location parameters of several features.

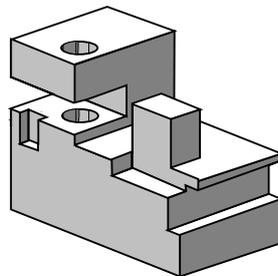


Figure 4. Modified model of Figure 3 showing feature interactions

This example emphasizes that interaction between features can occur in two ways: volume interaction or boundary interaction. In both cases, there is always a region of the space where the closed volumes associated with each feature intersect: this overlapping region (whatever its dimensionality) will be here called *interaction extent*, and its existence is a necessary and sufficient condition for the occurrence of form feature interactions. Hence the following

Definition: Two form features are said to be in interaction whenever their associated closed volumes intersect.

Clearly, this definition is more accurate, and somewhat simpler, than just requiring that topological entities intersect in a way that either the shape or semantics of a feature are altered from the standard or generic definition, as proposed by [Shah 91b]. In this way, adjacent features, as defined there (features that share one or more common topological entities) are naturally included in our definition above, as it will be shown in section 5.1.

4. Form Feature Secondary Representation

As anticipated in section 2.1, morphological completeness of solid models is a pre-requisite to support a convenient representation of form features [Gomes 92]. To represent form features, we are using a cellular representation scheme called A-Crep (Adaptive Cellular Representation) that is morphologically complete, as demonstrated in [Gomes and Teixeira 93].

The A-Crep model is based on the theory of adaptive cell complexes. An adaptive cell complex is a collection of generalized cell subcomplexes, each of which is labelled with P (positive) or N (negative). This model allows the secondary representation of form features by associating each form feature to a labelled cell subcomplex; for instance, a slot is represented by means of a negative cell subcomplex, whereas a protrusion is represented by a positive one.

Whenever two features interact, the respective subcomplexes are further decomposed. This cellular decomposition is interaction-driven, i.e., part of the resulting cells represent the interaction extent and, thus, belong to both subcomplexes (interaction cells), while the remaining cells belong to the subcomplex of either feature (non-interaction cells).

5. Reasoning Mechanisms for Feature Interaction Management

In this section we go into the problem of feature behaviour throughout various types of interactions. For each interaction class, we first give a formal definition, in terms of feature definitional entities, and explore its particular consequences on the validity of the involved interacting features. Our approach can be viewed as an intent to encapsulate interaction detection and reaction methods in each feature class definition, thus providing an automated mechanism for feature validity maintenance throughout interaction phenomena. When convenient, we will refer to the notions introduced in the previous section, in order to illustrate the response of the cellular representation level to feature manipulations and interactions management;

in this case, figures showing cellular decompositions will be presented in white, rather than shaded.

It should be remarked that the occurrence of invalid features in a model is most often an indirect side effect that takes place as an undesired or, at least, unanticipated result of feature interactions. For this reason, it can be allowed to protect intentionally a given feature, preventing it from being corrupted by interactions with others. This protection on the features present in a model introduces a hierarchy among them, based on the particular interaction classes it disables for each feature.

5.1. Topological Interaction

A large variety of interactions can be devised that preserve the designer intent for some specified local morphology, despite the overlapping of the involved features. In other terms, we can say that, after such interactions, each feature presents semantic completeness. In all these situations, topological entities of both features intersect in some way, and we say that the features are in *topological interaction*. Hence, we have the following

Definition: *Topological interaction is the interaction between form features that overlap (with boundary or volume intersection), while maintaining each one its own parameters and a semantically complete definitional entity set.*

Some examples of topological interaction situations are given in Figure 5.

