



Integration of Explicit Geometric Constraints in the Comparison of 3D CAD Models for Part Design Reuse

Antoine Brière-Côté, Louis Rivest, Roland Maranzana

► To cite this version:

Antoine Brière-Côté, Louis Rivest, Roland Maranzana. Integration of Explicit Geometric Constraints in the Comparison of 3D CAD Models for Part Design Reuse. 9th International Conference on Product Lifecycle Management (PLM), Jul 2012, Montreal, QC, Canada. pp.445-457, 10.1007/978-3-642-35758-9_40 . hal-01526130

HAL Id: hal-01526130

<https://inria.hal.science/hal-01526130>

Submitted on 22 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Integration of Explicit Geometric Constraints in the Comparison of 3D CAD Models for Part Design Reuse

Antoine Brière-Côté, Louis Rivest, Roland Maranzana

École de technologie supérieure, Department of Automated Production Engineering,
Montreal, Canada

antoine.briere-cote.1@ens.etsmtl.ca
{louis.rivest,roland.maranzana}@etsmtl.ca

Abstract. Advancements in 3D CAD allow product shape to act as a common language to represent and retrieve reusable product information in PLM systems. While shape-based retrieval techniques contribute to the part design reuse process by retrieving similar part models, selecting the optimal candidate for reuse remains a challenge. A more refined shape comparison process is required to locate single shape differences between reference and candidate CAD models, and to represent them intuitively and functionally in relation to part design. We have developed a 3D CAD model comparison method focused on the representation of shape differences between similar models with respect to the reference's geometric constraints. The proposed method comprises the explicit representation of CAD data, the mapping and differentiation of B-Rep model elements and the re-evaluation of geometric constraints according to shape differences. It will contribute to reliable decision making by promoting part design reuse during the development of new mechanical products.

Keywords: Design reuse, Model comparison, Product shape, Difference identification, Geometric constraints.

1 Introduction

Shape-based product information retrieval has been identified as a promising avenue for the PLM aspect of product reuse, thanks to its potential to lower costs, delays and risk. Product shape can now be used as a common language to represent and retrieve reusable product models in PLM systems, thanks to the capacity of modern 3D CAD systems to provide a reliable and unambiguous representation of the shape of mechanical parts. Shape-based retrieval techniques applied to product classification and information reuse have been proposed and surveyed in the last decade [1]. Commercial 3D shape-based search engines interfacing with current PLM systems are now available (e.g. [2, 3]).

However, like other design retrieval methodologies such as function-based [4] or case-based retrieval [5], shape-based retrieval only contributes to the overall part design reuse problem as an opening step [6, 7]. As pictured in Fig. 1, further analysis of the differences between each retrieved similar model and the reference model is

still required to identify the optimal candidate for design reuse and instruct its adaptation. Moreover, to aptly support the analysis and reinforce the selection, the model difference identification (MDI) solution ought to provide an intuitive and functional representation of model differences relating to the specific application of part design reuse.

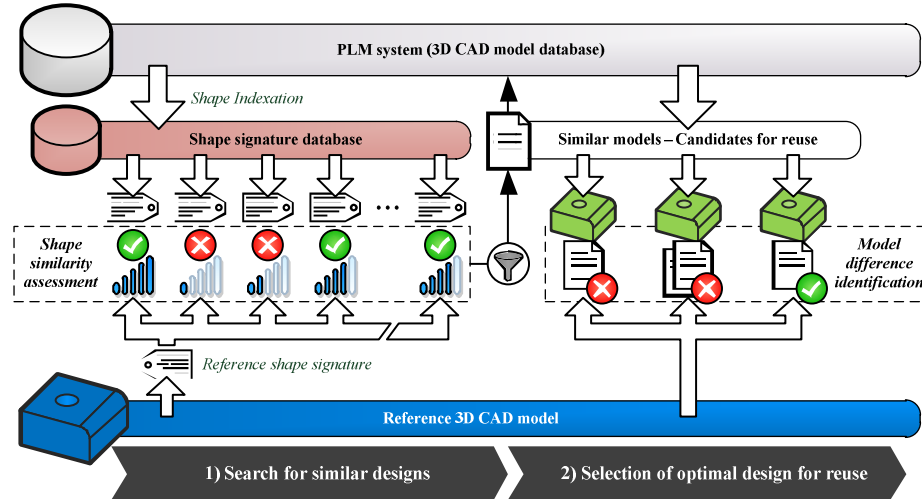


Fig. 1. Search for similar existing part designs and selection of the optimal part design for reuse

