A Topological Entity Matching Technique for Geometric Parametric Models

Dago Agbodan David Marcheix Guy Pierra Christophe Thabaud

Laboratoire d'Informatique Scientifique et Industrielle (LISI)
Ecole Nationale Supérieure de Mécanique et d'aérotechnique (ENSMA)
Téléport 2 — 1 avenue Clément Ader — BP 40109 — 86961 Futuroscope Chasseneuil cedex — France
(+33/0) 5 49 49 80 63

{agbodan | marcheix | pierra | thabaud}@ensma.fr

Abstract

Nowadays, many commercial CAD systems support history-based, constraint-based and feature-based modeling. Unfortunately, most systems fail during the reevaluation phase when various kind of topological changes occur. This issue is known as "persistent naming" which refers to the problem of identifying entities in an initial parametric model and matching them in the reevaluated model.

We propose in this paper a complete framework for identifying and matching any kind of entities based on their underlying topology. The identifying method is based on the invariant structure of each class of form features and on its topological evolution. The matching method compares the initial and the reevaluated topological histories. For each construction step, the matching consists of two phases. In the local phase, two measures of topological similarity are computed between any couple of entities occurring respectively in the reevaluated model and in the initial model. In the global phase, the final matching is defined as a binary relation that maximizes the topological similarity between the matched entities of both models.

The naming and matching method has been implemented using the 3D modeling application development platform Open Cascade.

Keywords

CAD, Parametric design, Persistent naming.

1. Introduction

Static solid modeling systems (B-rep, CSG, etc.) largely used in the Computer Aided Design (CAD) area are more and more replaced by dynamic modeling systems (known as history-based, constraint-based and feature-based modelers) which allow both to express and to record conceptual designs and "design intents". These dynamic modeling systems are often gathered under the term of parametric modeling systems. A parametric model is composed of a representation of an object, of a set of parameters (characterizing the object) and of a list of constraints (equations or functions) applied to the object. By extension, a parametric modeler is a system for geometric design which preserves not only the explicit geometry of the designed object (called *parametric object* or *current instance*), but also the set of constructive gestures used to design it (called *design process* or *parametric specification*).

This two-fold data structure enables rapid modifying by reevaluation. However, when reevaluation leads to topological modifications, references (between entities) used in the constructive gestures are difficult to match in the new context, giving results different from those expected. A persistent naming system, robust regarding some topological modification, proves useful to preserve, from a reevaluation to another, references between topological entities. It is the problem known as "persistent naming" or "topological naming" [8, 4].

This paper is structured as follows. In section two, we give a detailed account of the major issues about naming in parametric modeling. The third section discusses some pre-existing works, essentially two of the main works about topological naming. Each of these works only partially addresses these issues. We introduce, in section four, an alternative approach.

2. Issues

The main problem for parametric reevaluation is to characterize geometric and topological entities of a parametric model. Characterizing entities consists in giving them a name at design time and "finding them" again at reevaluation time (i.e. matching entities of the initial model with entities of the reevaluated model.)

Let us take the example of Figure 1 to illustrate this problem. In the example below, the initial model is designed by means of a parametric specification containing four successive constructive gestures. The fourth one consists of rounding edge *e*. If the initial model is saved after this fourth step, the current instance no longer contains edge *e*: it was removed by the rounding function. Thus, the rounding function, which has edge e as input parameter, cannot any longer be represented in the parametric specification part of the model. Therefore, "names" are needed to represent entities referenced in the parametric specification whether or not they exist in the current instance.

Moreover each constructive gesture creates a number of entities which have to be distinguished and therefore named, to be referenced by further constructive gestures, even if the same number of entities exist in all possible reevaluation (no topological change). Therefore, each entity should be named in a non-ambiguous and unique way at design time. The problem is even more complex for parametric models of which the entities and the number of entities change from one evaluation to another.

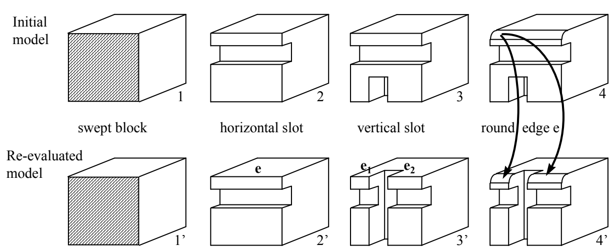


Figure 1: Naming problem

Let us return to the example above, but this time in the reevaluated model. We notice that, at step 3', the edge *e* has been split into edges e_1 and e_2. Thus at step 4' the problem is to determine which edge(s) has(ve) to be rounded. The problem is to identify, i.e. to match, edge *e* with edges e_1 and e_2 despite topology changes. Thus, when reevaluation leads to topology changes a new issue is to match two different structures. The naming mechanism should be powerful enough to perform a robust matching during reevaluation.

3. Related work

Following the pioneer work of Hoffmann and Juan [6], over the last few years several authors have analyzed the internal structure of parametric data models, proposing some editable representations [6, 10, 14, 11, 9], discussing their underlying mathematical structures [10, 12], describing the problems, either of the semantic of modeling operations [6, 5, 1] or of constraint management [3]. Most of them discussed parametric modeling in terms of creation (but not reevaluation). Several naming scheme and persistent naming mechanisms have also been proposed. In particular Kripac [8] and Chen [5] proposed solutions to address some of the problems mentioned in the previous section. The first essentially developed a matching algorithm whereas the second focused rather on the unambiguous entity naming.

3.1 Chen's approach

Chen [5] proposes a model that is composed of two representations. For the first one, he uses an editable representation, called Erep [6], which is an unevaluated, high-level, generative, textual representation, independent of any underlying core modeler. It abstracts the design operations, contains the parametric specification and stores all entities by name. The second representation, evaluated and modeler dependent, contains the geometry (the current instance). The link between these two representations is obtained by a name schema that establishes the link between the geometric entities of the geometric model and the generic names (persistent) of the unevaluated model.