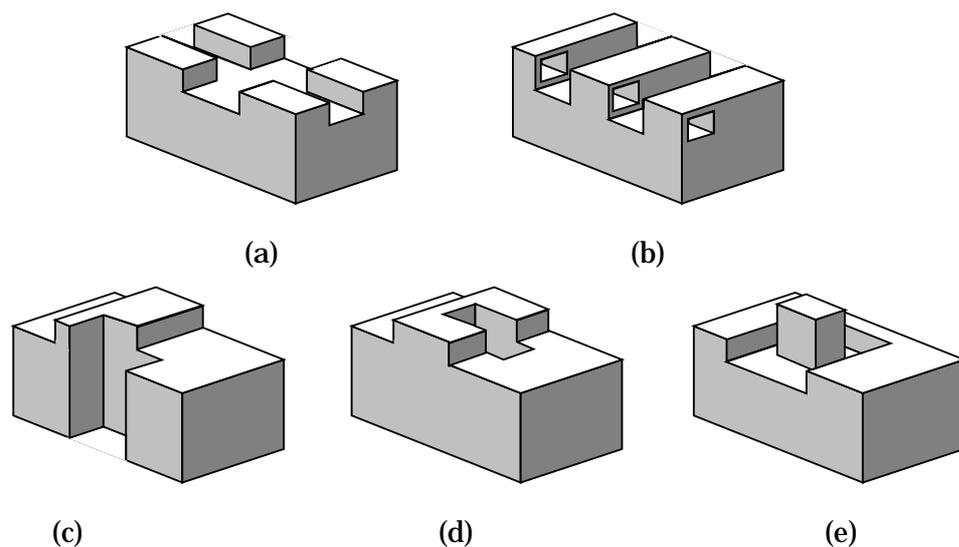


Figure 5. Examples of features in topological interaction

The form features that compose the models shown in Figure 5 are all valid, although some of them exhibit non-standard or disconnected topology. For instance, both slots in (a) have two topological components, split apart by each other. At the representation level, they have undergone a cellular decomposition that captures the interaction extent of both, by means of a shared cell, as depicted in Figure 6.

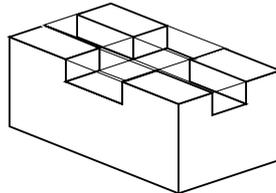


Figure 6. Cellular decomposition of the model in Figure 5.(a)

This example also illustrates the adequacy of such a cellular representation scheme to solve the problem of contiguity of form features, without need of artificial joining faces [Shah 91b]: the disconnected situation of the walls in both slots is overcome by the higher-level concept of feature definitional entities, that tie them together, as shown in Figure 6.

Editing features in topological interaction can be dealt with, at the representation level, in a straightforward way by propagation of the feature level manipulation to the cells of the corresponding subcomplex, subject to the various constraints imposed by the other interacting features. For instance, deletion of one of the slots of Figure 6 is carried out in three steps: first, the slot disposes the interacting cell of its subcomplex to the other interacting features (the other slot and the block); these are then asked to perform a cellular merging process over their respective subcomplexes, based on their sharing ownership attributes (adjacent cells with the same list of owners are merged); at last, the slot is removed from the model, after removing all references to its parameters in the other features.

Subtractive features in models (c) and (d) of Figure 5 exhibit a non-standard topology, due to the particular location of their interaction extent. In order to normalize the morphology they express, they include, at the representation level, a negative non-interaction cell, as depicted in Figure 7.(a).

In this way, feature properties such as dimension parameters or definitional entities can be kept consistent and, furthermore, subsequent editions are treated uniformly. For example, enlargement of the protrusion width in Figure 7.(a) could eventually lead to the total inclusion of the through

hole upper bound, as shown in (b), thus absorbing its former non-interaction cell.

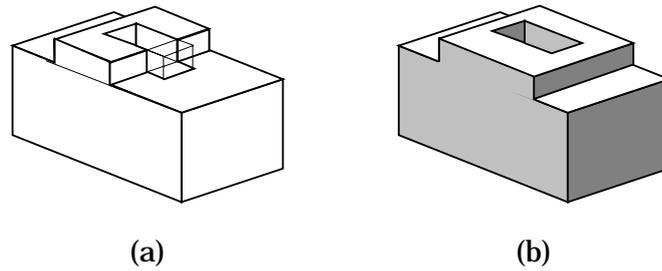


Figure 7. Negative non-interaction cell in the model of Figure 5.(d)

5.2. Transmutation Interaction

In certain conditions, operations that create or modify features may produce destructive consequences on the local morphology they express in the model. One of these effects is called *feature transmutation*, and it can be defined as follows:

Definition: *Feature transmutation is an essential morphological modification that causes a given feature to exhibit the morphology specific of another feature type or class.*

Transmutations may be expressed in terms of definitional entities, observing that they always cause a constrained definitional entity of a feature to become empty. A feature that suffers a transmutation is said to be in *transmutation interaction*, as stated in the following

Definition: *Transmutation interaction is the interaction between features that produces a transmutation in any of them.*

A few examples of transmutation interactions are shown in Figure 8. In (a), displacement of the through hole produces a slot; in (b), enlargement of the slot transmutes the pocket, whose morphology becomes that of a slot; in (c), deletion of a slot would cause the through hole to become a blind one; and in (d), two pockets, otherwise valid since disjoint, combine to yield a slot.