This paper presents a pair-wise 3D CAD model comparison technique focusing on the representation of differences between similar part geometries with respect to the reference model's set of specified explicit geometric constraints. Constraint-based difference representation allows precise and local characterization of shape differences. In mechanical product design, geometric constraints are naturally added to a design at the level of abstraction revealed by standard dimensioning practice (e.g. as found in ASME [8]), predictably representing actual form and fit specifications. Representing model comparison findings at the level of abstraction at which designers operate enables them to grasp the rationale behind the shape differences more easily and thus, make quick and reliable decisions towards part design reuse.

The paper is organized as follows. Section 2 provides background on the 3D CAD model difference identification (MDI) problem and briefly examines previous contributions. Section 3 outlines the difference meta-model (DMM) used by the proposed comparison technique. Details of the explicit geometric constraints' re-evaluation phase, performed for a better difference representation, are presented in Section 4. Section 5 presents the application of the proposed technique in the comparison of two 2D CAD sketches as a brief illustrative example.

2 Background

In 3D CAD model comparison, model difference identification (MDI) distinguishes itself from shape similarity assessment mainly in the level of details revealed by the comparison results [7].

Similarity assessment generally exploits highly abstracted or reduced geometric model contents, generally referred to as *shape signatures* [1], to produce quick diagnoses on two shapes' equivalence ("yes" or "no") or relative similarity (a qualitative, scale-based measure). Some signatures were actually found to be reasonably effective at matching human perception of geometric similarity [9]. Examples of abstracted geometric contents include B-Rep models [10] and machining features [11].

However, no details are provided by similarity assessment solutions on what actually distinguishes each similar model from the reference. When calculated recurrently, as in shape-based retrieval and classification, similarity measures attributed to similar models in the resulting set often lose their meaning when interpreted separately.

Conversely, MDI focuses on providing detailed information on what makes two 3D CAD models different. Single shape differences must at least be located with respect to the reference model, but may also be measured and/or characterized some way or another. The models' integrity is preserved as much as possible to prevent the abstraction of relevant differences; however, this renders MDI computationally expensive, limiting it mostly to pair-wise comparison applications.

2.1 Composition of the MDI Problem

Inspired by software model version management [12], we divide the MDI problem for 3D CAD models into three phases. Specific configurations of procedures carrying out these three phases are translated into distinct model comparison techniques:

- *Difference calculation* relates to algorithms establishing mappings between the compared models via specific classes of model elements and identifying the differences between mapped elements, according to some specific properties. For example, pose registration – best-fitting explicit geometries on top of each other in 3D space – establishes a global preliminary mapping between two 3D shapes [13].
- *Difference representation* processes the information from the calculation phase to construct a difference model (Δ) designed for subsequent analysis and manipulation. The difference meta-model (DMM) for a given MDI problem must be aligned with the information requirements of the ongoing reasoning process [14].
- *Difference visualization* renders the difference model in human-readable notation to enable designers to grasp the rationale behind the shape differences. This visualization is communicated in graphical outputs, often combined with graphical interactions, indented lists and/or reports.

2.2 Design-Oriented Difference Representation

A number of 3D CAD model difference calculation techniques were surveyed and detailed in [7]. It has been observed, most notably in commercial software, that 3D CAD model difference representation is often eclipsed by visualization schemes prematurely focused on displaying unrefined geometric calculation results. Little relevant information about shape differences from an engineering design viewpoint is actually provided. Three approaches to design-oriented shape difference representation have been identified and are illustrated in Fig. 2.

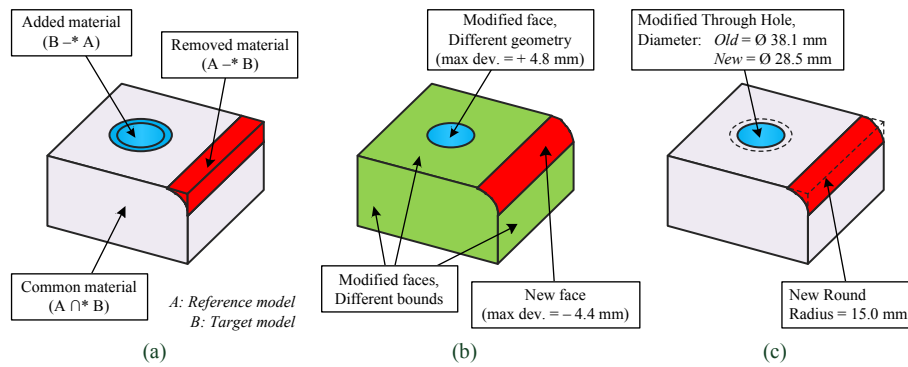


Fig. 2. Three approaches to the design-oriented representation of a given set of shape differences: a) Delta regions; b) Boundary-based; c) Procedural differences.