Chen defines a precise structure for naming entities generated by extrusion and revolution operations. Every entity generated by extrusion is named by reference to the corresponding source entity of the extruded 2D contour and the constructive gesture. He also proposes an identification technique for collision-generated entities based on compositions of topological contexts (more or less extended topological neighborhoods) and on feature orientations. Each one of these entities is described by its origin, either a source entity or an intersection of source faces, its smallest unambiguous topological context and its local orientation in the Brep model [4] [5]. To also ensure uniqueness of names in curved domains, one additional information based on geometry is added to the previous topological information : the orientation of any edge against the extrusion direction of the feature(s) it belongs to. The matching of an entity is realized through a local comparison of topological neighborhoods. For example, in case of faces, the face which must be matched is compared with the whole set of faces issued from the same invariant face (preliminary set). At each stage of the construction, the contingent faces inherit the same name of their parent face which makes it possible to construct the preliminary set. A grade is associated to each face of this preliminary set. The grade for each candidate face is the number of matched boundary edges. The face is kept if this number exceeds a threshold.

In his study, Chen restricted to three kinds of features: sweep (extrusion and revolution), blend and fillet. For these features, he showed the feasibility to identify unambiguously any topological entities of models defined by successive attachment of such features, even when faces are curved, in most practical case, i.e., when there are not too many symmetries in the model. A matching algorithm is also proposed that support some level of topological changes in the re-evaluated model. However, it is not clear how the reduced context is used in this algorithm. Moreover this algorithm uses some thresholds, and no precision is given on the choice of these thresholds and the rational for the choices. Finally, the matching algorithm is local to the entity to be retrieved (see 4.2). In case of Figure 4 for instance, and depending on the threshold used, F_2 would probably be mapped onto F_x.

The suggested model represents two major contributions in this domain : on the one hand, two main concepts for topological identification of entities, i.e. topological context and feature orientations which will be used thereafter by many of other approaches, and on the other hand a very precise study of cases of ambiguity.

3.2 Kripac's approach

Kripac [8] focuses on the name matching. He proposes an API (Application Programming Interface) encapsulating its topological identification system and guaranteeing the name persistence using a table of correspondence between an entity of the initial model and one or more entities of the reevaluated model. He proposes an interesting graph structure for identification of any topological entities based on face history (creations, splits, merges and deletions of faces) and a complex name matching algorithm. During each reevaluation, all the faces, as well as every referenced entity in the parametric specification, are matched with the new ones. In addition to the used face graph structure, Kripac's approach is innovative because the proposed matching mechanism is global. The robustness and reliability induced by the global character of the matching method imply an overcost both in spatial (maintaining of two parallel structures) and temporal complexity (more entities to compare). Kripac's model does not allow to record that a selected mapping was only approximated as it uses a discrete metrics. That strongly induces the later mappings and would deserve to be taken into account. Moreover, no explanation is given on the manner of representing and of using this relation in the graphs for the following operations.

His matching algorithm is very sensitive to the subdivision of the topological neighborhood. For example, as illustrated Figure 2, if we call γ_{F_i} the topological neighborhood of face F_i, then the topological neighborhoods of faces F_a et F_b during the model's construction are: $\gamma_{F_a}=\{F_1,F_2,F_3,F_4,F_5,F_6,F_{15},F_{14}\}$ and $\gamma_{F_b}=\{F_7,F_8,F_9,F_{10},F_{11},F_{12},F_{13},F_{16}\}$. During reevaluation, the split leads to new faces F_x and F_y which topological neighborhoods are: $\gamma_{F_x}=\{F_1,F_{15},F_{10},F_{11},F_{12},F_{13},F_{14}\}$ and $\gamma_{F_y}=\{F_2,F_3,F_4,F_5,F_6,F_7,F_8,F_9,F_{16}\}$. The algorithm proposed by Kripac tries to match these new faces with the old ones by analyzing the topological neighborhoods. The analyze consists of finding the longest common face cycle (here $\{F_2,F_3,F_4,F_5,F_6\}$ and $\{F_{10},F_{11},F_{12},F_{13}\}$) in the topological neighborhoods.

Unfortunately, as we can take note on this example, faces F_a and F_b are respectively matched with faces F_y and F_x and not with faces F_x and F_y. A later operation with F_a as input parameter, would have F_y as parameter during reevaluation.

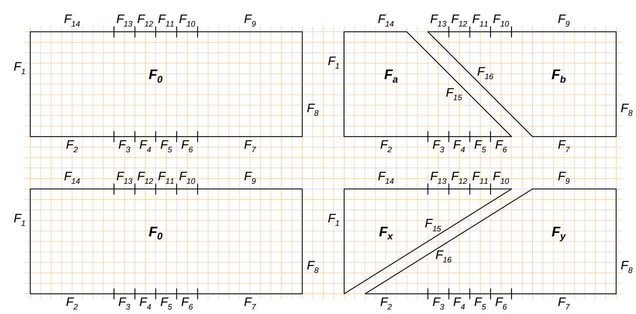


Figure 2 : Top view of slot in a block: construction and reevaluation

Another important problem of this approach is the “piece loss” during reevaluation. The matching algorithm consists on a backward-forward search in the graph and a *cross-analyzis*. More precisely, starting from a given face, a backward search is done in the reevaluated graph, until reaching a face matched with a face of the old graph. Then, starting from this common face, a forward search is done in the old and new graph, processing all branches and retrieving all leaves (faces). A cross-analyzis is done on the faces. The matching between the two faces is done approximately. Therefore, it is possible not to analyze all faces that should be analyzed. Figure 3 illustrates this problem. Matching faces F with T and G with U is done at the reevaluation’s fourth step. The cross-analyzis is done only between faces coming from G and faces coming from U. In particular, in this example, only faces K and L will be crossed with faces X and Y. The algorithm “looses” the face J which can be considered part of face X

Finally, in its approach, Kripac preserves a copy of the geometric models at each step of the construction process. This speeds up the reevaluation but it would require a memory space which is not compatible with the size of the real models used in CAD.