As shown, these situations can occur either from editing (a) or deleting (c) an existing feature, or due to the insertion of a new one in the model (d). Still, the edition of an existing feature, or the insertion of a new one, can produce a transmutation both in that feature, (a) and (d), and in any other that becomes subject to its interaction, (b) and (c).

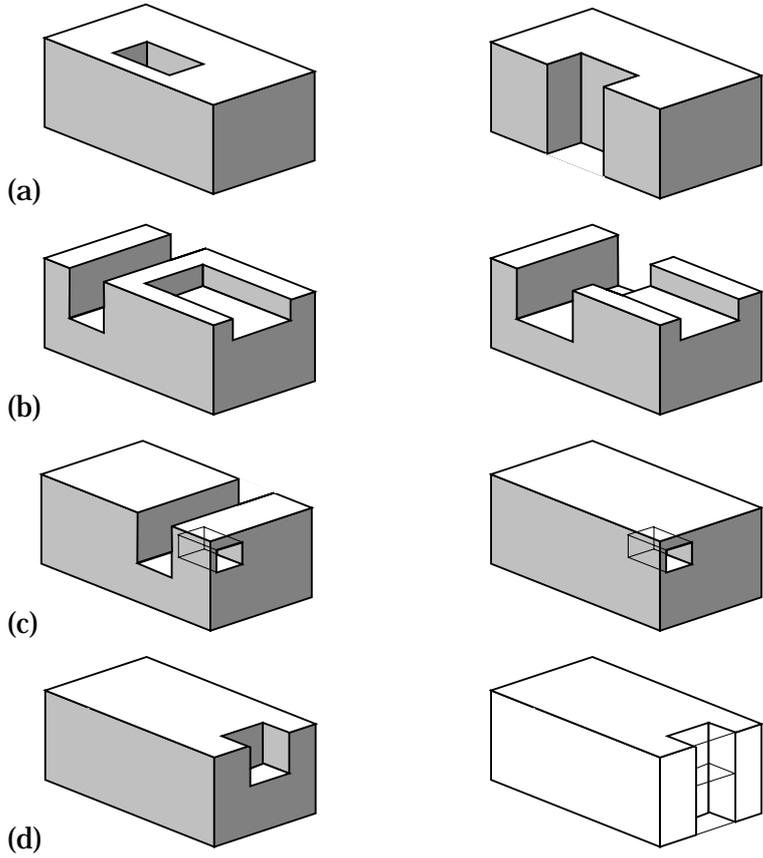


Figure 8. Examples of features in transmutation interaction

Transmutation interactions are easily detected because they always require a semantically constrained definitional entity of some feature to become empty. Providing that each feature class has its own semantic constraints defined on existing definitional entities (whether positive or negative), any attempt to modify their status can be directly signaled and handled accordingly to the knowledge expressed in the validity conditions of the respective class. For instance, the transmutation of the through hole of Figure 8.(a) into a slot causes one of its positive *walls* to become the negative *roof* of a slot. On the other hand, one of the three positive *sides* of the pocket in Figure 8.(b) is turned into a negative one, thus producing its transmutation into a slot.

A special case of transmutation can occur in which the affected feature changes to a different type of the some feature class. An example of such a

situation is given in Figure 9, where a pocket with four *sides* is changed to a pocket with three *sides*, due to the insertion of the slot; one of the real definitional *sides* of the pocket becomes negative, but it still keeps the morphology of a pocket, with positive *floor* and contiguous *sides*.

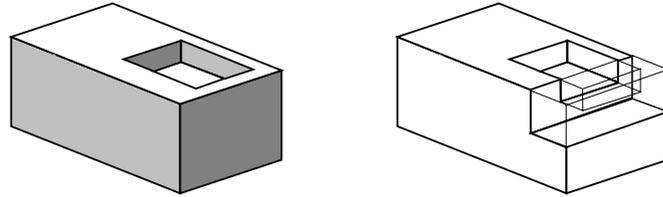


Figure 9. Example of pocket type transmutation interaction

This kind of *soft* transmutation could, in principle, be allowed unconditionally, provided that it always yields valid and, more important, reversible situations. In all the other cases, assuming that the designer's degree of intentionality and/or advertence is unpredictable, transmutations are always detected and signaled, requiring his/her confirmation before the transmutation is carried on.