A first approach leads to difference models that distinguish localized 3D delta regions of material addition, material removal and common material between part geometries (Fig. 2a). This is typically implemented jointly with spatial occupancy difference calculation algorithms and used, for example, in CAD/FEA integration applications such as the remeshing of modified part models (e.g. [15, 16]). Possible delta region measurements include delta volume evaluations and centroids.

Difference representation based on model's boundaries constitutes a second approach. Corresponding difference models commonly classify B-Rep faces and edges from explicit B-Rep models according to their respective differences (Fig. 2b), but these may also aggregate deviating facets between tessellated shapes to form distinct difference regions (e.g. [17]). Boundary-based differences can be calculated explicitly by using geometric algorithms such as point-to-part deviation calculation (e.g. [18]), which may provide additional details in the form of local deviation values, or calculated implicitly by means of B-Rep data structure matching (e.g. [19]).

A third difference representation approach results directly from the comparison of high-level procedural CAD representations. Shape differences are represented in terms of procedural – new, modified, reordered or suppressed modeling operations – and parametric differences between two models' construction histories (Fig. 2c). Design semantics associated with 3D CAD modeling operations, commonly depicted as parameterized form features, allow for a wide range of design-oriented difference measurements in the form of modified blend radii, resized and/or relocated holes, etc.

Among these three 3D CAD model difference representation approaches, only the procedural approach relates to a mechanical part's form and fit specifications at the level of abstraction at which designers naturally operate, an aspect that is highly relevant in assessing a part design's reuse potential. Then again, procedural CAD representations and corresponding difference models will not necessarily correlate with an actual part's geometric specifications; i.e. differences in construction histories do not systematically translate into differences at the resulting shape level.

At the same time, the scope of procedural CAD model comparison is strictly limited to the comparison of a model's own versions since the procedural representation of solids is highly variable. In applications where compared models are inherently unrelated, like when similar parts are collected via shape-based retrieval, the calculation and representation of shape differences with respect to construction histories is obviously considered unviable.

3 Difference Meta-Model (DMM)

Our solution aims to combine the flexibility of explicit model difference calculation techniques with the intuitiveness of the design-oriented procedural difference representation approach. The difference visualization phase of MDI has been left out of the scope of this paper: an efficient difference representation is considered to be at the basis of a good visualization scheme.

3.1 Application Setting

As part of the part design reuse process described in section 1, the MDI problem presents the following characteristics for the compared models:

- They are considered as detached, i.e. no pre-established relation is available between the models or their elements, except that their shapes have been found to be similar as a result of the preceding shape-based retrieval phase; and
- They may be expressed in different CAD formats, raising concerns relating to CAD interoperability.

We have therefore forward the following five statements to ensure that the proposed comparison technique will be practical for the problem at hand:

1. The shape-based retrieval solution applied previously develops pairs of 3D CAD models similar enough to enable the identification of discernible shape differences.
2. Shapes are represented explicitly via the B-Rep paradigm and convey accurate geometric information (e.g. no planar facet tessellation).
3. The reference model includes a set of explicit geometric constraints relating directly to the shape's B-Rep. Constraints should preferably represent actual form and fit specifications.
4. The set of geometric constraints is resolved and conflict-free. Over-constrained shapes are not allowed.

5. All geometric constraints are directed, i.e. they assert relationships between sets of constrained elements and one or more specified reference elements [20].

3.2 Explicit Representation

For flexibility reasons, and to counter CAD interoperability concerns, difference calculation is better achieved at the shape level, i.e. discrete mappings between models will be determined via their shape elements (described in section 3.3). To exploit geometric constraints in addition to shape differences for better difference representation, the distinction between constrained shape elements and constraints must then be preserved. An explicit or declarative approach to CAD data representation, allowing model elements to be referenced and manipulated individually, is required.

To represent geometric constraints explicitly, B-Rep shape elements – geometric and topological – must be declared first, and then constraints between these elements can be declared (e.g. as in ISO STEP Part 108 [20]). Geometric constraints specify relationships between shape elements and between shape elements and design parameters (i.e. logical and dimensional constraints, respectively), as pictured in Fig. 3.

