The Difference between two Feature Models

Matthijs Sypkens Smit[1] and Willem F. Bronsvoort[2]

[1]Delft University of Technology, M.SypkensSmit@tudelft.nl
[2]Delft University of Technology, W.F.Bronsvoort@tudelft.nl

ABSTRACT

In feature modelling, a sequence of feature models is created, in which the geometry and topology evolve to their desired state. For several applications, in particular remeshing models for FE analysis, it is useful to know the difference between two models.

In terms of features, the difference is often small. Many entities of feature geometry can be mapped between two related feature models, either completely or partially. Entities that can be mapped are persistent between the two models, the other entities are either new or old. Such a qualification is not unique, but depends on the feature that owns or interacts with an entity; entities shared by multiple features can have several qualifications.

We present a way to describe this difference between two models in terms of features, and a simple approach to construct such a description. A cellular model is used to store the persistence qualifications for each feature.

Keywords: feature modelling, geometric similarity, remeshing.

1. INTRODUCTION

Feature modelling is nowadays the prevalent approach to product modelling. Any system that, through shapes that compose the geometry, adds more information than geometry to the model, can be considered a feature modeller. Some examples of data commonly attached to the model through features are design intent, properties of the material and machining data. An interesting question in feature modelling is to describe or quantify the differences between two feature models.

The straightforward way to find the difference between two geometric shapes is with a boolean (symmetric) difference operation. This operation only takes the BReps of the models into account. Features, however, can describe more than just the geometry of the boundary. They can also overlap and interact internally to the volume bounded by a BRep. Therefore the boolean difference is not powerful enough to handle feature models in a more general way than just in terms of the geometry of their BRep. Furthermore, when considering the geometric difference of two models with a single boolean difference, the result does not properly capture differences and similarity as we intuitively perceive it: we can regard an individual, translated feature as identical in two models, whereas the global boolean difference will result in both the disappearance and creation of new geometry.

Recently there has been a lot of interest to identify geometric similarity between models in ways that better relate to a designer's intuition [11]. These methods roughly aim to identify geometric features and their relations, and compare the resulting *shape signatures* of the models to assess similarity. They clearly succeed in yielding a result that connects more closely to our intuition of similarity. However, typically the result of a similarity analysis does not result in a map that relates specific parts of the geometry directly between the models. The methods that are based on an explicit break-down of the geometry, such as the ones recently presented in [3] and [4], do help to relate specific parts or regions between models, but in general the relation does not allow for a one-to-one mapping between geometric elements. Moreover, their conception of a feature is purely geometric, with little correspondence to the broader feature concept that attaches meaning or knowledge to parts of the model.

The aim of our work is to describe the difference between two feature models, in terms of general features. All those features have a clearly defined geometry. Also, the features have a meaning that should be taken into account for the resulting difference. Furthermore, we want the description of the difference to be geometrically precise. It should be possible to map similar feature geometry between the two models and to identify exactly those parts that are exclusive to either one of the models.

Our work is primarily motivated by the problem of remeshing a 3D model with finite elements after model modification. Previous work in this area includes [10] and [12]; for an overview of current issues in meshing, see [13]. For the remeshing of complex geometries with minimal redundancy, we need a detailed description of the difference between the geometry of the previous, already meshed, model and the new model. It might also be of use to other applications that need to map data that is attached or related to features between models.

We begin the presentation of our work by discussing feature models. Then in Section 3 we describe what we mean by the geometric difference in terms of features. In Section 4 we discuss the representation of the difference and the outline of a method to construct such a representation. We end with a discussion of possible applications in Section 5 and general remarks and conclusions in Section 6.

2. FEATURE MODELS

Feature modelling is a major step forward from design with basic geometric shapes. A feature model connects closely with how the user intuitively sees an object as a combination of shape aspects. A feature represents such shape aspects and has some functional meaning [6]. The use of features is not limited to pure geometric design. Each design activity can have different features, related to the intuitive decomposition of the model from the point of view of that design phase [8]. An overview of current developments in feature modelling is given in [7].

