TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

2D	Bidimensionnel *Two-dimensional*
3D	Tridimensionnel *Three-dimensional*
API	*Application programming interface*
ASE	*Auxiliary shape element*
B-Rep	*Boundary representation*
CAD	*Computer-aided design*
CAM	*Computer-aided manufacturing*
CAO	Conception assistée par ordinateur
CAx	*Computer-aided systems*
CGE	Contrainte géométrique explicite
COPS	*Cloud of points*
CSG	*Constructive solid geometry*
DMM	*Difference meta-model*
DMU	*Digital mock-up*
EC	*Evaluation criterion*
ECM	*Engineering change management*
ECO	*Engineering change order*
EGC	*Explicit geometric constraint*
FEA	*Finite element analysis*
GVP	*Geometric validation properties*

MBD	*Model-based definition*
MDI	*Model difference identification*
MDSE	*Model-driven software engineering*
MM	Méta-modèle *Meta-model*
MOF	*Meta-object facility*
MPP	*Machining process planning*
NC	*Numerical control*
NURBS	*Non-uniform rational B-spline*
OEM	*Original equipment manufacturer*
PCA	*Principle component analysis*
PDMF4C	*Product, definition, model and formalism for comparison*
PDP	*Product development process*
PDQ	*Product data quality*
PLM	*Product lifecycle management*
PMI	*Product manufacturing information*
PSE	*Physical shape element*
TSE	*Target shape element*
UML	*Unified modeling language*

LISTE DES SYMBOLES ET UNITÉS DE MESURE

Δ	Delta *Difference model*

INTRODUCTION

Aujourd’hui, le développement de produits complexes s’appuie invariablement sur l’exploitation d’outils logiciels performants. Grâce aux avancées dans le domaine de la conception assistée par ordinateur (CAO), la modélisation géométrique tridimensionnelle des composants mécaniques constitue dorénavant une activité indispensable à la conception. Les outils de CAO permettent une représentation fidèle et précise de la forme d’un produit sur laquelle se basent et à laquelle contribuent une multitude d’autres disciplines, autant en conception que lors des processus subséquents.

Représentée au sein des nombreux modèles, la définition géométrique d’un produit évolue continuellement par le travail de nombreux spécialistes, parfois délocalisés, qui en réutilisent, analysent et modifient les informations, prennent des décisions et engagent des actions à partir de celles-ci. Conséquemment, les données géométriques sont échangées entre individus et entre unités d’affaires très fréquemment. De plus, la complexité et la richesse des modèles géométriques poussent les entreprises à conserver et réutiliser les données décrivant des produits fiables et éprouvés, et ainsi capitaliser sur les connaissances acquises par le passé.

Lorsque la définition géométrique évolue, i.e., des modifications sont apportées à la forme du produit, différents modèles utilisés et maintenus par les spécialistes doivent évoluer de manière cohérente. Cette évolution impactera du même coup les données techniques propres à chaque discipline, dites *expertes*, associées à ces différentes représentations de la géométrie, qui évolueront en conséquence. La transposition d’une modification géométrique d’un modèle expert à un autre doit préserver la cohérence de leurs représentations géométriques respectives. Cette transposition doit donc se baser sur une localisation précise et une représentation adéquate des différences géométriques entre le modèle de référence évolué et le modèle ciblé par la modification. Cette thèse propose une approche innovatrice de comparaison géométrique et un formalisme adapté permettant de capturer et de communiquer intelligiblement l’intention de conception derrière une évolution de la définition géométrique et ce dès les premières phases du cycle de vie du produit.

La complexité de la problématique de la transposition du changement entre modèles experts, abordée par cette thèse, réside dans la production de résultats de comparaison géométrique revêtant une signification fonctionnelle pour la discipline, ou le spécialiste, appelée à mettre à jour son modèle à partir d'une version évoluée de la géométrie. Par exemple, un ingénieur des méthodes sera souvent appelé à modifier son modèle détaillant notamment les opérations et les cotes de fabrication d'un composant à partir d'une représentation de la forme évoluée provenant du bureau d'études. Afin de le supporter efficacement dans sa tâche, la représentation des différences identifiées entre son modèle et la nouvelle géométrie doit en faciliter l'interprétation et l'assimilation de son point de vue d'expert en méthodes et en lien avec son modèle à modifier.

Cette thèse, présentée sous la forme d'une thèse par articles scientifiques, se compose de neuf chapitres, dont six présentent des articles de conférences et des articles de journaux qui font l'objet de publications.

Le premier chapitre présente le contexte de l'évolution de la définition du produit et y situe la problématique de la transposition du changement et ses enjeux. Le rôle que peuvent jouer la comparaison géométrique et la représentation des différences dans la propagation du changement à travers le modèle du produit y sont également mis en évidence. Des postulats de recherche sont décrits afin de circonscrire la portée de ces travaux de recherche et les objectifs sont établis. Ensuite, le second chapitre résume les activités de recherche en présentant les différentes phases méthodologiques qui ont été complétées dans le but d'atteindre les objectifs fixés. On y établit notamment les liens entre ces phases et les articles constituant cette thèse. Le sujet et les éléments clés de chaque article sont présentés, ainsi qu'un résumé des principaux résultats.

Les chapitres 3 à 8 présentent dans leur intégralité les quatre articles de journaux, dont deux ont été publiés, et les deux articles de conférences qui constituent cette thèse par articles.

Le chapitre 3 présente l'état de l'art dans le domaine de la comparaison des modèles CAO en trois dimensions (3D) selon trois perspectives particulières. D'abord, on y organise la variété de scénarios de comparaison des modèles CAO au sein de domaines d'application spécifiques. Les méthodes et autres approches existantes de calcul des différences sont comparées en identifiant leurs caractéristiques principales et leurs limitations. Puis, on présente un inventaire d'outils logiciels commerciaux capables d'identifier[1] des différences entre des modèles CAO. En conclusion, des pistes de recherche pour appliquer la comparaison des modèles CAO à la transposition du changement sont envisagées.

Le quatrième chapitre décrit des essais effectués sur des ensembles d'outils logiciels commerciaux accomplissant la comparaison de modèles CAO 3D. Ces essais ont permis d'évaluer les capacités de ces outils dans le calcul, la représentation et la présentation des différences appliqués à la transposition du changement. Ils sont basés sur deux scénarios de localisation et de représentation des modifications géométriques, chacun pilotant une série de tests réalisés sur des outils logiciels capables de comparer des modèles CAO procéduraux et explicites, respectivement. Des conclusions préliminaires sont émises en regard des résultats de ces tests.

Le cinquième chapitre présente une approche en trois étapes pour la représentation des scénarios de comparaison des modèles CAO 3D dans le but d'analyser leurs caractéristiques propres et de guider la sélection ou la conception d'une solution appropriée. On y représente les facteurs déterminant d'un scénario grâce au méta-modèle intitulé « *Product, Definition, Model and Formalism for Comparison* » ou PDMF4C. À titre d'exemple, le scénario de la réutilisation des modèles CAO via la recherche basée sur la géométrie est représenté en utilisant l'approche proposée.

---

[1] Le concept de l'*identification des différences entre modèles* fait référence à son équivalent en anglais *model difference identification* (MDI) et regroupe les étapes du calcul ou du repérage, de la représentation et de la visualisation des différences (Kolovos *et al.*, 2009).

Les sixième et septième chapitres introduisent en deux parties une nouvelle méthode de comparaison des modèles CAO 3D basée sur le calcul et la représentation des différences géométriques via les contraintes géométriques explicites (CGE). Au sixième chapitre, une approche pour la modélisation des différences, inspirée de la conception logicielle basée sur les modèles, est présentée. En plus de permettre la conservation des modèles des différences issus des comparaisons, cette approche se base sur un méta-modèle des différences géométriques qui demeure indépendant des formats des modèles CAO comparés. Ensuite, au chapitre 7, de nouvelles procédures de calcul des différences opérant la transposition des schémas de CGE, basées sur l'exploitation du méta-modèle des différences du chapitre précédent, sont décrites et exemplifiées par l'entremise de la comparaison de deux esquisses CAO similaires. Grâce à cette nouvelle méthode, les différences géométriques peuvent dorénavant être exprimées en termes de dimensions modifiées et de conditions géométriques enfreintes – en d'autres mots, selon un niveau d'abstraction de la forme plus intuitif pour un ingénieur mécanicien – plutôt qu'en termes de volumes ajoutés et supprimés ou de classifications d'éléments de forme primitifs.