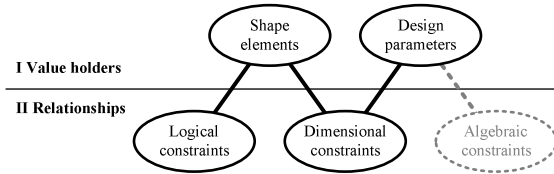


Fig. 3. The relationships between shape elements, design parameters and constraints¹ (Inspired by Bettig and Shah [21]).

Access to explicit geometric constraint data from 3D CAD models is feasible, but remains a challenge when traditional parametric feature-based models are involved. While 3D direct or “history-free” modeling and regular 2D sketch creation lead to the explicit representations of shape and geometric constraints, most 3D geometric constraints are implicit to parametric feature definitions and, therefore, cannot be referenced individually.

For the scope of this paper, we opt not to extract any implicit or undeclared information from 3D CAD models. Adding information not originally conveyed by the models, such as implicit constraints or via feature recognition, is considered as altering their content’s integrity and biasing the comparison. The matter of rendering parametric feature-based 3D CAD models in explicit form will be addressed in future work.

Explicit representation of compared 3D CAD data and subsequent difference modeling is achieved based on the approach described by Cicchetti et al. [22]. Schematized in Fig. 4, the approach is meta-model-independent, which allows us to define an explicit 3D CAD meta-model (MM) for referencing compared data while preserving

¹ Algebraic constraints, or “design rules”, specify relationships between design parameters.

the original models' integrity. The specified MM then relates to the DMM through extension. Difference models systematically include an image of the compared models for functional difference representation.

3.3 Shape Difference Calculation

For the proposed 3D CAD model comparison technique, we do not impose the use of one particular shape difference calculation algorithm. Instead, the focus is primarily on difference representation. In addition to good difference calculation precision and recall [9], a suitable algorithm must at least meet these requirements:

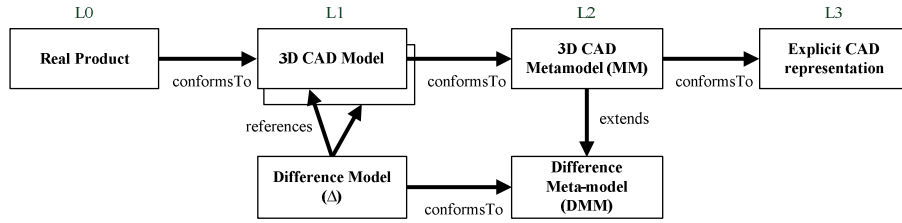


Fig. 4. Overall structure of the model difference representation approach

- It must operate on detached models, and therefore, static identity-based matching algorithms [7] are not applicable;
- It must operate on B-Rep data and produce face, edge and vertex mappings; and
- In addition to unique (unmapped) and equivalent (mapped) shape elements, it must be able to recognize “modified” shape elements, i.e. different elements that can still be mapped between the two shapes.

Shape mappings are to be recorded in a new difference model instance that conforms to the DMM presented in Fig. 5 in UML. When matched, model elements from either the reference or the target models are referenced accordingly to maintain the comparison directionality at a lower level of granularity. The proposed DMM accepts n-ary mappings and maintains parent/child relationships between mappings inferred from one another (e.g. mapped faces leading to their respective geometry’s mapping).

As an example, the B-Rep difference calculation algorithm described by CoCreate Software [19] and implemented in PTC CoCreate® Modeling PE [23] would constitute a suitable option for the proposed technique. It uses a syntax-specific matching algorithm that recursively produces vertex, edge and face mappings. Topological elements are then classified as equivalent, affected (relimited), geometrically different, or as found only in the reference or in the target model.