4. Principles of our approach

4.1 Naming model

To define robust names allowing solving the precedent issues we have proposed to distinguish two types of geometric and topological entities [1]:

- *Invariant entities*. An invariant entity is a geometric or topological entity that can be, completely and unambiguously, characterized by the structure of a constructive gesture and its input parameters, independently of involved values. In Figure 1, invariant entities include the end face of the swept block, the lateral shell of the horizontal slot with its begin and end faces (that may, or not exist), the face resulting from the rounding gesture, etc.. To characterize, i.e. to “name”, such entities, information models are to be defined that relate these entities to constructive gestures and to their input parameters.
- *Contingent entities*. Beside those invariant entities, there exist entities that depend on the context of a constructive gesture. We call contingent entity a geometric or topological entity that results from an interaction between the pre-existing geometric model and invariant entities resulting from a particular constructive gesture. For example, in Figure 1, the number of lateral faces of the horizontal slot in the initial model (step 3) and in the reevaluated model (step 3’) are not identical. A naming mechanism is also required to define how to name these contingent entities.

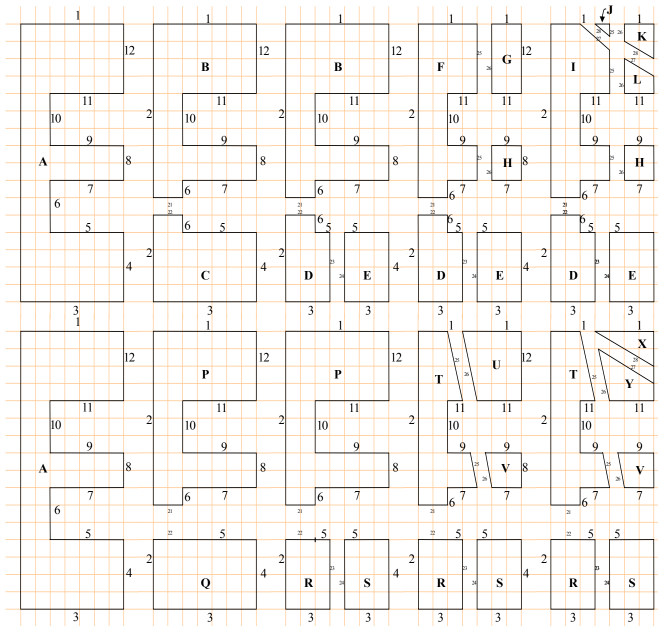


Figure 3: Loss of faces during matching (Is J matched with X ?)

The method we have developed here is based on the model proposed in [1]. This model enables to identify, in a single and non-ambiguous way, firstly invariant entities, then contingent entities according to the invariant entities.

4.1.1 Face graph

The goal is to be able to follow the face evolution in order to be able, during model design, to identify the involved faces, then, during reevaluation, to identify the effective faces (in the current instance) corresponding to the referenced faces.

Figure 4 illustrates a construction example and the associated face graph. Each constructive gesture can be broken up into two steps. The first step is the specification of rough feature. It corresponds to the invariant structure (six faces of the first block). This initial invariant structure represents the inputs of the face graph. The second step is the interaction with the object that produces contingent entities. Those entities result from the evolution of the initial structure. The face evolution is described by *historical links*. In Figure 4 we can see the partial graph structure associated with two slots on a block. In this example, the top face of block is split into two faces by the first slot, then into four faces by the second slot. The face graph represents the history (creation, split, deletion) of the top face. Notice that the initial graph and the reevaluated graph are not the same.

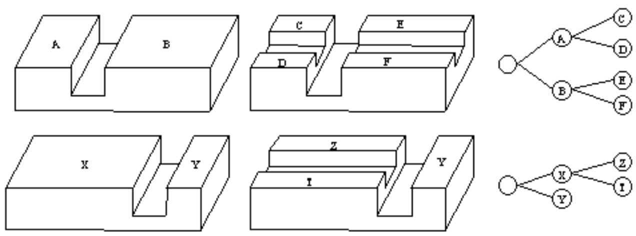


Figure 4 Face graph example. Initial and reevaluated objects and their corresponding face graph (only the top face).

Each face is identified by a unique name which is defined either by a unique topological entities traversal (invariant entities), or by an iterative number (contingent entities) (see [2]). Each node represents a face, which exists or has existed in the model. All the faces without outgoing historical links exist in the geometry.

4.1.2 Entity naming.

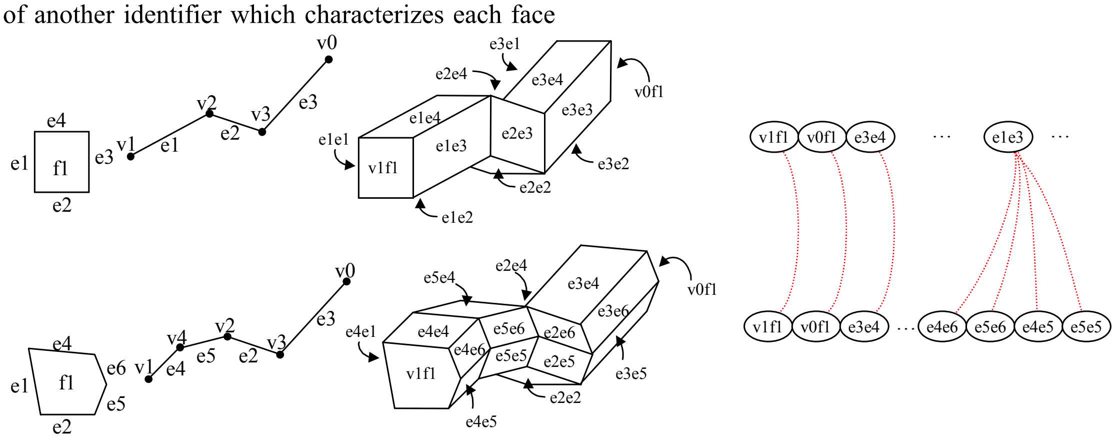
The entity (vertices, edges, paths etc.) identification is done by reference to faces. It is thus necessary to be able to name these faces in a unique and deterministic way. Generally, the identification of an entity is based on unchanging elements that characterize it in a unique way. In a parametric model, what never changes is the construction process (we consider the modification of the construction process as a model edition and not as a model reevaluation). Therefore, face naming is done by means of the construction step number (creation order) and by means of another identifier which characterizes each face in a unique way. The problem is to define this identifier which characterizes them in a unique way within each construction step.