The geometry of a feature is parameterised. The parameter values can be given explicitly by the designer or be derived from constraints on the model. For a specific model instantiation, all the values of feature parameters have to be known, together with a set of relations that uniquely defines the relative positioning of the features. This, however, is not enough for a complete description of the model. There also needs to be some information, either implicit or explicit, on how to resolve overlapping features. In the case of features that define the model geometry: if two features of positive and negative nature, indicating respectively that material is present or not, overlap, it needs to be resolved what the result in the overlapping parts is. A common solution is that the most recently added or modified feature determines the interpretation of the interaction domain. A variation is the use of a feature dependency graph, whereby dependent features determine the interpretation when interacting with the features on which they depend [6]. Neither of these solutions works for all possible cases [14]. For our work, it is not relevant how exactly the interactions are resolved, so we just assume that there is a method that takes care of this.

The problem of interpreting interacting features extends beyond the case of positive and negative nature, to interacting feature aspects in general. An artificial example would be a feature with a 'colour attribute', which gives both the boundary and the volume a colour value. This is not a simple boolean value, like in the case of nature. In areas where features of different colour overlap, it must be resolved which colour, possibly a 'mix', is assigned. The need for interaction resolution is common to all entities, including edges and vertices. In some cases, such as distilling the BRep from a set of interacting features, the contribution of lower dimensional entities depends trivially on the volumetric contributions, i.e. only entities that are adjacent to volumetric contributions of different nature, are part of the BRep. It cannot be assumed that a trivial solution for resolving the interpretation for entities of lower dimensions applies to feature interaction in general.

Various types of representation are used for feature models. The BRep is a straightforward choice, but it has deficiencies in the context of advanced feature models that need to store information on the overlap of features or, more in general, that need to maintain feature semantics [6]. The cellular model seems to be a more suitable representation in this context [5]. We therefore use the cellular model for the representation of the feature models.

The cellular model captures all interactions of features, including interactions that do not affect the geometry of the boundary of the model. It is a non-manifold geometric representation of a feature model. For a model that consists of a single volume, its cellular model is a connected set of quasi-disjoint cells such that the geometry of each feature is represented by a set of cells. The subdivision into cells is determined by the property that no two cells can overlap volumetrically. Whenever two features overlap, their geometry is split into cells such that each cell either completely describes an overlapping section or the volume covered by the cell is exclusive to a single feature.

Each face in the cellular model delimits either two adjacent cells, or a cell and the outside of the model. For each face we can thus discern two sides or *cell faces*. Each cell can be described in terms of its set of cell faces. This is the basic cellular model. The model can be extended such that the edges and vertices can also be accessed as entities belonging to a particular cell. In that case, however, we obviously cannot use the concept of 'sides', since the number of cells that can share an edge or vertex is not fixed.

For each cell, cell face, and optionally edge and vertex, it is recorded in an *ownerlist* to which features it belongs. This is essential for manipulation of and reasoning with the features. The meaning or interpretation of each entity is also stored. In the simplest case, considered here, this is the nature of the entity, which indicates for three-dimensional cells whether the volume lies inside or outside the model. The concept of nature can be extended to lower dimensional

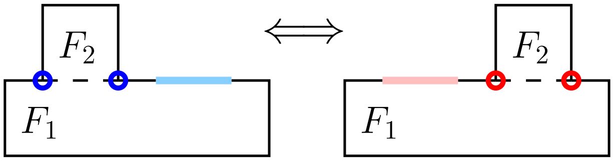

Fig 1: Two feature models with differences highlighted.

entities, where it indicates whether the entities are part of the model boundary. We refer to the entities that are part of the boundary and to the cells that lie inside the volume, as *positive* entities, and to the entities that are not positive, as *negative*. The cellular model can be built and modified by the addition and removal of individual features, and supports a range of methods for querying the ownership and interpretation of entities and their relation to other entities in the model [5].

In the sequel of the paper, we assume that the feature models under consideration are represented by a cellular model. Our representation of the difference between two feature models relies on a structure resembling a cellular model.

3. THE DIFFERENCE BETWEEN TWO MODELS

We assume that the differences between two models that we describe are relatively small, in particular, that the models are variations in the evolution of a single design. It should be possible to relate corresponding features between the two models.

To illustrate our ideas, we use simple, two-dimensional examples. However, the principal application in mind is three-dimensional modelling. For easy interpretation, all examples, unless stated differently, deal with the geometric contribution of the features to the boundary and volume of the model, thus whether the entities are considered positive or negative.