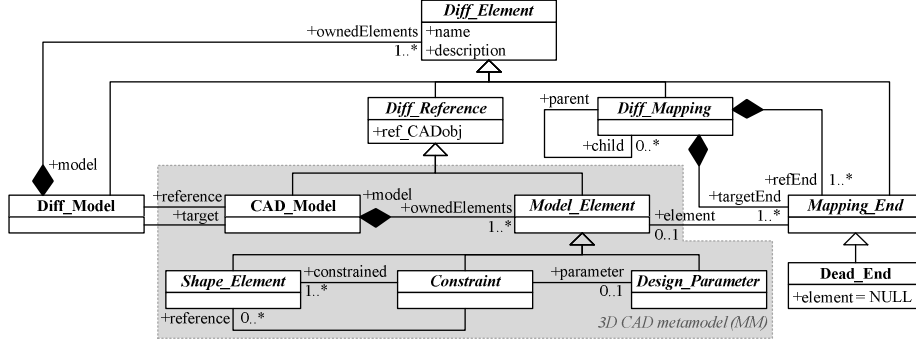


Fig. 5. Difference meta-model (DMM)

4 Explicit Geometric Constraints Re-Evaluation

Shape difference calculation provides an initial boundary-based representation of the differences between models. To elevate the level of abstraction of the difference model at the designer’s level, the reference model’s explicit geometric constraints are transposed on the target model’s shape via the mappings established earlier. Their validity and conformity is then re-evaluated with respect to the target shape. Design parameters related to dimensional constraints are also re-evaluated.

The concept is exposed in Fig. 6 with an example. In the reference model (left), a “parallel-distance” geometric constraint (e.g. *pgc_with_dimension* entity from ISO STEP Part 108 [20]) is specified between two faces *A* and *B* and leads to the definition of design parameter *d*. When compared to the target model (right), face mappings are established between faces *A* and *A'*, and between faces *B* and *B'*. Since both related elements are mapped, a similar parallel-distance constraint is projected on the target shape. The parallelism of faces *A'* and *B'* is then validated and the target design parameter *d'* is re-evaluated, identifying a difference with respect to the reference design parameter *d*.

4.1 Pre-processing of Explicit Geometric Constraints

The representation of explicit geometric constraints in the DMM implies that a trade-off must be established between practicality and functionality concerns. First, difference models must reference individually the geometric constraints and their variations currently used by most 3D CAD systems while minimizing original data alteration. Conversely, the geometric constraint re-evaluation phase calls for a comprehensive, yet constant and consistent set of geometric constraint types for easier manipulation. For example, concrete geometric constraint types as defined by ISO STEP Part 108 [20] form a practical set of geometric constraints, while the theoretical set derived by Bettig and Shah [21] is considered to be consistent.

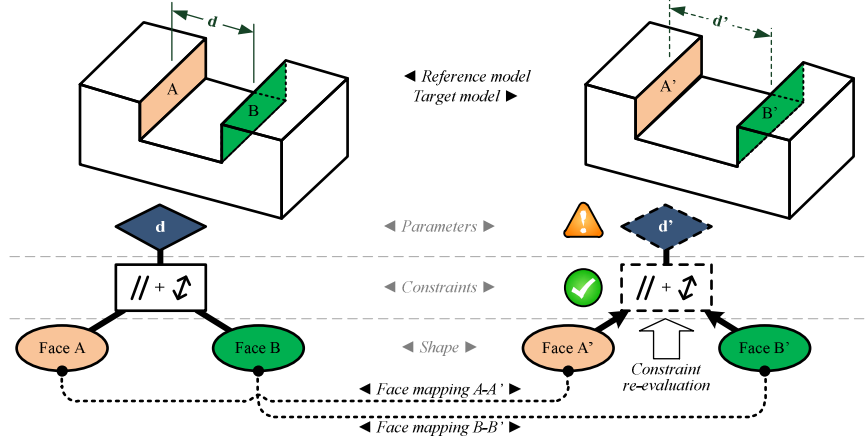


Fig. 6. Re-evaluation of geometric constraints and related design parameters on target model based on shape element mappings.

A satisfying trade-off is achieved in the DMM by generalizing all the currently-used geometric constraints while distinguishing composite from elementary constraints, as shown in Fig. 7. The purpose of composite constraints in the DMM is essentially to refer to original 3D CAD geometric constraints that may define multiple elementary relations between shape elements – such as the “parallel distance” constraint from Fig. 6 – and/or collate many shape element tuples via a single instance. Composite constraints are then decomposed into elementary constraints, each defining a single geometric relation among a minimal number of shape elements (corresponding to the relation’s arity). Only elementary geometric constraints are subject to re-evaluation; located differences are then represented with respect to the corresponding composite constraints.