Enfin, le chapitre 8 démontre la pertinence d'appliquer la méthode de comparaison des modèles CAO 3D, présentée au sein des deux chapitres précédents, à un scénario autre que la transposition du changement, soit celui de la recherche et de la réutilisation des modèles CAO en contexte de développement de nouveaux produits. Déjà, les différentes techniques de recherche basée sur la géométrie contribuent à la réutilisation des designs en permettant de réunir un échantillon de pièces similaires. La sélection du modèle candidat optimal pour la réutilisation requiert toutefois une méthode de comparaison plus raffinée, soit celle proposée par cette thèse, permettant de localiser chaque différence géométrique entre le modèle de référence et les modèles de l'échantillon et de représenter celle-ci de manière intuitive et fonctionnelle du point de vue de la conception.

Finalement, le neuvième et dernier chapitre souligne les contributions originales de ce projet de recherche au domaine de la comparaison des modèles géométriques, ses limitations et les recommandations qui en émanent. Les contributions sont d'abord présentées en rapport aux objectifs de cette thèse, mais également du point de vue plus vaste de l'état de l'art. Les limitations touchent les aspects liés aux postulats émis en début de travaux ainsi qu'à la solution proposée. Les recommandations pour la poursuite des travaux sont énoncées dans le but d'améliorer la solution proposée en rapport avec ces limitations.

CHAPITRE 1

PROBLÉMATIQUE DE RECHERCHE, POSTULATS ET OBJECTIFS

1.1 Problématique de recherche

L’évolution de la définition du produit correspond à sa transformation progressive depuis l’expression d’un besoin jusqu’à sa mise hors service, en passant par sa réalisation et son exploitation. Cette transformation progressive s’opère en une suite de phases organisées dans le temps qui s’inscrivent dans le cycle de vie du produit. Il s’agit de la « période qui comprend toutes les étapes de la vie d'un produit, depuis sa conception et sa fabrication jusqu'à son déclin, y compris son retrait du marché, son élimination et son rejet dans l'environnement » (Office québécois de la langue française, 2011).

Selon la perspective de cette thèse, l’évolution de la définition du produit relève principalement de la succession de changements opérés par les intervenants de l’entreprise à travers le cycle de vie du produit. Comme le décrit Bouikni (2005), sachant que chacun de ces intervenants possède sa propre expertise, leurs interventions doivent être faites conjointement afin de maintenir la cohérence de la définition du produit en plein expansion. En d’autres termes, il est impératif de maintenir la compatibilité entre les différents aspects d’un même composant. On y parvient lorsqu’il y a consensus entre ces intervenants sur l’évolution de la définition du produit.

Afin d’être compétitive, l’entreprise moderne doit être en mesure de rendre son produit disponible au marché dans des temps concurrentiels. Ceci correspond à réduire le délai durant lequel la définition du produit évolue pour parvenir à maturité. L’organisation du travail au sein des différentes phases du cycle vie du produit, ainsi que les outils supportant le travail des intervenants, constituent deux facteurs influents sur lesquelles une entreprise peut intervenir pour réduire ce délai.

Du point de vue organisationnel, il est légitime d'affirmer que la démonstration des bienfaits de l'approche concourante, aussi appelée « concurrent engineering » ou « ingénierie simultanée », n'est plus à faire. En effet, cette approche favorise la réduction de l'intervalle de temps nécessaire à son exécution. Par exemple, dès les phases initiales d'un projet, la constitution d'équipes pluridisciplinaires regroupant les expertises concernées de l'entreprise lui permet de devancer la mise en marché d'un nouveau produit (Bouikni, 2005; Cerezuela, Limam et Riopel, 2001).

Après réorganisation du travail au sein des phases du cycle de vie du produit, le temps nécessaire à la réalisation de chacune d'elles peut ensuite être réduit en améliorant les outils utilisés par chacun des spécialistes intervenant dans le développement du produit. En cette ère de l'information numérique, on explore alors la possibilité de fournir des logiciels performants qui favorisent une meilleure maîtrise et exploitation des données techniques définissant le produit, en particulier en fonction de la vision de la définition du produit propre à chaque discipline. L'application de fonctions spécifiques au sein d'environnements de travail adéquats vise ainsi à appuyer les spécialistes dans la transposition de leur savoir-faire respectif vers la résolution d'un problème de conception de produit.

Cette thèse aborde spécifiquement ce second volet, logiciel plutôt qu'organisationnel, du problème de la réduction du délai accordé à l'évolution de la définition d'un produit. Une telle réduction s'opère notamment par un meilleur soutien à la prise de décision qui, du coup, diminue le nombre d'erreurs commises et les délais consacrés à les corriger. Nous procéderons alors en ciblant certains paramètres clés agissant sur la définition du produit afin de faciliter la compréhension de la problématique associée à notre sujet de recherche.

1.1.1 Évolution de la définition d'un produit

Pour comprendre l'environnement dans lequel évolue la définition du produit, nous nous référons d'abord la figure 1.1 tirée de la thèse de Bouikni (2005) qui présente une excellente synthèse de trois aspects importants: (a) le volume des évolutions, (b) l'évolution dans un contexte d'ingénierie simultanée, et (c) le niveau de formalisme des évolutions.

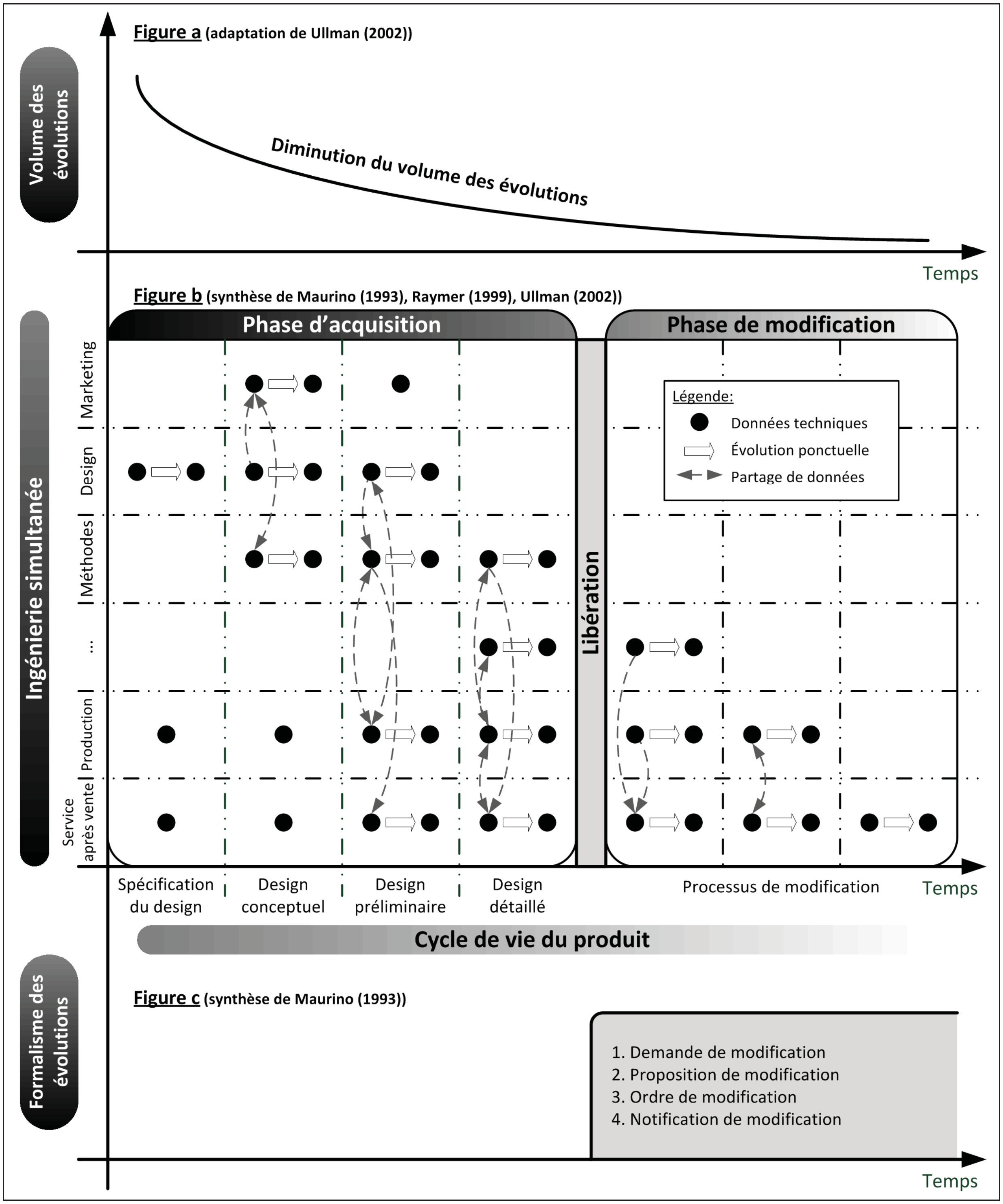


Figure 1.1 Évolution de la définition d'un produit au cours des phases d'acquisition et de modification
Redessiné à partir de Bouikni (2005)