We aim to describe the difference between two models in terms of their features. The difference can be split into 1) difference in geometry, and 2) difference in interpretation of geometric entities. For a particular feature, the local geometry depends on the feature shape and its interaction with other features, but *not* on its absolute location. Each feature has its own geometry and, when it is translated, it carries this geometry with it to the new location. This means that the translation of a feature will not necessarily result in a difference in geometry, *from the point of view of that feature*. A change in the interpretation of geometric entities amounts to a change of meaning, e.g. a change in nature.

The question we attempt to answer is: given two feature models, what is the difference in terms of features and how can the similarities between the models be mapped? As we have pointed out in Section 1, a global geometric difference, e.g. a Boolean difference, can only give a coarse result that does not take sub-part correspondences into account. Methods that do look at local geometric similarities, such as [3] and [4], are too general to consistently allow the mapping of individual geometric elements to one another. Furthermore, none of these methods deal with the meaning that is attached to the model by means of features. That is why we regard the difference from the point of view of individual features.

We aim to describe for each feature separately how its geometric involvement and the feature meaning it carries differ between two models. Stated differently, we consider which parts of the vertices, edges, faces and volume can be mapped between the two models, which parts belong to only one of the two models, and for which parts the feature interpretation has changed. The last question is only relevant for those entities that can be mapped between the two models.

Describing the difference therefore breaks down into marking entities as either geometrically *persistent* (*P*) or *non-persistent* (*N*). Persistent entities can be mapped between the two models. For *P* entities we further specify whether its feature interpretation is identical in both models (*Pi*) or different (*Pd*). In our examples, the feature interpretation is limited to two options: positive or negative. Instead of specifying that the interpretation has changed, we can therefore, in this case, directly indicate how the interpretation is different; we mark the entities that are positive in the first model as *Pd1* and those that are positive in the second model as *Pd2*. For the non-persistent entities, it can be indicated whether they, regardless of feature interpretation, exist exclusively in the first model (*N1*) or exclusively in the second model (*N2*).

We emphasise that the difference includes an explicit volumetric description. Volume is treated as a separate, explicit entity in addition to vertices, edges and faces, enabling us to record changes in its feature interpretation, e.g. its nature.

Each entity of either model is categorised from the point of view of each feature that owns the entity. This means that an entity, as part of a specific model, can be categorised more than once and possibly in different ways. Figure 1

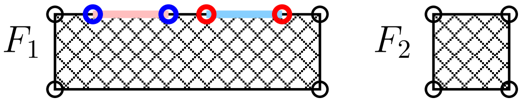

Fig. 2: Feature difference between models in Figure 1.

illustrates this. The vertices on the bottom of *F2* are persistent from the point of view of *F2* and, being located on the model boundary, they maintain their interpretation as positive entities. From the point of view of *F1*, however, the vertices on the bottom of *F2* do not persist between the two models, since they have different locations. Additionally, the light blue and pink coloured edges in Figure 1 represent persistent feature geometry for *F1*, but the interpretation of the model geometry differs for these edges. The light blue edge has a positive interpretation in the left model, but that same part of *F1* in the right model has a negative interpretation; for the pink edge this is the other way around.

Notice that the coloured edges *do exist* in both feature models in Figure 1. This is essential to our perception of geometry in a feature model: all entities that are part of an individual feature, including those that are not on the boundary, are always part of the feature model, and together with the geometry emerging from interactions with other features, they form the complete geometry of the feature model.

Since features can have a different qualification of how geometry changes, we describe the changes from the point of view of each individual feature. Such a point of view only covers the geometry that is owned by the feature or interacts with the feature in question. Thus, when categorizing geometric entities for a particular feature, we handle only those entities inside or on the boundary of that feature.

It should be obvious that in case of identical feature shapes, all the geometry that defines a particular feature is always persistent between two models from the point of view of that feature. Only geometry that emerges from interaction with other features can be non-persistent, since the interaction can change between two models. All persistent geometry can, however, still have a different feature interpretation when compared between two models.

We will now further illustrate the ideas with some examples and discuss how to handle:

- removal or addition of features,
- change in feature shapes,
- change in relative positioning of features.

In our illustrations we use colour to indicate how each entity fits into the difference from the point of view of one particular feature. We refer to this combined qualification for all entities covered by a feature as the *view* on the difference of that feature. Entities that are identical in both models, *Pi*, are indicated with black. Persistent features with different interpretation between the models are coloured light blue (*Pd1*) and pink (*Pd2*). The non-persistent entities are either blue or red, respectively indicating that the feature geometry in our examples can be interpreted as *old* (*N1*) or *new* (*N2*).