5.3. Geometric Interaction

Some interactions do not affect the morphology of a particular feature, but instead cause a transformation on the geometry of its definitional entities, and we say it is in *geometric interaction*. This interaction can occur to subtractive features whose geometry is modified by the edition or insertion of another interacting feature, while keeping the complete set of definitional entities.

Definition: *Geometric interaction is the interaction between features in which the particular interaction extent causes some dimension parameters, established at feature insertion stage, to lose their correspondence to the actual feature geometry.*

Examples of geometric interactions are presented in Figure 10, where the internal cellular decomposition of each feature is also shown (white pictures). In (a), the length of the blind hole is reduced because of the insertion of a pocket; in (b), the pocket sees its actual length reduced, first due to the insertion of a slot, and second because of the slot enlargement.

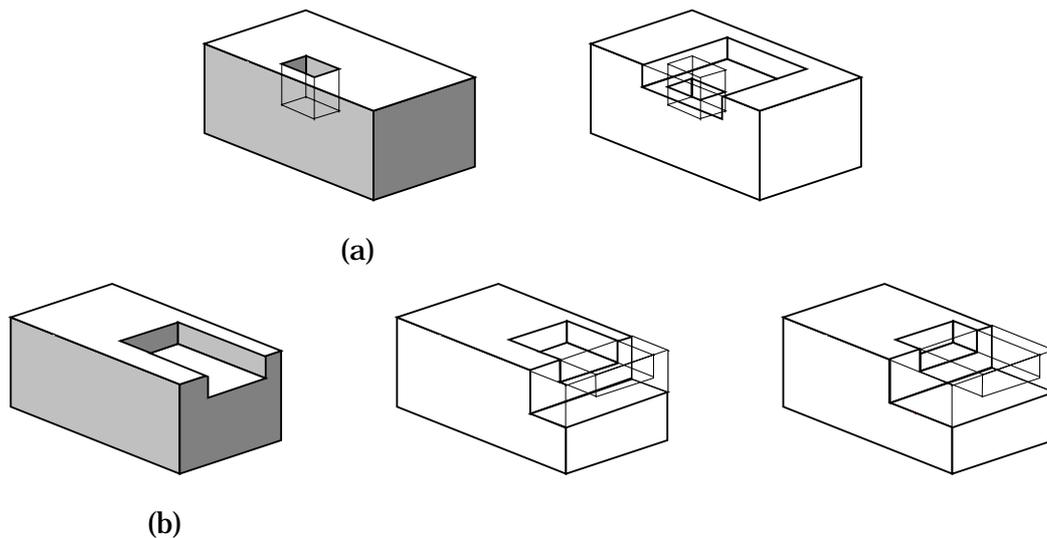


Figure 10. Examples of features in geometric interaction

The problem with geometric interaction is the indirect change in dimension parameters that were introduced intentionally by the designer for some feature. On the one hand, it seems reasonable to keep their original values, for the sake of history maintenance and consistency of subsequent editions. For instance, deletion of the slot in Figure 10.(b) should bring the model back to the former situation shown; however, the same deletion in the similar model of Figure 9, should restore its original situation. On the other hand, actual parameters should be available for user interrogation and information purposes. One solution consists in distinguishing between virtual (or insertion) parameters and real (or actual) parameters. At feature insertion stage, they are both assigned the same values. When a geometric interaction takes place, a real parameter is updated, accordingly to the new geometry of the feature, thus reflecting the effects of the interaction. This can be achieved by establishing for the real parameter an expression involving the former virtual value and the appropriate parameter(s) of the other interacting feature(s).

The interaction-driven cellular representation used also captures naturally this class of interaction: the final decomposition produced splits the affected feature's subcomplex in two, such that one of the parts, representing the interaction extent, is embedded in the subcomplex of the new interacting feature, and the other part still reflects the desired morphology as expressed through its original semantic constraints. This situation is clearly depicted in Figure 10, where the cell decomposition exhibits both split parts of the affected feature.