Le terme '***évolution***' regroupe ici tous les changements d'ingénierie qui surviennent au cours de la définition du produit, soit autant lors des phases d'***acquisition*** que de ***modification***. La phase d'acquisition gère l'en-cours de la conception où les modèles et dossiers en cours d'évolution sont concernés par le processus de correction ou de reprise de la conception. Dans la phase de modification, les modèles et dossiers sont validés et libérés. De nouvelles évolutions appartiennent donc au domaine de la gestion des modifications (*change management*), ce qui nécessite le recours à un processus de modification bien formalisé (Maurino, 1995). La libération des modèles, dossiers ou données de définition, schématisée par la bande verticale sur la figure 1.1, marque la transition de la phase d'acquisition vers la phase de modification.

Le volume des évolutions au cours de la définition du produit diminue significativement entre les premières et dernières phases du cycle de vie, comme le montre la figure 1.1(a). Au moment de sa gestation, la définition du produit demeure abstraite et extrêmement muable et moins de ressources sont engagées dans le projet de développement du produit. Le coût de ces nombreux changements est donc moindre. Celui-ci augmente toutefois lorsque la définition du produit murit et que davantage de ressources viennent à être engagées dans le projet. Le volume des évolutions diminue donc au fur et à mesure que la définition se stabilise et que les coûts qu'elles engendrent deviennent de plus en plus importants.

Étant donné le coût d'un changement lors des dernières étapes de la définition du produit, son impact, à la fois sur le produit et sur les documents afférents à sa définition, doit être soigneusement évalué au cours du processus de modification avant de statuer sur sa pertinence et sur ses conditions d'application. Le niveau de formalisme exigé de la part du processus par lequel sont représentés et validés les changements lors de la phase de modification devient donc significatif, comme le représente la figure 1.1(c). Ce formalisme est notamment caractérisé par la segmentation du processus de modification entre quatre étapes, soit (1) la demande de modification, (2) la proposition de modification, (3) l'ordre de modification et (4) la notification de modification (Maurino, 1995).

En contrepartie, les évolutions de la définition du produit lors de la phase d'acquisition ne sont que peu ou pas structurées par un processus formel de validation. Il demeure toutefois d'une importance capitale que tout changement apporté à la définition du produit, même en phase d'acquisition, soit clairement décrit, représenté et disséminé à travers les différentes disciplines. Il faut faire évoluer de manière cohérente l'ensemble de l'en-cours de conception et permettre aux experts d'incorporer efficacement l'impact d'un tel changement dans leurs domaines respectifs d'intervention.

La figure 1.1(b) établit quant à elle une subdivision des différentes activités caractérisant le domaine des changements d'ingénierie. Le cycle de vie du produit, d'une part, et les disciplines participant à l'ingénierie simultanée, d'autre part, constituent les deux dimensions de cette subdivision. Ainsi, en abscisse, selon l'axe temporel, l'évolution dans le temps de la définition du produit correspond à son passage au travers les différentes phases de son cycle de vie. En phase de modification, cette évolution est incarnée notamment par des versions et des révisions. En phase d'acquisition, cette évolution correspond à la maturation de la définition du produit. Toutefois, étant donné le volume des évolutions et l'absence de formalisme dans le processus de changement, peu de représentations de cette maturation de la définition du produit sont disponibles.

En ordonnée, selon l'axe organisationnel, l'organisation du travail des différents spécialistes de l'entreprise représente l'intégration progressive des diverses spécifications et contraintes reliées à la réalisation du produit. Les données techniques définissant le produit sont représentées par des bulles noires. Chacune est assujettie à l'évolution de la définition du produit par la contribution de chacun des spécialistes à travers le temps. Les liens représentés par des flèches reliant les différentes bulles caractérisent la dépendance entre les données techniques partagées par des acteurs de l'entreprise qui travaillent conjointement.

1.1.2 L'évolution partagée entre disciplines

Le cadre de recherche de cette thèse se base sur la subdivision de l'évolution de la définition d'un produit selon les axes temporel et organisationnel dans un contexte d'ingénierie simultanée. Notre attention se situe plus spécifiquement sur l'aspect de partage de données techniques entre disciplines en situation de changement. La dynamique permettant à un ensemble de données techniques propres à une discipline d'évoluer indépendamment d'un état initial vers un nouvel état, tel que représentée par les flèches blanches au sein de la figure 1.1(b), va de soi. Toutefois, cette dynamique d'évolution sera plus complexe lorsque le partage et l'influence de données provenant d'autres champs d'expertises, représentées par les flèches pointillées, elles-mêmes en évolution, entrent en ligne de compte.

La figure 1.2 représente en quelque sorte un agrandissement de la figure 1.1(b) avec comme principal objet le partage entre deux disciplines (A et B) de données techniques en évolution lors d'une phase particulière du cycle de vie du produit, qu'elle survienne avant ou après la libération de définition du produit. Entre autres, les bulles noires représentant les données techniques sont désormais subdivisées en deux parties, l'une centrale et l'autre périphérique, distinguant chacune les données partagées entre disciplines, ou *communes*, des données spécifiques à chaque disciplines, ou *expertes*, respectivement.

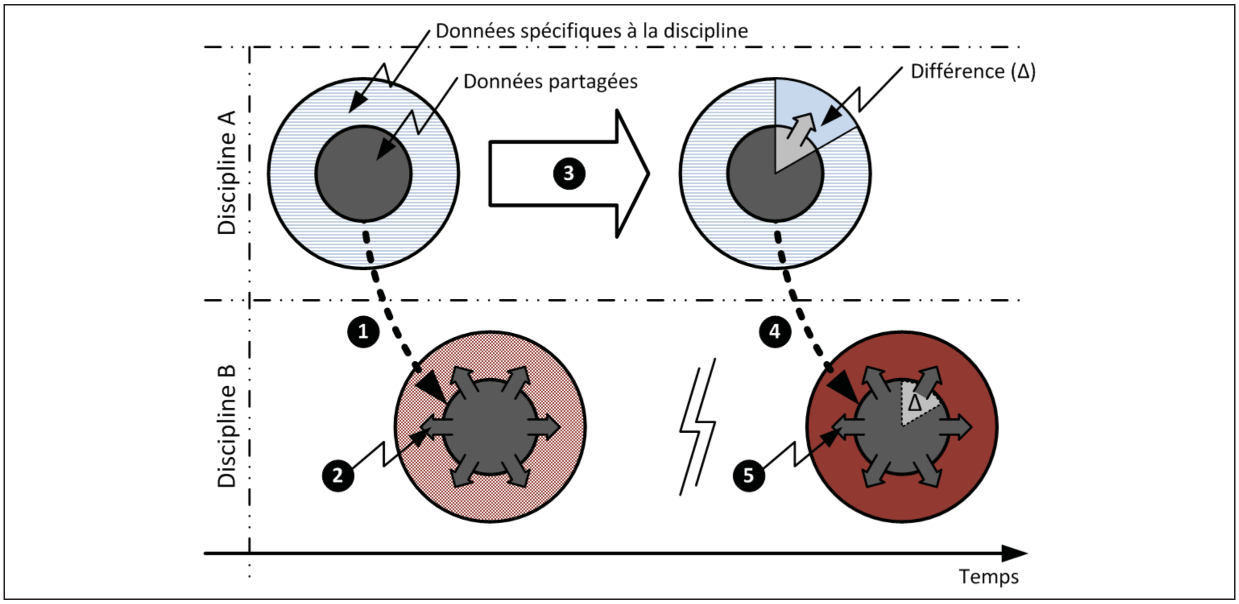


Figure 1.2 Régénération complète du modèle cible suivant l'évolution de la définition du produit et le partage de données entre discipline

Selon cette représentation de l'évolution de la définition du produit, la discipline A est caractérisée comme étant la *source* des données, en ce sens qu'elle génère un premier modèle synthétisant un ensemble d'informations relatives à la définition du produit. Un sous-ensemble de ces informations – les données communes – est ensuite partagé avec la discipline *cible* B (1) afin qu'elle puisse générer un second modèle à partir de ce sous-ensemble. Ce second modèle est alors constitué des données communes, ainsi que des nouvelles données spécifiques à la discipline cible (2). À ce point, la définition du produit n'a évolué que selon l'axe organisationnel. On dira de ces deux modèles qu'ils représentent deux points de vue dépendants sur un même état, ou une même version, de la définition du produit.