Figure 2 shows the two features of the two models in Figure 1 and their view on the difference. The coloured edges show that the interpretation of the feature geometry has changed there; the light blue edge was part of the boundary in the first model and not in the second model, whereas the reverse holds for the pink edge. The coloured vertices in the view of *F1* indicate that they existed only in the first model (the blue ones), or they existed only in the second model (the red ones).

We will now consider the addition of a feature, as illustrated in Figure 3. The corresponding differences are shown in

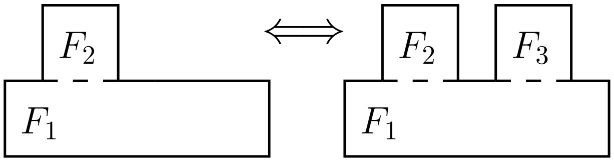

Fig. 3: Changing a model by adding a feature.

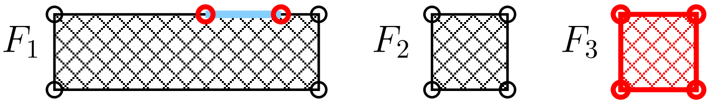

Fig. 4: Feature difference between models in Figure 3.

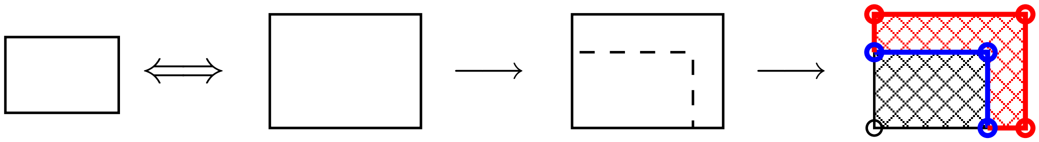

Fig. 5: Difference through self-interaction of different feature parameterisations.

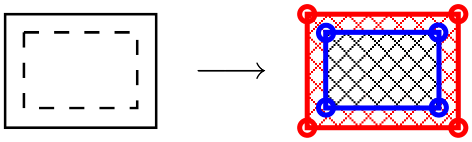

Fig. 6: Alternative difference through self-interaction of feature in Figure 5.

Figure 4. None of the geometry of feature *F3* is persistent, since the feature only exists in one of the models; it is colored red since it is newly present in the second model.

Removal of a feature is essentially the same as addition, only with the order of the two models reversed. If we regard the right model in Figure 3 as the original, from which we get the left model by removal of feature *F3*, then the difference result would be identical to the one presented in Figure 4, except that the red entities of *F3* would be blue and the light blue edge in the view of *F1* would be pink. In general, the feature difference is symmetric. There is no preference for either of the models when describing their difference. Each feature that exists in either of the compared models will thus have to be represented in the difference model.

Another issue we have to deal with is changing the shape of features. We could handle this as a feature removal and addition, but this precludes all feature geometry from being mapped. Instead, we let the two different versions of a feature interact with each other and combine their geometries. In this case, part of the geometry of the feature will not be persistent, since the shape, i.e. geometry, of the feature differs between the two models.

Figure 5 illustrates how we categorise the entities in a self-interaction of two different versions of the same feature. The volume and edge-segments that overlap, together with the single lower left vertex, are considered persistent. All entities that do not overlap with geometry of the same dimension are marked as exclusive to either one of the models.

There is no unique way to compute the interaction of two versions of a feature. The result depends on how the two geometries are overlaid. Typically we would use some internal coordinate system of the feature to align both geometries, but the choice of this coordinate system is arbitrary. In many cases there will be a natural preference for a certain system, but in particular for features with symmetries the choice can be argued. If instead of a corner, the centre of the feature in Figure 5 is used as a point of reference, then the resulting difference will be as in Figure 6.

In some situations, the way the designer modified the feature might indicate a preference for how the different versions ought to self-interact. For instance, he might drag a face along one of the feature axes to elongate the feature, which is a hint that the designer sees the opposite face as a fixed reference. However, this does not apply in general. The shape of a feature might be modified in multiple steps, the feature might be translated, or it might be modified as a consequence of dependency relations. For all these cases, we cannot deduce a single, natural point of reference.