For each construction step, we consider that there are two phases. Firstly, the creation of the feature where all faces must be named. Secondly, the feature positioning within the existing geometry. This interaction with the existing geometry leads to modification and deletion of existing faces and to creation of new (contingent) faces. These contingent faces must also be named. Therefore, there are two types of naming to implement: one for invariant faces and another for contingent faces.

4.1.2.1 Invariant faces

According to the feature taxonomies proposed in [7, 2], invariant faces present in the graph are generated by four type of features (primitive, transition, extrusion and revolution). For the two first cases, the invariant naming of the generated faces are ensured by a unique topological traversal of the object (see [2]).

In an extrusion case, the *generator contour* is swept along a *director path*. Each resulting topological entity corresponds to the cartesian product between a topological entity of the profile and a topological entity of the path. For example in Figure 5, face *e1e4* from the extruded object corresponds to the cartesian product between the director edge *e1* of the director path and the generator edge *e4* of the generator contour. In a similar way, the internal face *v2f1* comes from *v2* (director path) and *f1* (generator contour). Robust naming of each contour entity and of each path entity is fundamental to enable robust naming of faces in the graph. Therefore a matching is done, between contour and path of the initial model and contour and path of the reevaluated model, to ensure name persistence. Each face name is build as follows: ***<step***



Contour	Reevaluation	e1	e2	e5	e6	e4
	Construction	e1	e2	e3	e3	e4

Path	Reevaluation	e4	e5	e2	e3
	Construction	e1	e1	e2	e3

Figure 5: Invariant face naming

number, generator entity, director entity>.

To simplify writing, the step numbers have been omitted in Figure 5. In this example, during reevaluation, the vertex between edge *e1* and *e4* (of the contour) has been moved. This modification, as every geometric modification, has no influence on topological naming of the generator contour, nor the director path, and therefore has no influence on invariant entity naming. However topological modification of the contour (e.g. split of edge *e3*) or the path (e.g. split of edge *e1*) should be traced to ensure robust naming. The matching table shown in Figure 5 permit to store relation between the contour and path of the initial model and the contour and path of the reevaluated model. So, despite geometric deformations and topological subdivisions (edges *e3* and *e1*) of the contour and the path, face *e3e4* is uniquely identified as well in construction as in reevaluation. In a similar way faces *e4e6, e5e6, e4e5* and *e5e5* would be identified as a subdivision of face *e1e3*. We obtain an identification relation between invariant faces of the face graph generated during construction (called **AG**) and those of the face graph generated during reevaluation (called **NG**) (see the end of section 4.2.1.1 for more details).

4.1.2.2 Contingent faces

Contingent face names consist of a step number and an iterative number (an arbitrary but unique number for each construction step). For contingent faces this names is not sufficient to allow an ulterior matching. Therefore, each contingent face in the graph is associated with information about his topological neighborhood. Thus, contingent face names consisting in a step number and an iterative number is sufficient to allow an ulterior matching (see section 4.2).

4.2 Contingent entity matching method

Matching entities consists in associating *n* entities of the initial model with *m* entities of the reevaluated model in order to decide if each of *n* entities corresponds to one or several entities of the reevaluated model, and conversely if each of *m* entities corresponds to one or several entities of the initial model. The matching may be realized by using the geometry and/or the topological neighborhood of entities to be referenced. Topology use allows to get a robust matching method in relation to important geometric variations and small topological variations. However, in some particular cases, when the model contains non-linear entities, topological neighborhood, even extensive [8], are ambiguous and do not allow characterizing in a unique way an entity of this model. Thus, it would be proper to use an additional geometric mechanism (feature orientation, etc.) allowing to raise this ambiguity [5].

Matching quality is very relative and depends generally on operations on the one hand and on semantic the designer wants to express on the other hand. For instance, Face *J* in Figure 3 may be matched onto two different ways according to the semantic given to the operation:

- Either one considers that face *J* is a “part of” face *X* because it has a topological similarity and a common invariant ancestor (face *A*).
- Or one considers that face *J* comes from split of face *F* by the fourth slot. Face *F* is matched onto face *T*, thus *J* can only be matched onto one face of the faces resulting from split of *T*. Consequently, in this example, *J* would not be matched onto any face.

Our approach consists in using the first semantic that turns out to be more general and allows to avoid the tracability loss of a face such as *J*. As we will see it, this tracability loss is highly linked with the type of matching which can be represented in a model. Thus, choosing a too restrictive representation may turn out to be restraining. Indeed, the second semantic is more restrictive. Since it does not take into account the fact that to match an entity with another means that both entities are geometrically and topologically similar but not necessarily identical.

Our approach consists in calculating a matching value for faces present in the graph. Others entities (edge, vertices) are named according to this matching (see section 4.2.2).