Un changement s'opère éventuellement selon l'axe temporel au niveau du modèle source (3); celui-ci est raffiné ou modifié par la discipline A. Cette évolution impacte à la fois la portion commune et la portion experte des données qui y sont représentées, générant un écart, ou ***delta*** (Δ), entre l'état original et le nouvel état de la définition du produit. Dans l'objectif d'assurer la cohérence des différents points de vue sur tout nouvel état de la définition du produit, les données communes du modèle source doivent être repartagées (4) vers la discipline cible B et l'évolution, incarnée par le delta au sein des données partagées, doit être propagée aux données expertes cibles.

Nous identifions ici deux scénarios par lesquels l'évolution de la définition du produit selon l'axe temporel, d'abord représentée au sein d'un modèle source, se propage à un modèle cible via le partage de nouvelles données communes. Ces scénarios sont appelés respectivement la *régénération* du modèle cible et la *transposition* du changement.

La figure 1.2 présente le scénario de la régénération du modèle cible, où la portion experte des données techniques spécifiques à la discipline cible est complètement régénérée en fonction du nouvel état des données communes partagées (5). Le processus initial de création de ces données expertes (2), nécessitant l'application rigoureuse du savoir-faire de la discipline, est simplement repris en totalité, assurant à la régénération le même ordre de cohérence entre les points de vue sur ce nouvel état de la définition du produit que lors de la création.

Dans le scénario de la régénération, la nature précise du delta caractérisant l'évolution au sein des données communes est négligée : sa seule existence commande la régénération du modèle cible. Il n'y a donc aucune intégration ou adaptation de ce delta du point de vue de la discipline A vers celui de la discipline B. La cohérence des points de vue de chaque discipline sur la définition du produit est facilement assurée dans ce premier scénario. Une régénération complète des données expertes de la discipline cible, basée sur l'application ponctuelle d'un savoir-faire, exige des délais importants puisque tout le travail original est repris. Ce faisant, ce scénario est jugé improductif, car la régénération de données expertes inchangées représente un investissement redondant de ressources, typiquement du temps-homme. De plus, en l'absence d'un lien de continuité entre les versions originales et régénérées de ces données expertes, aucun mécanisme autre que l'application de procédures formelles de génération (et de régénération) des données expertes ne peut garantir la cohérence des versions.

La transposition du changement, illustré à la figure 1.3, constitue le second scénario. Celui-ci diffère du premier justement par le fait que l'état original des données expertes de la discipline cible n'est pas abandonné, mais conservé (4') pour être modifié. L'intervention du spécialiste et l'application de son savoir-faire sont ainsi mieux ciblées et circonscrites que lors d'une régénération complète, réduisant le délai nécessaire à la propagation du changement. La modification des données expertes en B (6') en fonction du nouvel état des données communes exige toutefois que la nature du delta ainsi transmis (5') soit connue. L'impact du changement ainsi transposé vers les données spécifiques à la discipline cible peut alors être évalué plus minutieusement.

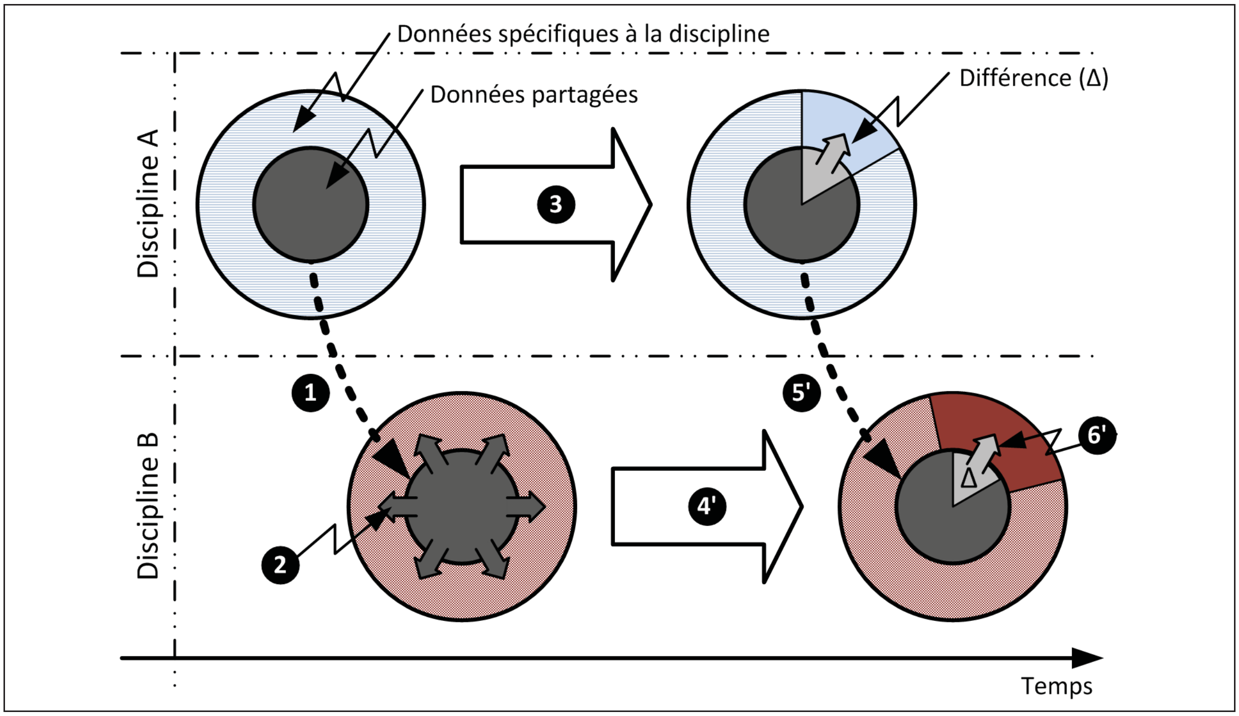


Figure 1.3 Transposition du changement entre modèles source et cible suivant l'évolution de la définition du produit et le partage de données entre discipline

Aujourd'hui, la quête de compétitivité pousse les entreprises à minimiser les délais de mise en marché, améliorer leur réactivité au changement et capitaliser sur leur expérience et leur savoir-faire, composant ainsi le contexte moderne de développement de produits. Selon ce contexte, nous émettons ici le postulat, fondamental à cette thèse, que le scénario de la transposition du changement apporte, dans de nombreux cas, davantage de bénéfices au processus de développement que celui de la régénération. Ce postulat se base sur la prémisse que la régénération du modèle cible implique une répétition presque entière du travail d'une discipline pour la génération de données expertes qui lui sont propres à partir des données partagées, ce qui consomme des ressources. À l'échelle du cycle de vie du produit, de telles reprises paralyseraient le processus de définition du produit, surtout lorsque le changement survient en amont autant selon l'axe temporel – lors de la phase d'acquisition où les changements sont nombreux – que selon l'axe organisationnel – lorsque les données communes modifiées sont issus des toutes premières phases du cycle de vie.

En supposant le scénario de la transposition interdisciplinaire du changement comme étant la voie à suivre lors de l'évolution de la définition du produit, on fait ainsi face à ses problèmes centraux que sont la caractérisation du changement ou du delta au sein des données communes et la représentation de ce delta en lien avec les données expertes cibles à modifier. Autrement dit, afin de limiter l'intervention d'une discipline sur son modèle du produit aux seules données expertes impactées par le delta au sein des données communes, il est impératif pour cette discipline de connaître avec précision ce delta et de l'interpréter rapidement et sans équivoque selon son point de vue propre sur le produit.