We will now discuss a slightly more involved example that shows how reshaping a feature is handled in the context of a model with multiple features and a change in relative positioning of these features. Figure 7 shows two models built from the same three features. Feature *F2* has been translated and reshaped in the evolution from the left to the right model. Features *F1* and *F2* have positive nature, whereas *F3* has negative nature, which is also the resulting nature in areas of interaction with *F1* and *F2*. In Figure 8 the feature difference between these two models is shown. The difference for *F2* and *F3* has been magnified for clarity. All blue and red coloured entities are again exclusive to either one of the models. This is, as always, caused by a change in interaction with other features or with itself through position or shape change. Light blue and pink are again those entities that are persistent between the models, but have changed in interpretation, either turning from positive to negative or vice versa. Note that the difference for *F2* contains two such areas caused by a changed interaction with *F3*, partially enclosed by respectively a red and a blue arc coming from part of the geometry of *F3*.

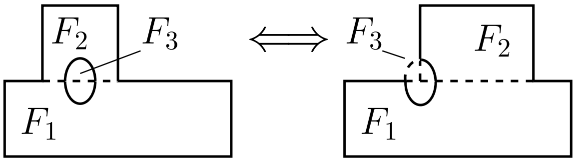

Fig. 7: Example of two versions of a feature models.

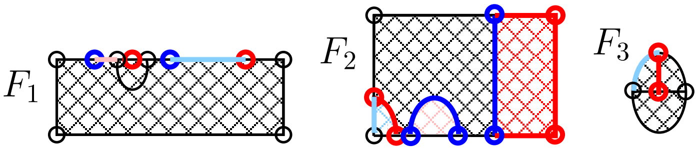

Fig. 8: Feature difference between models in Figure 7.

4. REPRESENTING AND CONSTRUCTING THE DIFFERENCE MODEL

The difference between two feature models is represented in a so-called difference model. This model is conceptually close to the schemes used to illustrate the ideas. It is composed of the views on the difference of all features individually. These views cannot be geometrically combined into a single model, since each view is actually an aggregate of a feature's interactions with two different models.

The data structure of the cellular model can be used to represent the geometry of the view of a single feature on the difference, i.e. each view is represented by a separate cellular model. Its cells arise from interactions in both models with other features, and possibly between different versions of the feature itself. Attached to each entity is its categorisation of the difference: persistent (*P*) or non-persistent (*N*). If the geometric entity is persistent, then additionally it is stored whether its feature interpretation has remained identical (*Pi*) or that it differs between the two models (*Pd*). For the non-persistent entities, it is recorded to which of the two models it belongs. For each entity, the interpretation in the models to which the entity can be mapped, can be accessed. Each entity also has an ownerlist with the features the entity belongs to.

From the difference model of a feature, the contribution of that feature to both feature models can be derived. For efficient use of the model, additional data structures may be used, e.g. to support methods to access all entities of a particular difference categorisation, per feature or globally. The requirements of the data structures will vary per application. In particular we note that, in common applications, it is not necessary to explicitly construct the difference model for features that have an identical shape and interaction with surrounding features in both models. In those cases we just record that, from the point of view of these features, *every* entity is persistent.

One way to construct the difference model is based on the two cellular representations of the feature models under comparison. The difference view of a feature is the result of a non-regular union operation between the cellular complex that covers the shape extent of the feature in the first model and that in the second model. In a non-regular union, all geometry of the combining objects is kept, e.g. faces that are part of another face are not combined into a single face, but remain explicitly available in the description as separate faces. The operation is supported by ACIS [1]. Through application of the non-regular union, the difference attributes ate determined.

The process is illustrated in Figure 9 for feature *F2* from Figure 7. The geometry of the first model is flagged with blue and the geometry of the second model with red, i.e. only the edges are coloured. When the geometries are combined, all vertices, edges, faces and cells are collected in a single geometric description. By default all entities will be marked non-persistent with a reference to the model they are part of. During the combination, the attributes of merging entities of the same dimension are compared and from this follows the difference attribute. All these merging entities, since they overlap with an entity of the same dimension, will be marked persistent (*P*). Additionally, the feature interpretation is compared for the overlapping entities, and thus it is decided whether the meaning is identical (*Pi*) or different (*Pd*). The non-persistent (*N*) entities will not take part in a merge and thus keep the default categorisation, which will give the correct result.

Figure 10 illustrates part of the resulting data structure for four entities in the difference for feature *F1* from the models in Figure 7. If an entity is non-persistent, then the reference to one of the two models being compared, will not exist.