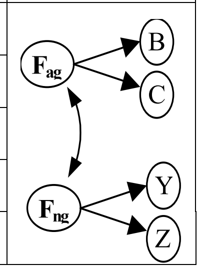
Our face matching method breaks up into two main steps: the *generic cover calculation* which allows to evaluate topological covers between faces of AG and faces of NG (see section 4.2.1.1), and the *real matching calculation* allowing to calculate a specific matching according to the semantic of operation (see section 0). This partition seems to be basic because it allows to distinguish the generic and specific part of matching methods. Such an approach offers numerous interests as for instance the possibility to define a system which proposes a default matching the user will be able to specialize if it doesn’t suit him. Moreover, the cover calculation method is a global method of topological matching between two sets of faces, which may be used in other fields using “pattern matching” as shape recognition, feature recognition and extraction, etc.

4.2.1 Contingent face matching

4.2.1.1 Generic cover calculation

At the reevaluation step, we calculate a cover that consists in evaluating topological matching between *p* faces of AG and *q* faces of NG. Thus, we speak about “crossing” based on each face topological neighborhood. For each face *F*,

Table 2: Inter-graphs relations

Topological neighborhoods	δ_0	δ_1	Graph
$\gamma_{F_{ag}}$ is equal to $\gamma_{F_{ng}}$	1	1	
$\gamma_{F_{ng}}$ is included in $\gamma_{F_{ag}}$	1	]0,1[	
$\gamma_{F_{ag}}$ is included in $\gamma_{F_{ng}}$	]0,1[	1	
$\gamma_{F_{ag}}$ and $\gamma_{F_{ng}}$ partially overlaps	]0,1[	]0,1[	
$\gamma_{F_{ag}}$ and $\gamma_{F_{ng}}$ doesn't overlap	0	0	

we note $\gamma_F=\{o_0,o_1,..o_n\}$ the circuit of oriented edges $(o_i)_{i=0..n}$ of the boundary of *F*. The crossing result is a set of inter-graphs relationships that may exist between faces of AG and faces of NG.

Let $\gamma_{F_{ag}}=\{o_0, o_1,.. o_n\}$ and $\gamma_{F_{ng}}=\{o_0', o_1',.. o_m'\}$ be the circuits associated with faces F_{ag} of AG and F_{ng} of NG. We define $\Gamma_{F_{ag}}$ and $\Gamma_{F_{ng}}$ the sets of the partial sub-paths of $\gamma_{F_{ag}}$ and $\gamma_{F_{ng}}$; a partial sub-path of a circuit is a sub-path of the circuit where some oriented edges have been deleted.

First, one could notice that actually an oriented edge cannot appear in two distinct face circuits in an oriented model. If an oriented edge appears in the circuit of face *F* and the circuit of face *G* then it means that *F* and *G* have opposite orientation: the model is not oriented. So, for each oriented edge *o*, there is a unique face of which circuit uses *o* and we call *neighbor adjacent face* of *o*, the adjacent face of the edge associated with *o* that does not use *o* in its circuit.

In order to quantify topological matching, we define the equivalence relation $\sim_{Adj}$ between face circuits γ and γ', defined by: $\gamma\sim_{Adj}\gamma' \Leftrightarrow \exists(o_i)_{i=0..n}$ and $(o_i')_{i=0..n}$ / $\gamma=o_0.. o_n$, $\gamma'=o_0'.. o_n'$ and $\forall i\in\{0..n\}$, the invariant ancestor face of the neighbor adjacent face of o_i is also the invariant ancestor face of the neighbor adjacent face of o_i'. In other words, when you come along γ and γ' and you consider only the invariant ancestor of neighbor adjacent faces, you get the same circular list of invariant faces around the faces of which circuits are γ and γ'.

Therefore, we can define, $\Gamma_{F\cap G}$ the set of elements of Γ_F that are equivalent to an element of Γ_G according to our relation. This way, $\Gamma_{F\cap G}$ contains all partial sub-paths of γ_F such as there is at least an element of γ_G of which circular list of adjacent faces, in terms of invariant faces, is identical. Then, to solve the problem of subdivisions of topological neighborhood illustrated on Figure 2, we propose to introduce a coefficient allowing to weight each edge influence in the topological neighborhood according to the edge length. Thus, we introduce three functions:

- π such that for each edge *e*, $\pi(e)$ is the length of *e,*
- Π such that for each circuit $\gamma=\{o_0, o_1,.. o_n\}$, $\Pi(\gamma)=\Sigma_{i=0..n}\pi(o_i)$,
- Θ, such that for each element γ of $\Gamma_{F\cap G}$, $\Theta(\gamma)=\max\{\Sigma_{i=0..n} \min(\pi(o_i), \pi(o_i'))$ with $(o_i)_{i=0..n}$ and $(o_i')_{i=0..n}$ / $\gamma=o_0.. o_n$ and $o_0.. o_n\sim_{Adj}o_0'.. o_n'\}$.

$\Theta(\gamma)$ can be interpreted as the maximum common weight between γ and an equivalent element in Γ_G.

Finally, we define $\sigma=\max\{\Theta(\gamma),\gamma\in\Gamma_{F\cap G}\}$.

σ is the maximum sum of edge lengths that we can extract from the boundaries of F_{ag} and F_{ng} such as the edges appear in the same order in the boundaries of F_{ag} and F_{ng}.

We calculate two ratios: $\delta_0=\sigma/\Pi(\gamma_G)$ and $\delta_1=\sigma/\Pi(\gamma_F)$. δ_0 is the ratio of inclusion of $\gamma_{F_{ag}}$ in $\gamma_{F_{ng}}$ and δ_1 is the ratio of inclusion of $\gamma_{F_{ng}}$ in $\gamma_{F_{ag}}$. As shown in Table 2, δ_0 and δ_1 range in interval [0,1] according to the similarity of both weighted topological neighborhoods.

Let us observe example of Figure 2. We have to cross two faces of AG (F_a, F_b) with two faces of NG (F_x, F_y)**.** Table 1 illustrates one calculation step of δ_0 and δ_1.