1.1.3 Exemple de partage de la géométrie entre études et méthodes

La notion de partage de données entre disciplines ayant lieu lorsque la définition du produit évolue dans le cycle de vie du produit (axe organisationnel) abordée jusqu'ici se concrétise ici dans le cas du partage de la géométrie d'une pièce mécanique entre les bureaux d'études et des méthodes au sein d'une entreprise. L'ingénierie d'études ou de conception correspond par exemple à la discipline source d'où origine la définition de la géométrie de la pièce, et l'ingénierie des méthodes ou de fabrication représente la discipline cible requérant le partage de cette géométrie pour la définition de la réalisation de cette pièce. Ainsi, à un moment donné de l'évolution de la pièce dans le temps, chaque discipline gère un modèle qui lui est propre où la géométrie de la pièce est représentée : il s'agit ici des données communes.

La figure 1.4 représente l'exemple décrit ici selon le schéma générique de transposition présenté précédemment à la figure 1.3. On y retrouve les mêmes concepts – données communes et données expertes spécifiques à chaque discipline – incarnés cette fois par des concepts plus concrets tels la géométrie et des schémas de cotation, respectivement, ainsi que les mêmes étapes du processus d'évolution.

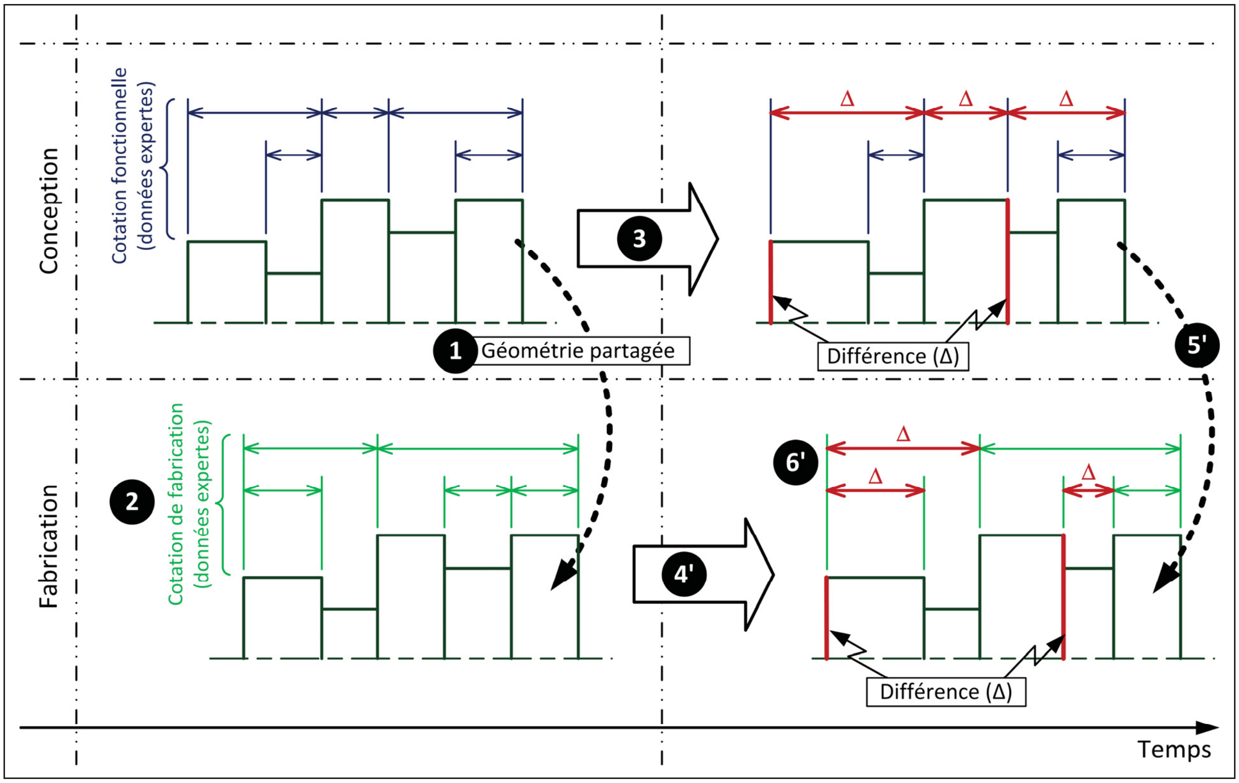


Figure 1.4 Transposition de changements géométriques initiés en conception vers le schéma de cotation de fabrication d'une pièce

Comme données expertes du point de vue de la conception, le schéma de cotation fonctionnelle, directement dépendant de la géométrie, est obtenu à la suite de l'analyse fonctionnelle de la pièce au sein du produit dont elle fait partie. À la suite du partage des données géométriques (1), l'ingénieur des méthodes conservera la géométrie du produit pour y joindre un schéma de cotation différent (2), soit celui de fabrication qui regroupe notamment les cotes-machines, les cotes-outils et les cotes d'appareils, comme résultat de la préparation d'une gamme d'usinage. Certes, le schéma de cotation fonctionnelle sera consulté ou « consommé » par les méthodes puisqu'il définit littéralement la pièce, mais celui-ci n'est pas conservé intégralement.

Une modification apportée par le bureau d'études au modèle de conception (3) peut l'être au niveau de la géométrie elle-même ou au niveau du schéma de cotation fonctionnel qui, communément, entraînera ensuite une modification de la géométrie. Dans l'exemple de la figure 1.4, deux surfaces planes sont identifiées comme le delta géométrique ayant pour source ou pour impact, selon les motifs soutenant la modification, le redimensionnement de trois cotes fonctionnelles sur cinq. On dénote du coup deux représentations, à des niveaux d'abstraction différents, de l'évolution du modèle de conception : une représentation purement géométrique et une représentation dimensionnelle.

Au niveau du modèle de fabrication, la modification du schéma de cotation de fabrication en fonction de la géométrie évoluée, telle que prescrit par le scénario privilégié de la transposition, commande avant tout la conservation temporaire du schéma original (4') en lien avec la géométrie, préservant un lien de continuité entre les versions du modèle de fabrication. Le partage du delta (5') vers le modèle de fabrication doit ensuite permettre l'identification précise des cotes de fabrication impactées par l'évolution (6').

Ici, malgré un niveau d'abstraction commun – une même notation symbolique, qui plus est – la représentation du delta partagé du modèle de conception vers le modèle cible ne peut pas être sous la forme de cotes, puisque les deux schémas présentent des éléments et une structure qui sont propres aux raisonnements de chaque discipline. Le contenu commun des deux modèles est géométrique : le partage du delta afin de transposer celui-ci au schéma de cotation de fabrication doit donc se faire au niveau géométrique.

Ce qui nous ramène à la problématique de cette thèse. Appliquée à l'exemple de la figure 1.4, la transposition de l'évolution du modèle de conception vers le modèle de fabrication nécessite de répondre à deux problèmes qui distinguent les approches de la transposition et de la régénération :

- Comment localiser, avec certitude et précision, le delta géométrique à partager du modèle de conception au modèle de fabrication?
- Comment adapter le niveau d'abstraction du delta géométrique à celui du schéma de cotation de fabrication afin d'en faciliter l'interprétation par l'ingénieur des méthodes et ainsi supporter l'application de son savoir-faire dans la mise-à-jour du modèle de fabrication?

La même problématique persiste lorsque, par exemple, le sens du partage est inversé. En considérant l'approche de l'ingénierie simultanée où les phases plus tardives de définition du produit telle la fabrication peuvent intervenir tôt en conception, l'évolution d'une pièce pourrait émaner du modèle de fabrication, se traduire en modifications géométriques devant être transposées au modèle de conception. L'impact fonctionnel d'un changement géométrique aux motifs de fabrication pourrait alors être mieux évalué, puis incorporé à la définition de la pièce ou contesté en cas de conflit.

La nature des données expertes propres à la conception et à la fabrication peut aussi varier ou, comme dans le cas illustré à la figure 1.5, changer de niveau d'abstraction. En effet, on peut imaginer la décomposition volumique d'une pièce en caractéristiques de conception – constituant ainsi l'arbre de construction du modèle de conception – et d'usinage – énumérant les principales opérations d'une gamme au sein du modèle de fabrication – comme une autre expression des données expertes. Encore une fois, on constatera que, malgré des niveaux d'abstraction similaires, les modifications apportées aux données expertes d'un des modèles ne peuvent être transmises telles quelles. Seul un delta géométrique doit être partagé et transposé afin de mettre à jour les caractéristiques volumiques du modèle cible. Il s'agit notamment de la problématique abordée par Subramani et Gurumoorthy (2005).