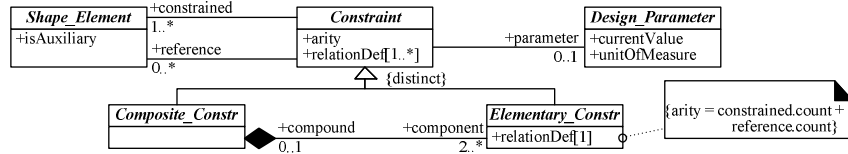


Fig. 7. Representation of composite and elementary geometric constraints in the DMM

4.2 Boundary and Auxiliary Shape Elements

In 3D CAD models, explicit geometric constraints are recurrently combined with auxiliary shape elements. These are elements that do not make up the solid’s boundary, but which still participate in its definition together with geometric constraints. Reference geometry or datums used in 3D space and construction geometry used in 2D sketches are all examples of 3D CAD objects referenced as auxiliary shape elements in the DMM.

Without dependable counterparts in the target model, auxiliary shape elements from the reference model are excluded from the shape calculation phase. Therefore, geometric constraints relating to auxiliary shape elements cannot all be re-evaluated, as compared to boundary shape elements for which mappings can be readily examined. Figure 8 describes how shape element mappings are ultimately processed to elevate shape differences at the level of geometric constraints and design parameters.

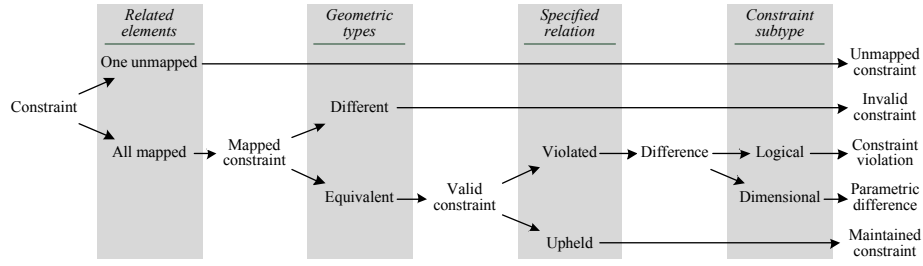


Fig. 8. Shape element mappings and differences elevated at the level of geometric constraints

4.3 Generating Auxiliary Shape Elements in the Target Model

Auxiliary shape elements are handled similarly to intermediate variables in function composition problems. For example, given auxiliary shape element a of geometric type A from the reference model, a is originally related to shape element x of geometric type X via explicit geometric constraint f , and to shape element y of geometric type Y via explicit geometric constraint g .

Given that all geometric constraints are directed, and $f: X \rightarrow A$, $g: A \rightarrow Y$, then:

$$g \circ f = \{(x, y) \in X \times Y | \exists a \in A : (x, a) \in f \wedge (a, y) \in g\}. \quad (1)$$

Transposed in the target model, transient auxiliary shape element $a' \in A$ can be calculated depending on the existence of shape elements $x' \in X$ and $y' \in Y$. The existence of such elements in the target model is determined either via boundary shape element mappings or by the recurring calculation of other transient auxiliary shape elements. Specific geometric constraints must be deliberately enforced in the target model in order to evaluate the auxiliary elements they relate to. These constraints are thereby withdrawn from the re-evaluation process.

Modification of a directed constraint systematically impacts its constrained elements. Also, auxiliary shape elements act mainly as reference elements in geometric constraint schemas. Since we consider expressing differences with respect to the target shape's boundary to be more relevant, we choose to enforce the explicit geometric constraints for which auxiliary shape elements are specified as constrained elements (e.g. the function f in Eq.(1)).

However, one exception to this enforcing rule is upheld in cases where enforced geometric constraints are dimensional constraints. When possible, difference is evaluated with respect to design parameters, which is considered more intuitive from a design viewpoint. For example, in simple cases such as that expressed by Eq.(1),

where f and g are dimensional and logical constraints, respectively, resolution will amount to solving the inverse expression ($f \circ g$) and enforcing g instead of f .

5 Example: 2D Sketches

This section presents the proposed model comparison technique through an illustrative example. For clarity, it is applied to the comparison of two 2D constrained sketches, as presented in Fig. 9; nonetheless, the concepts illustrated here fully apply to the comparison of 3D shapes. The left side of Fig. 9 displays the reference sketch with labeled shape elements and explicit geometric constraints, while the right side displays the target sketch complemented with comparison results. Reference and target sketches are also presented according to two different levels of abstraction: a shape level (top) and a geometric constraint level (bottom).