Previous calculations allow evaluating in an individual way probabilities δ_0 and δ_1 with mutual inclusion of F_{ng} and F_{ag} faces and so topological matching between both faces. This very local approach does not take into account topological matching of adjacent faces. Once δ_0 and δ_1 are calculated, we have to define a method allowing to evaluate in a global way covers between crossed faces. This method consists in handling, in an iterative way, the

Table 1: Crossing

Partial sub-path of circuits of faces F_a and F_y maximizing σ

Weighted topological

Initial graph faces / Reevaluated graph faces	F_a: F_1 F_2 F_3 F_4 F_5 F_6 F_{15} F_{14} — 10 8 2 2 2 2 14.1 6	F_b: F_7 F_8 F_9 F_{10} F_{11} F_{12} F_{13} F_{16} — 8 10 10 2 2 2 2 14.1
F_x: F_1 F_{15} F_{10} F_{11} F_{12} F_{13} F_{14} — 10 18.9 2 2 2 2 8	F_1 F_{15} F_{14} — 10 14.1 6 — $\delta_0=30.1/44.9, \delta_1=30.1/46.1$	F_{10} F_{11} F_{12} F_{13} — 2 2 2 2 — $\delta_0=8/44.9, \delta_1=8/50.1$
F_y: F_{16} F_2 F_3 F_4 F_5 F_6 F_7 F_8 F_9 — 18.9 6 2 2 2 2 10 10 8	F_2 F_3 F_4 F_5 F_6 — 6 2 2 2 2 — $\delta_0=14/60.9, \delta_1=14/46.1$	F_7 F_8 F_9 F_{16} — 8 10 8 14.1 — $\delta_0=40.1/60.9, \delta_1=40.1/50.1$

whole table cells in decreasing order of matching possibilities. For that, we apply the following process:

- Find a cell which is not already "handled" of which sum $\delta_0+\delta_1$ is maximum (if there exist several cells giving this maximum sum, we take any cell of them). Let us suppose that this cell corresponds to the crossing of faces F_{ng} and F_{ag}.
- Decrement edge weights for edges in $\gamma_{F_{ag}}$ and $\gamma_{F_{ng}}$ according to the weight of corresponding oriented edges in the element $o \in \Gamma_{F \cap G}$ that lead to the maximum σ; actually, a temporary weight function replaces π that makes edges appear 'shortened' since some length is no more available for further cell computing.
- For cells which are not already handled, calculate numerators σ of δ_0 and δ_1 with remaining weights.
- Mark this cell as handled
- Iterate the process until all cells are marked.

Note that handling a cell of which coefficients δ_0 and δ_1 are equal to zero does not change anything for the table. Thus, when a cell has both coefficients equal to zero, it can be considered as handled.

Note also that during the handling, coefficients δ_0 and δ_1 only decrease.

Observe the result of this method on example of Figure 3. We can see, on the second step table, that the grayed cell is selected because it is the maximum coefficient sum. Edge coefficients (through a temporary weight function) of γ_X and γ_A are zero because every edge length has been totally used. Coefficients of the row and the column are recalculated. The result is zero because there is no faces which can be used on *X* or *A* to identify other faces. Coefficients being equal to zero, cells are considered as already handled (dashed cells). At third process step, only one cell has to be handled. No computing of coefficients δ_0 and δ_1 is needed because all cells of row and column are handled.

At each construction step, how to know which faces should be used for the crossing. This problem is fundamental because the matching of AG face set with NG one is too expensive on the one hand and rough matching may generate "piece loss" one the other hand, as shown in Figure 3.

Covers allow to know at step *i* of reevaluation, which face of AG and NG have to be crossed. These last ones are defined according to covers obtained at the previous step. At step *i*, faces to be use in a same crossing are leaves of AG and NG appeared up to step *i*. Leave or father faces are connected by *cover links* higher than a given threshold. For that, only covers which appears in NG leaves are necessary to know faces it is advisable to cross. The threshold named $\varepsilon \in [0,1]$ defines the precision of covers. At least one coefficient δ_0 or δ_1 has to be greater than ε in order to represent the inter-graph covering link. A threshold $\varepsilon = 0$ means that all covers are represented and therefore avoid any "piece loss" during the matching. Conversely, a threshold close to $\varepsilon = 1$ means that only covers close to equality will be represented.

Let us observe the evolution of different reevaluation steps in Figure 3. We choose in this example $\varepsilon = 0.15$, which allows eliminating covers which, are not enough significant. The choice of this coefficient depends on the topological matching process accuracy that we want to implement. Initially, at the first reevaluation step, an identification between invariant entities (see section 4.1.2.1) exists and is symbolized by the dotted link between face *A* of AG and of NG (see Figure 6).

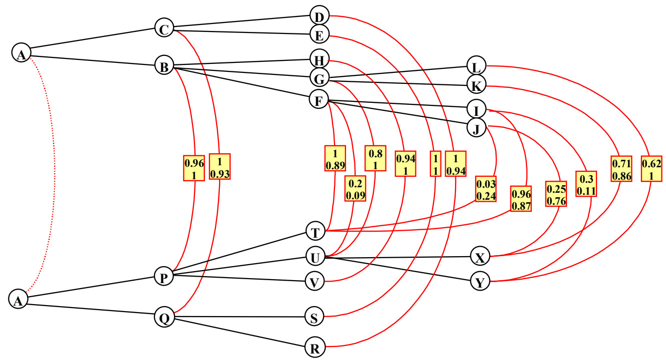


Figure 6: Cover links after reevaluation

At the second reevaluation step, face *A* is split into two new faces *P* and *Q*. Face *A* of NG, father of both faces, is connected by a *cover link* (identification link in this particular case because it is an invariant face) to face *A* of AG of which leaves, appeared up to the second step, are faces *B* and *C*. Both faces *P* and *Q* have to be crossed with faces *B* and *C*. The crossing of both faces gives the following result.