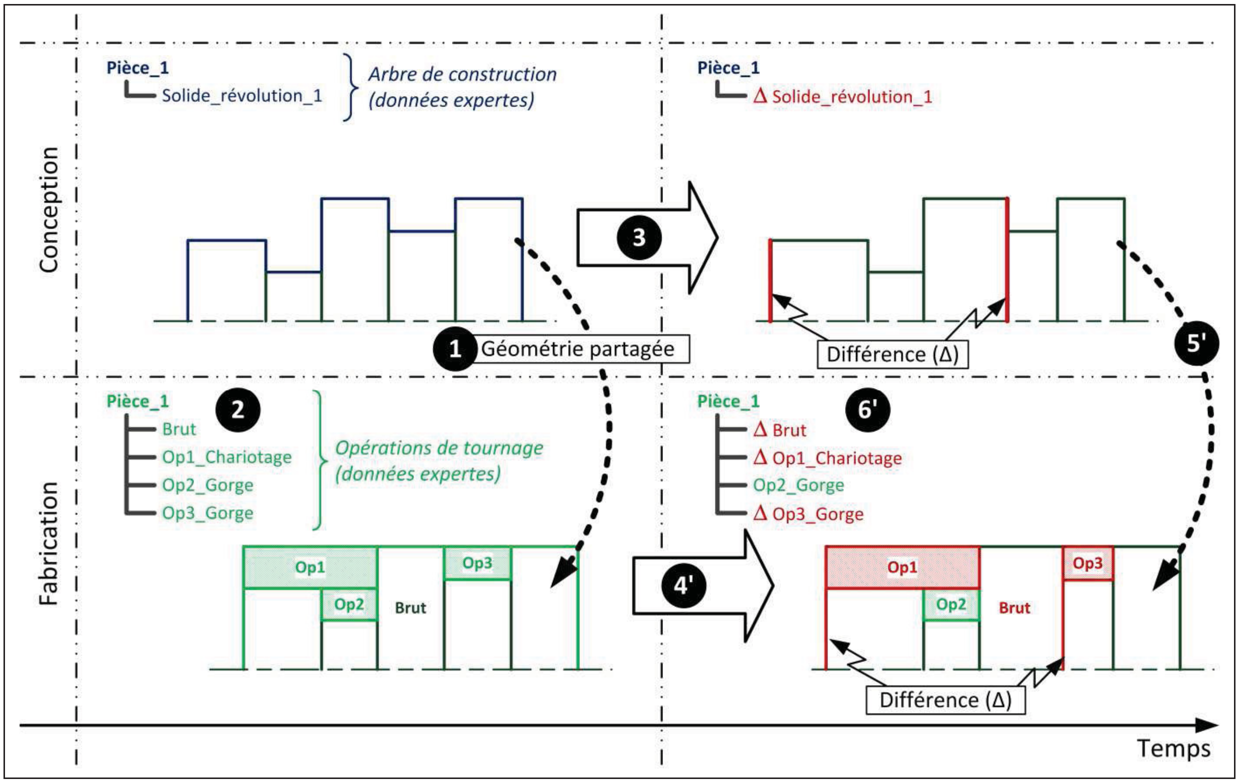


Figure 1.5 Transposition vers la décomposition volumique de l'usinage d'une pièce de changements géométriques initiés en conception

1.2 Postulats de recherche

Relativement à la problématique de recherche exposée précédemment, les postulats suivants sont émis afin de circonscrire le contexte de recherche de cette thèse.

1.2.1 Scénario de la transposition du changement

Entre les deux scénarios exposés précédemment sur l'évolution de la définition du produit et le partage des données entre disciplines, il est considéré que le scénario de la transposition du changement constitue le plus efficace. Il est en effet supposé que, conditionnellement à la disponibilité d'un processus ou d'un outil capable de localiser et de représenter adéquatement le delta distinguant deux versions d'un produit modélisé en CAO, la transposition du changement permet dans de nombreux cas l'économie non-négligeable de temps et de ressources dans l'exécution du processus de gestion de l'évolution. Ce faisant, on minimise la génération de données techniques en délimitant précisément le sous-ensemble de données techniques affecté par l'évolution et en prévenant la génération redondante des données non impactées.

1.2.2 Nature des données partagées

En considérant le modèle du produit tel qu'exposé par la figure 1.1(b), constitué de plusieurs modèles concourants propres aux disciplines impliquées dans le développement du produit mécanique, il est supposé qu'un élément commun fédérateur existe entre ces différentes vues sur le même produit et que cet élément est la géométrie nominale du produit. Ainsi, dans le scénario de transposition de changement illustré aux figures 1.3, 1.4 et 1.5, la géométrie de la pièce ou de l'ensemble étudié constitue les données partagées entre disciplines et au sein desquelles un delta est d'abord identifié entre les modèles original et évolué. La représentation de ce delta selon un modèle ou formalisme plus adéquat à l'interprétation et l'assimilation de ce delta géométrique par la discipline est à réaliser.

On entend ici par « géométrie » uniquement la forme spécifiée du produit telle qu'incarnée par les modèles CAO. On néglige ainsi toute différence relative à la représentation de la géométrie. D'emblée, on rejette toute différence qui pourrait être identifiée entre deux géométrie identiques représentées différemment. Les modes de représentation des modèles comparés peuvent diverger; par exemple, un arbre de construction solide générative (CSG) versus une représentation par les frontières (B-Rep). De plus, aucun mode de représentation en CAO n'est unique : un même objet peut être représenté par plus d'un arbre CSG ou par différents découpages de sa frontière en B-Rep.

Ainsi, dans un contexte de conception mécanique, toute différence entre deux modèles qui est fondamentalement géométrique, i.e., relative à la forme de la pièce, est considérée sémantique. Inversement, dans cette thèse, les différences issues de la représentation numérique des modèles, de leur structure interne, n'incarne en rien l'évolution de la définition du produit.

1.2.3 Dispersion du modèle du produit

Cette thèse adopte une perspective réaliste par rapport à l'exploitation des logiciels CAO en industrie aujourd'hui. Des notions telles l'entreprise étendue, l'exploitation du web (Vezzetti, 2009) et la grande diversité des systèmes CAO (logiciels et expertises) parmi les entreprises contemporaines incarnent cette perspective et sont fortement pris en compte. Iraqi Houssaini, Kleiner et Roucoules (2012) décrivent trois approches à ce problème de l'interopérabilité entre systèmes et modèles CAO :

- *L'intégration* : tous les modèles experts sont représentés à l'aide d'un unique formalisme;
- *L'unification* : une structure de données commune de haut niveau offre la possibilité d'établir des correspondances sémantiques entre les différents modèles experts; et
- *La fédération* : plusieurs modèles experts sont associés dynamiquement selon des schémas de correspondance basés sur la reconnaissance de concepts équivalents ou similaires.

La tendance industrielle depuis le début du 21^e^ siècle semble délaisser l'intégration du modèle produit pour aller en direction inverse, soit plutôt vers un éclatement du modèle produit avec la croissance des échanges de données, l'utilisation de logiciels spécialisés, l'émergence des modèles allégés (*lightweight*), etc. Ainsi, les multiples modèles représentant différents aspects du produit ne sont pas regroupés sous une même et unique niche, mais plutôt éparpillés autant au niveau disciplinaire que géographique. Les mécanismes assurant la cohérence de ces modèles disséminés doivent donc être repensés.

Dans le contexte spécifique de la transposition de changement entre modèles de disciplines différentes – voire délocalisées – une telle réalité se matérialise dans l'incapacité à exprimer a priori les liens complexes entre les modèles comparés en fonction de la connaissance qu'ils représentent, comme en discutent Tremblay *et al.* (2006). Ainsi, il est supposé qu'aucun lien formel et/ou détaillé n'existe entre les modèles autres qu'un lien de haut niveau établissant que le second représente l'évolution du produit représenté par le premier

Depuis plus de vingt ans, plusieurs ouvrages ont abordé le problème de la transposition du changement entre modèles de caractéristiques. Ceux-ci reposent toutefois sur l'hypothèse d'une intégration forte entre les modèles du produit, intégration incarnée par l'exploitation soit d'un système de modélisation unique (Bronsvoort *et al.*, 1997; De Martino, Falcidieno et Hassinger, 1998) ou d'une fédération de systèmes (Hoffmann et Joan-Arinyo, 2000; Hoffmann et Joan-Arinyo, 1998). Autrement dit, ces travaux reposent sur l'hypothèse que des liens persistants sont créés et maintenus à un bas niveau de granularité entre les modèles propres à chaque discipline tout au long du cycle de vie du produit afin d'assurer la cohérence, essentiellement géométrique, entre tous ces modèles.