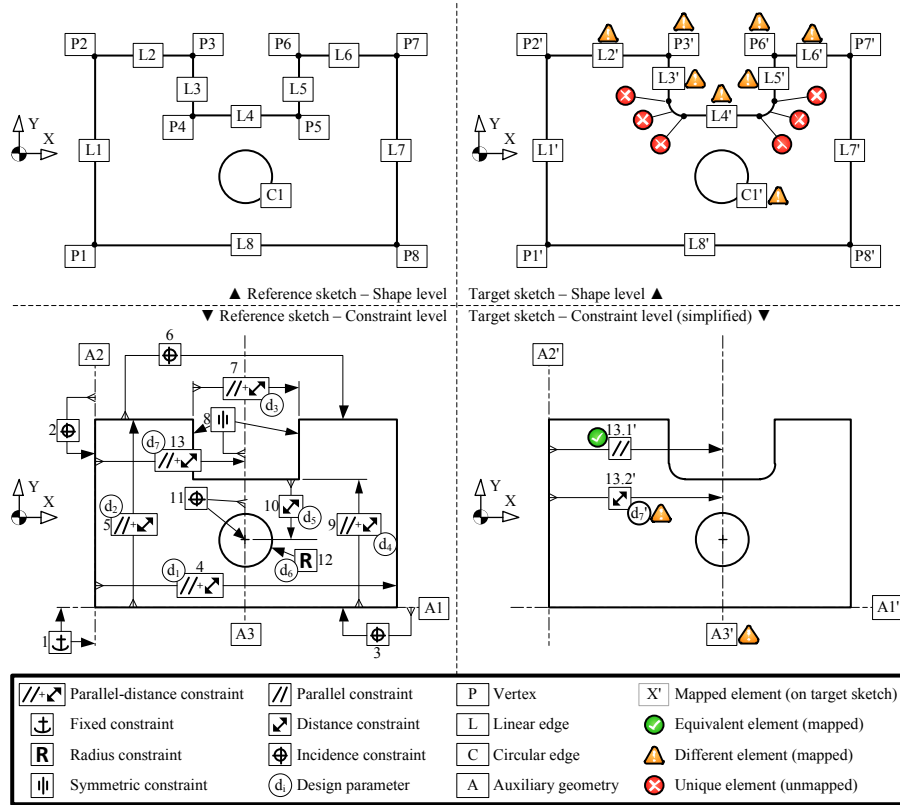


Fig. 9. Results from the proposed model comparison technique applied to two 2D sketches

The differences between the reference and the target sketches include the circular hole and upper notch being moved jointly to the right side, as well as the notch's bottom corners being rounded. At the shape level, the target sketch is annotated with results

of the shape calculation phase. Accordingly, the entire notch region and the circular edge for the hole are tagged as being either different or unique. If difference representation is to be useful in a context of design reuse, such results need refining.

Re-evaluation of explicit geometric constraints from the reference sketch on the target sketch leads to a more intuitive representation of shape differences, displayed in the lower right corner of Fig. 9. Only one of the original thirteen geometric constraints is displayed (divided into two elementary constraints), since all twelve others were found to be fully maintained. Design parameter d_7' , which is related to a parallel-distance constraint, is identified as the key source of difference between the two sketches.

This example reveals a limitation of the geometric constraint re-evaluation phase: the newly rounded corners were not identified as differences at the constraint level. New boundary shape elements in the target model will systematically be overlooked, because no shape element or geometric constraint from the reference model can relate to them. One must therefore go back to the shape level representation to collect all the relevant details about the located differences. Complete difference representation must be achieved concurrently at both the shape and the constraint levels.

6 Conclusion

One aspect of PLM stresses that any product lifecycle contributor should always have easy access to the information they need for the realization of their task. In the specific context of part design reuse, the 3D CAD model comparison technique proposed in this paper specifically aims at providing designers with an intuitive and functional representation of model differences identified between a reference and similar candidate models collected via shape-based retrieval.

Shape differences calculated at the B-Rep level by current algorithms can now be expressed at the higher level of geometric constraints and design parameters, i.e. the level of abstraction at which designers naturally define a product's form and fit specifications. Explicit geometric constraints specified in the reference 3D CAD model are transposed with respect to the target model's shape and re-evaluated accordingly. This process ultimately leads to comparison findings expressed in terms of, for example, constraints that are maintained or violated and parametric differences, instead of as old, new or modified faces and edges. The new technique should therefore provide more reliable assistance to designers in the assessment and selection of parts for design reuse.