Coefficients δ_0 and δ_1 which correspond to the cover calculation of faces *P* and *C* are less than the threshold $\varepsilon = 0.15$. As result, the cover link between P and *C* is not represented. Only covers which appeared in NG leaves will be necessary to know which faces will be advisable to cross at next step. The link between faces *A* of AG and NG is thus eliminated. Cover link obtained after second reevaluation step are represented in Figure 6 by tagged links between nodes *B*, *C* and *P*, *Q* containing values δ_0 and δ_1.

At the third reevaluation step, face *Q* is split into two new faces *R* and *S*. Face *Q*, father of both faces is connected by a cover link to face *C* of AG of which leaves, appeared up to the third step are faces *D* and *E*. Both faces *R* and *S* have to be crossed with faces *D* and *E*.

Finally, the whole graph, obtained after the fifth reevaluation is shown in Figure 6.

Step 1	B 6 7 8 9 10 11 12 1 2 21 1 5 2 5 3 5 4 7 10 2	C 2 3 4 5 6 22 5 7 4 5 1 2
P 6 7 8 9 10 11 12 1 2 21 2 5 2 5 3 5 4 7 10.7 2	$\delta_0 = {}^{44}\!/_{45.7}$ $\delta_1 = {}^{44}\!/_{44}$	$\delta_0 = {}^{6}\!/_{45.7}$ $\delta_1 = {}^{6}\!/_{24}$
Q 2 3 4 5 22 4.3 7 4 5 2	$\delta_0 = {}^{4.3}\!/_{22.3}$ $\delta_1 = {}^{4.3}\!/_{44}$	$\delta_0 = {}^{22.3}\!/_{22.3}$ $\delta_1 = {}^{22.3}\!/_{24}$

Step 2	B 6 7 8 9 10 11 12 1 2 21 0 0 0 0 0 0 0 0 0 0	C 2 3 4 5 6 22 5 7 4 5 1 2
P 6 7 8 9 10 11 12 1 2 21 1 0 0 0 0 0 0 0 0.7 0	$\delta_0 = {}^{44}\!/_{45.7}$ $\delta_1 = {}^{44}\!/_{44}$	$\delta_0 = {}^{1.7}\!/_{45.7}$ $\delta_1 = {}^{1.7}\!/_{24}$
Q 2 3 4 5 22 4.3 7 4 5 2	$\delta_0 = {}^{0}\!/_{22.3}$ $\delta_1 = {}^{0}\!/_{44}$	$\delta_0 = {}^{22.3}\!/_{22.3}$ $\delta_1 = {}^{22.3}\!/_{24}$

Step 3	B 6 7 8 9 10 11 12 1 2 21 0 0 0 0 0 0 0 0 0 0	C 2 3 4 5 6 22 0.7 0 0 0 0 1 0
P 6 7 8 9 10 11 12 1 2 21 1 0 0 0 0 0 0 0 0.7 0	$\delta_0 = {}^{44}\!/_{45.7}$ $\delta_1 = {}^{44}\!/_{44}$	$\delta_0 = {}^{1.7}\!/_{45.7}$ $\delta_1 = {}^{1.7}\!/_{24}$
Q 2 3 4 5 22 0 0 0 0 0	$\delta_0 = {}^{0}\!/_{22.3}$ $\delta_1 = {}^{0}\!/_{44}$	$\delta_0 = {}^{22.3}\!/_{22.3}$ $\delta_1 = {}^{22.3}\!/_{24}$

Step 4	B 6 7 8 9 10 11 12 1 2 21 0 0 0 0 0 0 0 0 0 0	C 2 3 4 5 6 22 0 0 0 0 0 0
P 6 7 8 9 10 11 12 1 2 21 0 0 0 0 0 0 0 0 0 0	$\delta_0 = {}^{44}\!/_{45.7}$ $\delta_1 = {}^{44}\!/_{44}$	$\delta_0 = {}^{1.7}\!/_{45.7}$ $\delta_1 = {}^{1.7}\!/_{24}$
Q 2 3 4 5 22 0 0 0 0 0	$\delta_0 = {}^{0}\!/_{22.3}$ $\delta_1 = {}^{0}\!/_{44}$	$\delta_0 = {}^{22.3}\!/_{22.3}$ $\delta_1 = {}^{22.3}\!/_{24}$

4.2.1.2 Specific matching calculation

The previous calculation of topological covers is generic in so far as it is only evaluating and quantizing different possible matching while leaving to a method more specific the choice of a particular matching according to some application needs.

Both methods presented in this section are examples of specific matching calculation methods, based on generic covers.

At each reevaluation step, we calculate, according to obtained cover links, the matching of entities generated at this step. If we consider the set of tagged cover links between the old and the new graph present at this step, we get a bipartite graph G={AG, NG, E}. A specific matching corresponds thus to the reconstruction of a bipartite graph G'={AG, NG, E'} where E' is a subset of E and each link (arcs) of E represents a matching relationship between the two nodes.

The method that enables to calculate the links of G which should be kept in G' consists in maximizing the sum of the coefficients δ_0 and δ_1 of the tagged links of E'. Indeed, the more this sum is high, the more whole matching corresponds to exact topological identifications. For that, we assign to each node *i* of G' a coefficient $\delta^i = \sum_{\substack{\text{j= all links} \\ \text{connected to node i}}} \delta_0^j + \delta_1^j$. For the node *i*, this coefficient represents the matching quality of its topological neighborhood. Then, the graph G' giving the maximum sum $\Phi = \sum_{\substack{i=\text{nodes of} \\ \text{the graph } G'}} \delta^i$ corresponds to best realizable matching.

An example of characteristic specific matching is the one where G' is built so that each path length is less or equal one. That means that each entity will be matched with at most one entity.