Or, Sypkens Smit et Bronsvoort (2009b) constatent finalement en abordant la problématique similaire du partage de données entre modèles de conception et d'analyse par éléments finis qu'aucune solution globale pour la création et, plus particulièrement, pour le maintien de ces liens n'a été trouvée. Ainsi, malgré les avancements en automatisation dans la génération des données expertes à partir d'une géométrie donnée, aucun mécanisme de liaison entre les différents modèles-vues du produit pouvant être maintenu dynamiquement advenant une évolution de la définition du produit n'existe. Cette thèse avalise l'observation de ces auteurs en faisant le postulat que, concrètement, le modèle du produit n'est pas intégré ou unifié tout au long de son cycle de vie, mais bel et bien dispersé. Elle adopte plutôt l'approche de la fédération des modèles experts et de la mise en correspondance dynamique de ces modèles basée sur la géométrie.

1.3 Objectifs des travaux

Ce projet de recherche examine donc le scénario de la transposition du changement entre deux modèles d'un même composant, propres à des disciplines distinctes mais connexes. Il met plus particulièrement l'emphase sur les étapes de la localisation et de la représentation a posteriori des modifications.

L’objectif principal de cette thèse est de proposer une approche de comparaison permettant la caractérisation des différences repérées entre deux versions d’un même modèle géométrique d’un composant mécanique. Le modèle de représentation des différences proposé devra offrir un niveau adéquat de sémantique afin de favoriser l’interprétation et l’assimilation du delta géométrique du point de vue particulier d’une discipline.

Ainsi, une propagation efficace de l’évolution de la définition d’un composant à travers les différents modèles dispersés qui la composent repose sur une connaissance et une compréhension approfondie du delta qui la caractérise.

1.3.1 Premier objectif spécifique : Localiser avec précision

Une première exigence envers l’approche de comparaison recherchée sera de l’établir d’abord sur une méthode de calcul des différences géométriques entre deux modèles permettant la localisation d’un delta géométrique avec certitude et précision. Cette méthode pourra faire l’objet d’une nouvelle proposition dans le cadre de cette thèse ou provenir, avec ou sans adaptation, de travaux existants.

La réduction du délai accordé à l’évolution de la définition d’un produit par des solutions logicielles sous-entend que ces dernières soient fiables, i.e. qu’elles puissent être opérées en garantissant un minimum d’erreurs dans les informations retournées à l’utilisateur. Lorsqu’il est question de calculs, on est alors en droit de s’attendre à obtenir des résultats qui n’induiront pas l’utilisateur en erreur, ce qui, autrement, nuirait au processus de gestion des modifications plutôt que de l’améliorer. La localisation des différences constituant le delta géométrique entre les deux modèles devra notamment être exempte d’omissions (critère de rappel) et d’erreurs (critère de précision).

1.3.2 Second objectif spécifique : Communiquer la différence

Comme l’indique Caplat (2008), toute représentation d’un concept au travers d’un modèle est intimement liée au formalisme ou langage à partir duquel elle est élaborée. En fait, tout formalisme influence grandement, par l'ensemble de symboles et de règles syntaxiques qui lui sont propres, le pouvoir d'expression d'un modèle dans la représentation d'un point de vue sur le sujet en en définissant les limites de ce qui peut être exprimé.

Le second objectif spécifique de cette thèse sera de proposer un formalisme fournissant des attributs permettant la caractérisation significative de la différence géométrique du point de vue d'un ingénieur, par exemple d'ingénieurs concepteurs et/ou d'ingénieurs des méthodes. Le niveau d'intelligibilité de ces attributs par des ingénieurs réalisant la transposition du changement entre des modèles experts influencera corrélativement l'efficacité et la justesse du processus. En d'autres mots, sans exécuter la transposition elle-même, le formalisme optimal devra rapprocher l'expression du delta géométrique le plus possible de celle de l'évolution de la définition incarnée par les données expertes obtenues par l'application du savoir-faire de la discipline cible. Par exemple, les attributs proposés devront faciliter l'identification de l'épicentre d'un changement, i.e. le ou les éléments de formes du composant dont la modification est à la source des différences des éléments voisins ou associés.

1.3.3 Troisième objectif spécifique : Formaliser l'évolution

Tel qu'exposé à la section 1.1.1, l'évolution de la définition du produit se décline en deux phases consécutives démarquées par la libération des modèles, soit la phase d'acquisition et la phase de modification. Le troisième objectif spécifique de cette thèse consiste à assurer l'applicabilité de l'approche de comparaison proposée durant ces deux phases.

La contrainte principale à l'atteinte de cet objectif spécifique provient du niveau élevé de formalisme requis lors de la phase de modification, tel qu'illustré à la figure 1.1(c). En ce sens, l'approche proposée devra, au minimum, permettre une documentation adéquate de la transposition du changement pour les registres. Idéalement, la représentation du delta géométrique devra pouvoir demeurer en liaison avec les versions originales et modifiées des modèles experts. Ainsi, la représentation du delta géométrique serait élevé au rang de données pérennes au sein du cycle de vie du produit plutôt que transactionnelles, exploitables a posteriori dans des contextes, par exemple, de gestion des versions et de réutilisation des modèles.

Ces contraintes devront toutefois être surmontées sans ralentir et pénaliser la transposition du changement en phase d'acquisition dont la fréquence d'occurrence est de beaucoup supérieure. Ce faisant, l'approche de comparaison proposée apportera un niveau de formalisme nouveau à la phase d'acquisition.

1.4 Synthèse

Donc, cette thèse se veut une contribution aux efforts pour la réduction des délais impliqués dans l'évolution de la définition d'un produit en proposant une nouvelle approche logicielle au problème de la transposition du changement entre modèles géométriques incarnant différents points de vue experts sur le produit. Plus spécifiquement, l'approche de comparaison des modèles proposée visera la localisation et la représentation du delta géométrique caractérisant l'évolution de la définition géométrique de manière à faciliter la propagation de cette évolution aux données expertes d'un modèle cible, i.e., qui n'est pas la source de la différence géométrique. Le cas de la propagation d'une modification émanant de l'évolution d'un modèle de conception mécanique, tel un modèle CAO, vers un modèle de fabrication mécanique, tel un modèle FAO, et vice versa, servira d'exemple principal tout au long des travaux.

Afin de supporter l'ingénieur dans l'interprétation et l'assimilation de l'évolution de la définition géométrique du composant mécanique, des critères de fidélité et de précision pour le calcul du delta géométrique sont prescrits. L'intelligibilité de la représentation du delta par le choix d'attributs significatifs dans un contexte d'ingénierie constitue également une cible de l'approche proposée. On vise l'applicabilité de l'approche tout au long du cycle de vie du produit dont les différents modèles experts constituant le modèle du produit sont supposés dispersés plutôt qu'intégrés.

CHAPITRE 2

MÉTHODOLOGIE DE RECHERCHE

Ce chapitre présente un résumé des activités incluses à la méthodologie de recherche qui ont été menés dans le cadre de cette thèse afin d'atteindre les objectifs énoncés au chapitre précédent. La structure de ce chapitre, illustrée grâce à la figure 2.1, se base sur la description des différentes phases de la méthodologie de recherche ainsi que sur leur association aux articles scientifiques et autres articles de conférence constituant cette thèse.