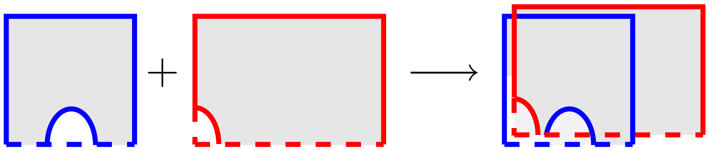

Fig. 9: Merging two cellular complexes in the construction of the difference.

The actual implementation is not as simple as the concept of the operation. In particular the tracking and merging of cells is a reasonably complex problem. The cellular topology component of the ACIS geometric kernel [1] provides basic support to work with cellular models, but propagation and merging of entities, in particular of 3D cells and their attributes, requires substantial effort to make it work correctly and efficiently.

5. POSSIBLE APPLICATIONS

The primary application that prompted us to develop this approach to model differences and similarity between feature models, is the re-meshing of finite element models. Many of the latest methods for constructing tetrahedral meshes are impressively efficient, but the efficiency declines when strict criteria for element shape have to be satisfied. The interest in anisotropic and optimal quality meshes is high. Their construction often depends on the application of iterative optimisation steps, which are more CPU-intensive than the basic algorithms. This builds on the general trend that meshes for analysis are not constructed in one go, but rather, starting from a basic, valid tessellation, are smoothed, refined and otherwise iteratively improved. The work presented in [9] is a nice and recent illustration hereof. There are even indications of this practice of construction through iterative improvements gaining traction in hexahedral mesh generation.

If a previously meshed model is modified, either to increase the level of detail of the model in those areas where it is required, or because the analysis result indicated that the design needed to be adapted, then it will be more efficient to reuse large parts of the previous mesh, than to generate it again from scratch. In particular with large 3D meshes that have specific demands on the element shapes, benefit is to be expected. Also, having identical elements in two analysis models, even if only in some parts, simplifies the comparison of the analysis result in those areas, since the values at the identical nodes can be compared directly.

The difference in terms of features helps to identify the areas where the mesh needs to be adapted. Cells with new volume, need to have nodes and elements added, whereas cells outside the new model can be removed. Nodes that were fixed on a vertex, edge or face that no longer exists as part of a boundary may be moved from their position to further enhance the quality of the mesh. Depending on the details of the approach, not all information in the difference model might be necessary for fulfilling these tasks. In particular for efficiency reasons, one might decide to only construct those parts of the difference model that are actually needed for the algorithm at hand.

Of course, when the geometry of the model changes, then the demands for the mesh can change as well. In particular the required mesh density may change. With the type of changes between models that we have in mind, we do not expect the need for model-wide adaptation of the density, but the change in density requirements does need to be taken into account. Some regions will have to be adapted. These areas can be identified either by comparing the copied mesh with the new global requirements or, alternatively, through the use of analysis features that specify the density requirements throughout the model. When these areas are explicitly part of the model through such features,

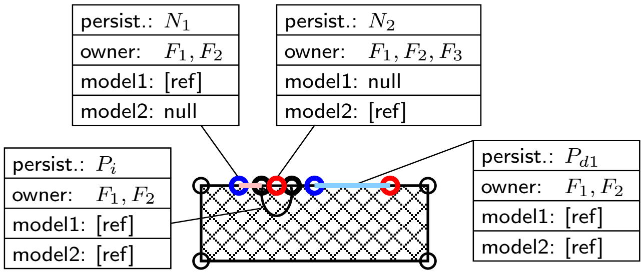

Fig. 10: Description of entities in data structure.

then the difference between the density requirements between the two models can be determined exactly. It needs to be researched how feasible the addition of density requirements through analysis features is.

The details of how parts of a previous mesh can be efficiently mapped and used in the construction of the complete new mesh, are open for further research. When a method such as variational tetrahedral meshing [2] is used, then the mapping comes down to just copying the points. The Delaunay property defines the connectivity. Of course, when the connectivity had been changed to further optimise the mesh, e.g. to remove near-slivers by flipping, this will not be transferred to the new mesh by merely copying the points. In that case, we would need to copy the connectivity too or repeat the final optimisation step.

Since the feature difference is geometrically exact and close to our intuitive idea of difference, because it is in terms of the features that compose the model, we suppose that there might be more applications than just meshing. Any application that has complex data attached to or through specific features, and needs to map this to a similar model, might benefit from using the approach introduced here.