5.4. Closure and Absortion Interactions

Subtractive features are supposed to be always accessible from outside of the model, as we cannot devise real objects with voids inside. When the associated volume of such features becomes a void completely closed inside of the model (Figure 11), they are said to be in *closure interaction*:

Definition: *Closure interaction is the interaction in which all negative definitional entities of a subtractive feature become empty, i.e., its boundary totally lies on the model's boundary.*

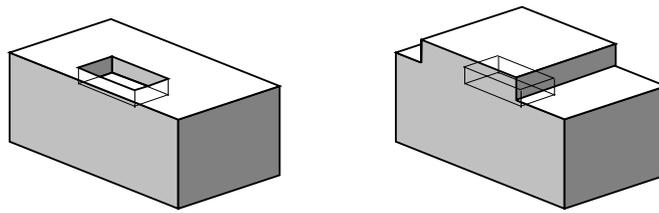


Figure 11. Example of a feature in closure interaction

When the associated volume of a subtractive feature becomes completely incorporated in some subtractive feature (Figure 12), we say it is in *absortion interaction*:

Definition: *Absortion interaction is the interaction in which all positive definitional entities of a given feature become empty, i.e., no subset of its boundary lies on the model's boundary.*

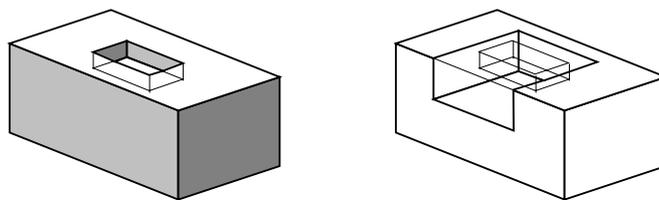


Figure 12. Example of a feature in absortion interaction

As defined above, closure and absortion interactions are dual, in terms of definitional entities modifications. They are detected in a straightforward way, just like the other classes of interactions, due to the semantic expressiveness of definitional entities. Their detection relies on the feature's capability of signalling emptiness in each definitional entity, which is easily achieved, in our object-oriented approach, by means of a general method defined for all feature classes with that purpose.

6. Conclusions

The necessity for modeling solid objects with complex shapes demands from feature-based solid modelers the ability to properly manage form feature interactions.

A semantic framework that supports reasoning mechanisms for handling interaction situations was developed, based on the morphological expressive power of form features and on their semantic composition. For this purpose, we introduced the concepts of definitional entities and semantic constraints, which proved to be of great utility both for the classification of the various types of interactions and for their automatic detection.

Feature validity maintenance throughout interaction phenomena is a key point in feature-based modeling, and higher level mechanisms for assisting edition and interrogation of feature models need to be further investigated. Future trends in our research work include developing some of these mechanisms, validating them in an object-oriented version of A-Crep, that is currently under implementation, and extending the above concepts to compound and user-defined form features .

Acknowledgements

The authors would like to thank Abel Gomes for many fruitful discussions on several topics covered in this paper, particularly those related to the cellular representation of form features, and for his valuable suggestions and careful reading of the manuscript.

References

- [Arbab and Wang 88] Arbab, F. and Wang, B.: Reasoning about Geometric Constraints. *Proceedings of II IFIP WG5.2 Workshop on Intelligent CAD*, Cambridge, UK, 1988.
- [Bruzzone and De Floriani 91] Bruzzone, E. and De Floriani, L.: Representing a Modular Boundary Model Through a Cellular Decomposition. *Eurographics Workshop on Computer Graphics and Mathematics*, Genova, Italy, October 1991.
- [Chung *et al.* 88] Chung, J., Cook, R., Patel, D., and Simmons, M.: Feature-Based Geometry Construction for Geometric Reasoning. *Proceedings of ASME Computers in Engineering Conference*, San Francisco, USA, July/August 1988.
- [Cutkosky *et al.* 88] Cutkosky, M., Tenenbaum, J., and Miller, D.: Features in Process based Design. *Proceedings of ASME Computers in Engineering Conference*, San Francisco, USA, July/August 1988.