References

1. Cardone, A., Gupta, S.K., Karnik, M.: A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering* 3, 109-118 (2003)
2. 3DSemantix inc., 3DPartFinder: Geometric Search Engine, <http://www.3dpartfinder.com/en/>

3. Siemens PLM Software inc., Geolus Search, http://www.plm.automation.siemens.com/en_us/products/open/geolus/index.shtml
4. Xu, Q.L., Ong, S.K., Nee, A.Y.C.: Function-based design synthesis approach to design reuse. *Research in Engineering Design* 17, 27-44 (2006)
5. Watson, I., Perera, S.: Case-based design: A review and analysis of building design applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM* 11, 59-87 (1997)
6. Msaaf, O., Maranzana, R., Rivest, L.: Part data mining for information re-use in a PLM context. *Proceedings of GT2007. American Society of Mechanical Engineers, New York, NY* 10016-5990, United States (2007)
7. Brière-Côté, A., Rivest, L., Maranzana, R.: Comparing 3D CAD models: Uses, Methods, Tools and Perspectives (Accepted). *Computer-Aided Design and Applications* (2012)
8. ASME: Dimensioning and Tolerancing. *ASME Y14.5-2009. American Society of Mechanical Engineers, New York, NY* (2009)
9. Clark, D.E.R., Corney, J.R., Mill, F., Rea, H.J., Sherlock, A., Taylor, N.K.: Benchmarking shape signatures against human perceptions of geometric similarity. *Computer-Aided Design* 38, 1038-1051 (2006)
10. McWherter, D., Peabody, M., Regli, W.C., Shokoufandeh, A.: Solid model databases: Techniques and empirical results. *Transactions of the ASME. Journal of Computing and Information Science in Engineering* 1, 300-310 (2001)
11. Cicirello, V., Regli, W.C.: Machining feature-based comparisons of mechanical parts. *Proceedings International Conference on Shape Modeling and Applications*, pp. 176-185. *IEEE Computer Society* (2001)
12. Kolovos, D.S., Di Ruscio, D., Pierantonio, A., Paige, R.F.: Different models for model matching: an analysis of approaches to support model differencing. In: *2009 ICSE Workshop on Comparison and Versioning of Software Models (CVSM 2009)*, 17 May 2009, pp. 1-6. *IEEE*, (Year)
13. Yang, Y., Lin, H., Zhang, Y.: Content-based 3-D model retrieval: A survey. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 37, 1081-1098 (2007)
14. Brière-Côté, A., Rivest, L., Maranzana, R.: A Three-Step Approach to Structuring 3D CAD Model Comparison Scenarios. In: *IFIP (ed.) IFIP WG5.1 8th International Conference on Product Lifecycle Management. Inderscience Enterprises Ltd* (2011)
15. François, V., Cuillière, J.C.: 3D automatic remeshing applied to model modification. *Computer-Aided Design* 32, 433-444 (2000)
16. Sypkens Smit, M., Bronsvort, W.F.: Efficient tetrahedral remeshing of feature models for finite element analysis. *Engineering with Computers* 25, 327-344 (2009)
17. Lattice Technology Co.: *XVL Studio Professional*. Lattice Technology inc., San Fransisco, CA (USA) (2010)
18. CapVidia NV: *CompareVidia*. CapVidia NV, Leuven (Belgium) (2010)
19. CoCreate Software GmbH, Gutierrez, S.: Method for comparing a first computer-aided 3D model with a second computer-aided 3D model. In: *Office, U.S.P.a.T. (ed.), vol. A1*, pp. 11. CoCreate Software GmbH, U.S. (June 3, 2010)
20. International Organisation for Standardization: *Industrial automation systems and integration -- Product data representation and exchange -- Part 108: Integrated application resource: Parameterization and constraints for explicit geometric product models*. *ISO 10303-108:2005*. International Organisation for Standardization, Geneva (2005)
21. Bettig, B., Shah, J.: Derivation of a standard set of geometric constraints for parametric modeling and data exchange. *Computer-Aided Design* 33, 17-33 (2001)

22. Cicchetti, A., Di Ruscio, D., Pierantonio, A.: Managing model conflicts in distributed development. In: Model Driven Engineering Languages and Systems. 11th International Conference, MoDELS 2008, 28 Sept.-3 Oct. 2008, pp. 311-325. Springer-Verlag, (Year)
23. CoCreate Software GmbH: CoCreate Modeling : Personal Edition 2.0. Parametric Technology GmbH, Unterschleissheim (Germany) (2008)