6. CONCLUSIONS

The feature difference between two models categorises, through geometric entities, the complete geometry of both models as similar, thus mappable, or exclusive to either of the models. In addition to the differences in feature geometry, it records the differences in feature interpretation. We have shown how this works out for the case where the interpretation is simply a question of whether the entity contributes to the boundary or, in the case of cells, to the enclosed volume. However, the difference concept extends to a broad class of features where the interpretation might not be mappable to a boolean value. The only requirements are that the features have a clearly defined geometry and that the interpretation resulting from feature interaction is determinate.

Unfortunately it is nearly impossible to illustrate the concept of feature differences with three-dimensional examples that are easily interpretable and add valuable insight. That is why the concepts have been illustrated exclusively by means of two-dimensional examples, but we emphasise that it all readily extends to three-dimensional models. Obviously, to compute geometric interaction in a three-dimensional context is more complex than the two-dimensional case, but there are no fundamental obstacles preventing practical realisation even with complex geometries. The primary difficulty lies with the construction and maintenance of the cellular model, which has been demonstrated to be practically feasible [5], whereas the bulk of the remaining work can be implemented through attributes on the cellular model.

It is probably only possible to consider an exact feature difference between two models, if the features between the models can be mapped. This limits the application to models that are descendants from the same model or a modification of each other. For instance, for feature models wherein the features have been recognised, creating this mapping is likely to be a problem.

The information contained in the feature difference model is virtually unobtainable from a basic BRep structure. In general, any modelling activity that deals with the volumetric interactions of shapes needs a data structure similar to the cellular model. Without such a data structure, everything would need to be calculated on the fly, which for large and complex models will be more costly than maintaining the more complex cellular data structure, and thus needlessly hampering interactive performance.

7. ACKNOWLEDGEMENTS

Matthijs Sypkens Smit's work is supported by The Netherlands Organisation for Scientific Research (NWO).

8. REFERENCES

[1] 3D ACIS Modeler, http://www.spatial.com/products/acis.html, November 2006.
[2] Alliez, P.; Cohen-Steiner, D.; Yvinec, M.; Desbrun, M.: Variational tetrahedral meshing, ACM Transactions on Graphics, 24(3), 2005, 617-625.
[3] Biasotti, S.; Marini, S.; Spagnuolo, M.; Falcidieno, B.: Sub-part correspondence by structural descriptors of 3D shapes, Computer-Aided Design, 38(9), 2006, 1002-1019.
[4] Bespalov, D.; Regli, W. C.; Shokoufandeh, A.: Local feature extraction and matching partial objects, Computer-Aided Design, 38(9), 2006, 1020-1037.
[5] Bidarra, R; de Kraker, K. J.; Bronsvoort, W. F.: Representation and management of feature information in a cellular model, Computer-Aided Design, 30(4), 1998, 301-313.
[6] Bidarra, R.; Bronsvoort, W. F.: Semantic feature modelling, Computer-Aided Design, 32(3), 2000, 201-225.
[7] Bronsvoort, W. F.; Bidarra, R.; Nyirenda, P. J.: Developments in feature modelling, Computer-Aided Design & Applications, 3(5), 2006, 655-664.

[8] Bronsvoort, W. F.; Noort, A.: Multiple-view feature modelling for integral product development, Computer-Aided Design, 36(10), 2004, 929-946.

[9] Dittmer, J. P.; Greg Jensen, C.; Gottschalk, M.; Almy, T.: Mesh optimization using a genetic algorithm to control mesh creation parameters, Computer-Aided Design & Applications, 3(6), 2006, 731-740.

[10] François, V.; Cuillière, J.-C.: 3D automatic remeshing applied to model modification, Computer-Aided Design, 32(7), 2000, 433-444.

[11] Regli, W. C.; Spagnuolo, M.: Introduction to shape similarity detection and search for CAD/CAE applications, Computer-Aided Design, 38(9), 2006, 937-938.

[12] Sheffer, A.; Ungor, A.: Efficient adaptive meshing of parametric models, Journal of Computing and Information Science in Engineering, 1(4), 2001, 366-375.

[13] Shimada, K.: Current trends and issues in automatic mesh generation, Computer-Aided Design & Applications, 3(6), 2006, 741-750.

[14] van der Meiden, H. A.; Bronsvoort, W. F.: Solving topological constraints for declarative families of objects, SPM 2006: Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling, ACM Press, New York, NY, 2006, 63-71.