- [Gomes 92] Gomes, A.: Modelos Algébricos de Sólidos e Morfologia. Work submitted to "Provas de Aptidão Pedagógica e Capacidade Científica" (Master Thesis, in portuguese), Mathematics Department, University of Coimbra, 1992.
- [Gomes *et al.* 92] Gomes, A., Bidarra, R., and Teixeira, J.: A Cellular Approach for Feature-Based Modelling. In *Workshop on Graphics Modelling & Visualization in Science and Technology*, Darmstadt, April 1992 Göbel, M. and Teixeira, J. (Eds.), Springer-Verlag (to be published)
- [Gomes and Teixeira 93] Gomes, A. and Teixeira, J.: On the Morphology of Cellular Solid Models. Paper submitted to IFIP TC5/WG5.10 II *Working Conference on Geometric Modeling in Computer Graphics*, Genova, Italy, June/July 1993.
- [Gossard *et al.* 88] Gossard, D., Zuffante, R., Sakurai, H.: Representing Dimensions, Tolerances, and Features in MCAE Systems. *IEEE Computer Graphics & Applications*, vol.8, no.3, 1988.
- [ISO 92] ISO TC184/SC4/WG3 N102 P5, Industrial Automation Systems - Product Data Representation and Exchange - Part 48: Integrated Generic Resources: Form Features, Working Draft, Mark Dunn (Ed.), 1992
- [Miner 85] Miner, R.: A Method for the Representation and Manipulation of Geometric Features in a Solid Model. MS Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, USA, May 1985.
- [Ovtcharova *et al.* 92] Ovtcharova, J., Pahl, G., and Rix, J.: A Proposal for Feature Classification in Feature-Based Design. *Computers & Graphics*, vol.16, no.2, 1992.
- [Pratt 87] Pratt, M.: Recent Research in Form Features. In *Advanced Topics in Solid Modelling*, ACM Siggraph 87, Course #26, Anaheim, CA, July 1987.
- [Pratt 88] Pratt, M.: Synthesis of An Optimal Approach to Form feature Modeling. In *Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, vol.1, The American Society of Mechanical Engineers, New York, pp.263-274, 1988.
- [Rossignac 90] Rossignac, J.: Issues On Feature-Based Editing and Interrogation of Solid Models. *Computers & Graphics*, vol.14, no.2, pp.149-172, 1990.
- [Rossignac and O'Connor 89] Rossignac, J. and O'Connor, M.: SGC: A Dimension-Independent Model for Pointsets with Internal Structures and Incomplete Boundaries. In *Geometric Modeling for Product Engineering*, Proceedings of the 1988 IFIP/NSF Workshop on Geometric Modeling, Rensselaerville, NY, September 18-22, 1988, Wozny, M., Turner, J., and Preiss, K. (Eds.), North-Holland, The Netherlands, pp.145-180, 1989.

- [Roy and Liu 88] Roy, U. and Liu, C.: Feature-Based Representational Scheme of a Solid Modeler for Providing Dimensioning and Tolerancing Information. *Robot. & Comput. Integr. Manuf.*, vol.4, no.3/4, 1988.
- [Shah 91a] Shah, J.: Assessment of Features Technology. *Computer Aided Design*, vol.23, no.5, 1991.
- [Shah 91b] Shah, J.: Conceptual Development of Form Features and Feature Modelers. *Research in Engineering Design*, vol.2, pp.93-108, 1991.
- [Shah and Rogers 88] Shah, J. and Rogers, M.: Expert Form Feature Modeling Shell. *Computer Aided Design*, vol.20, no.9, 1988.
- [Suzuki *et al.* 90] Suzuki, H., Ando, H., and Kimura, F.: Geometric Constraints and Reasoning For Geometrical CAD Systems. *Computers & Graphics*, vol.14, no.2, pp.211-224, 1990.
- [Teixeira and Kilian 89] Teixeira, J. and Kilian, C.: The Associativity as a Tool for CAD. In *Computer Applications in Product and Engineering - CAPE '89*, Kimura, F., and Rolstadas, A. (Eds.), Elsevier Science Publishers, B.V. (North-Holland), IFIP, 1989.
- [Wilson and Pratt 87] Wilson, P. and Pratt, M.: A Taxonomy of Features for Solid Modeling. In *Geometric Modeling for CAD Applications*, Wozny, M., McLaughlin, H., and Encarnaçao, J. (Eds.), Elsevier Science Publishers, B.V. (North-Holland), IFIP, 1988.