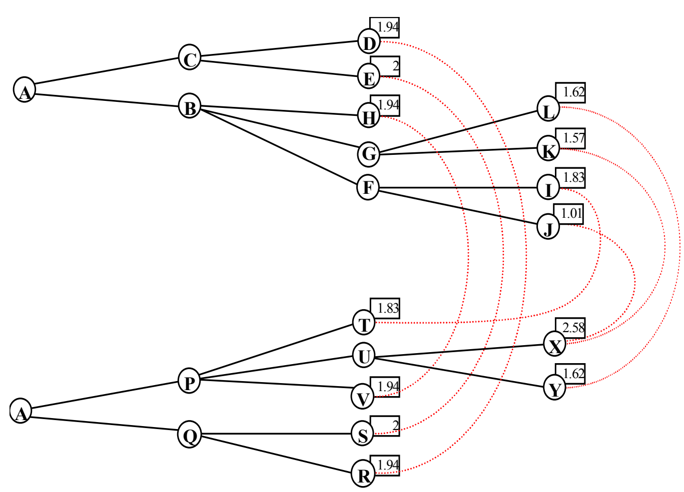


Figure 7: Matching links after reevaluation

Another example of characteristic specific matching is the one where G' is built so that each path length is less than or equal two. From a semantic point of view, that means that one face of AG can either be matched on several faces of the NG, or merged with others faces of AG and matched on a single face of NG. This choice is mutually exclusive. Let us use such a matching on the example of Figure 3. For the nodes appeared in the graph at the last revaluation step, the maximization of Φ enables us to obtain the following G' graph, where the δ^{j} coefficients of each nodes are represented and where the links in dotted lines represent a matching relationship (see Figure 7).

These matching relations are stored in the graph at each revaluation step. We can note that faces *K* and *J* are matched with one same face *X* in the NG.

4.2.2 Other entity matching

The matching of faces being robust, other entities (loops, edges, vertices, etc.) can then be named in term of faces or sets of faces. The characterization of these entities can be carried out in a way similar to the one introduced by Chen [5]. For example, an edge will be characterized by its two adjacent faces, the ordered list of the adjacent faces at its ends, as well as an orientation according to the feature orientation making it possible to raise some topological ambiguities.

5. Conclusion

We proposed a new mechanism of persistent naming associated with a hierarchical structure allowing to trace the historical evolution of easily identifiable invariants in each constructive gesture. The proposed matching method uses weighted topological neighborhoods to characterize in a precise way each entity. It breaks up into two main steps: first the calculation of the *generic covers* allowing to evaluate the topological covers between faces of the old graph (AG) and faces of the new graph (NG), and second, the effective *matching calculation*, allowing to calculate, according to the operation semantics, a specific matching.

We think that this subdivision is fundamental because it makes it possible to distinguish between the generic and the specific part of matching methods. The calculation method of generic covers offers various advantages. First of all, it is a global method of topological matching in the sense that it involves two sets of faces in order to find the best possible matching for all faces. Then, this method makes it possible at each step to know which faces have to be crossed. Finally, it addresses the tracability loss problems which are strongly related to the use of a specific matching.

6. References

[1] Agbodan D., Marcheix D., Pierra G., “A Data Model Architecture For Parametrics”; *Journal for Geometry and Graphics*, 1999, Vol.3, N°.1, pp.17-38.

[2] Agbodan D., Marcheix D., Pierra G., “Persistent Naming for Parametric Models”, *WSCG’2000*, 2000, pp.17-38.

[3] Bouma W., Fudos I., Hoffmann C.M., Cai J., Paige R., “Geometric constraint solver*”, Computer-Aided Design*, June 1995, vol. 27, n° 6, pp 487-501.

[4] Capolylas V., Chen X., Hoffman C.M. “Generic naming in generative, constraint-based design”, *Computer-Aided Design*, Vol. 28 pp. 17-26.

[5] Chen X., “Representation, Evaluation and Editing of Feature-Based and Constraint-Based design.”, *Ph.D. thesis*, Department of Computer Sciences, Purdue University, West Lafayette, Indiana, 1995.

[6] Hoffmann C.M., Juan R., “EREP: an editable high-level representation for geometric design and analysis”, *Technical Report CER-92-24*, Department of Computer Sciences, Purdue University, West Lafayette, Indiana, 1993.

[7] ISO 10303-224: 1999, “Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 224: Application protocol: Mechanical product definition for process planing using machining features”, ISO, Geneva, 1994.

[8] Kripac J., “A mechanism for persistently naming topological entities in history-based parametric solid models (Topological ID System)”, *Proceedings of Solid Modeling’95*, Salt Lake City, Utah USA, 1995, pp.21-30.

[9] Laakko T., Mäntylä M., “Incremental constraint modeling in a feature modeling system”, *Computer Graphics forum*, Vol.15, N°3, EUROGRAPHICS’96, Poitiers, France, 1996, pp.366-376.

[10] Pierra G., Potier J.C., Girard P., “The EBP system: Example Based Programming for parametric design, Workshop on Graphic and Modeling”, *Science and Technology*, Coimbra, Springer Verlag. 27-28, June 1994.

[11] Pierra G., Ait-Ameur Y., Besnard F., Girard P., Potier J.C., “A general framework for parametric product model within STEP and Part Library”, *European Conference Product Data Technology*, London, 18-19 April 1996.

[12] Raghothama S., Shapiro V., “Boundary Representation Variance in Parametric Solid Modeling”, *Report SAL 1997-1, Spatial Automation Laboratory*, University of Wisconsin-Madison, 1997.

[13] Shah J.J., Mäntylä M., “Parametric and feature-based CAD/CAM : Concepts, Techniques, Applications”, John Wiley and Sons Inc., july 1995

[14] Solano L., Brunet P., “Constructive Constraint-based

model for parametric CAD systems", *Computer-Aided Design*, 1994, Vol.26, N°8, pp.614-621.