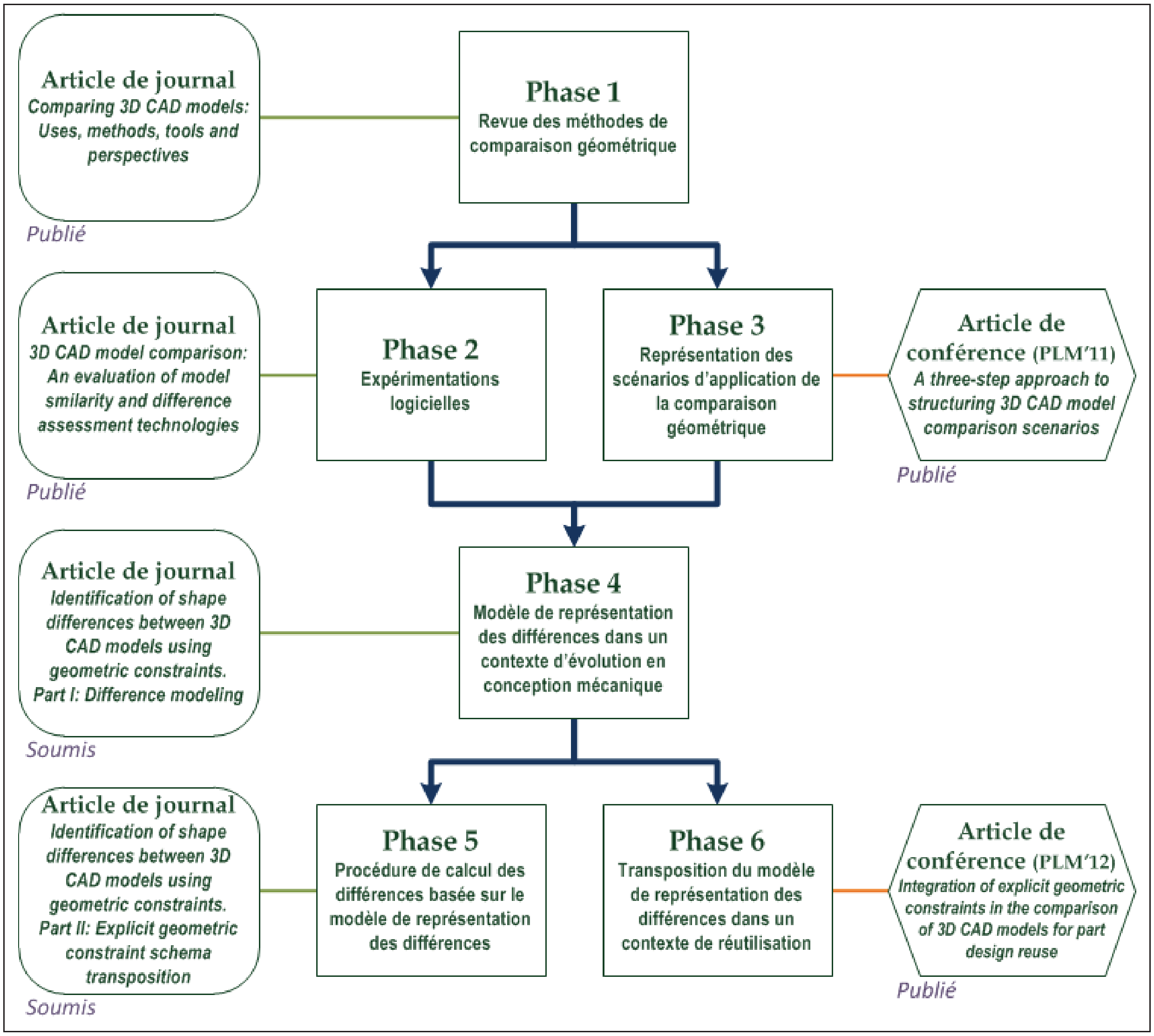


Figure 2.1 Schéma représentant la méthodologie de recherche adoptée

2.1 Phase 1 : Revue des méthodes de comparaison géométrique

Cette thèse débute par une phase exploratoire visant la collecte d'information et une mise à niveau des connaissances de base sur le sujet d'étude, soit le repérage et la représentation des différences entre deux modèles géométriques incarnant la forme de produits mécaniques. La littérature scientifique est d'abord analysée afin de dresser un portrait se voulant le plus juste et complet possible des récents développements relatifs à la comparaison de modèles géométriques, toutes applications confondues. Autrement dit, cette analyse dépasse évidemment le principal champ d'application de la conception mécanique, dans la perspective d'introduire de nouvelles techniques, notamment de la géométrie algorithmique (Akenine-Möller, 2005; Pottman, Leopoldseder et Zhao, 2003), en CAO.

La documentation consultée ne se limite également pas à la conventionnelle littérature scientifique, car il est observé durant les recherches que peu d'articles et d'ouvrages abordent directement le sujet de la comparaison géométrique. Les recherches couvrent aussi la documentation industrielle (CAx Implementor Forum, 2008; Cessna Aircraft Company, 2010; Frechette, 1996), les brevets d'invention (CoCreate Software GmbH et Gutierrez, 3 juin 2010; Parametric Technology Corporation, 23 septembre 2003; Translation Technologies Inc., 12 décembre 2006) et les spécifications logicielles (Cheney, 2008; ITI TranscenData, 2010b). Ce regard plus vaste, notamment sur les enjeux industriels, permet ainsi de dresser un inventaire très exhaustif des applications de la comparaison des modèles 3D de pièces mécaniques.

L'analyse de l'état de l'art a été réalisée selon trois points de vue :

- Les applications de la comparaison géométrique dans un contexte global de développement de produits;
- Les diverses méthodes de calculs permettant la mesure et/ou la caractérisation de la similitude/différence entre modèles géométriques; et
- Un inventaire des outils logiciels d'origines commerciales et universitaires capables d'identifier les différences entre deux modèles CAO.

De cette analyse ont été dégagées trois pistes de recherche à explorer sur le sujet de la comparaison des modèles CAO, et plus particulièrement relativement à la problématique de la transposition du changement, à la suite des observations réalisées durant cette première phase de la recherche. En résumé, ces pistes de recherche sont :

- Beaucoup d'efforts ont été dirigés jusqu'ici sur le développement de méthodes rapides et précises de calcul des différences entre modèles, mais très peu sur leur représentation, pourtant un élément tout aussi important du problème de l'identification des différences entre modèles. Dans un contexte de transposition de changement où le delta entre deux états de la définition géométrique du produit doit pouvoir être interprété rapidement et sans ambiguïté selon le champ sémantique précis d'une discipline cible, la représentation des différences revêt un caractère fondamental et requiert davantage d'attention dans le choix d'une solution.

- La comparaison des modèles CAO modernes représentés de manière procédurale – par leur arbre ou historique de construction – semble plus prometteuse vis-à-vis la production d'information pertinente sur les différences géométrique du point de vue de la conception étant donné le haut niveau de sémantique portée par des entités telles les caractéristiques de modélisation (ex. : extrusion, perçage, poche, etc.). En contrepartie, cette approche pour la comparaison présente des déficiences, notamment par rapport à sa flexibilité d'application et au fait que les représentations géométriques comparées sont implicites. Une solution prometteuse au problème de transposition de changement serait de parvenir à combiner les avantages de la comparaison géométrique explicite, présentant une flexibilité d'application supérieure, ainsi que des précisions et des taux de rappel intéressants, à la richesse des représentations des différences géométriques obtenues grâce à la comparaison géométrique implicite.

- Malgré la variété des outils logiciels pour l'identification des différences géométriques et des approches de calcul, peu ou pas de solutions n'incluent la représentation des différences sous la forme de données paramétriques; i.e., les dimensions et autres contraintes géométriques constituant le premier niveau de sémantique d'ingénierie présent en CAO. Pourtant, autant en modélisation par caractéristiques qu'en modélisation directe, les données paramétriques constituent généralement le point d'entrée d'une modification apportée à un modèle. Dans la recherche d'une solution de comparaison géométrique intuitive favorisant l'interprétation des résultats, la représentation des différences géométrique sous la forme paramétrique demeure donc une avenue à explorer.

Le compte-rendu de cette analyse de la littérature a été produit sous la forme d'un premier article scientifique intitulé « *Comparing 3D CAD models : uses, methods, tools and perspectives* » (Brière-Côté, Rivest et Maranzana, 2012a). Le contenu de ce premier article, entre autres une synthèse des recherches documentaires, est présenté au chapitre 3 de cette thèse. Des informations sur la publication de cet article dans la revue « *Computer-Aided Design and Applications* » sont présentées à l'annexe I.

2.2 Phase 2 : Expérimentations logicielles

Lors de la seconde phase, les activités de recherche reliées à cette thèse sont gouvernées par une approche plus empirique que lors de la première phase, davantage théorique, en mettant l'accent sur l'étude des outils logiciels implémentant la comparaison géométrique de modèles CAO. Réalisée en deux temps, cette phase « expérimentale » débute avec une participation au développement d'un logiciel portant sur la comparaison géométrique. Ensuite, l'inventaire de l'offre logicielle élaboré à la phase précédente est étudié et évalué plus en profondeur grâce à l'essai méthodique de plusieurs outils logiciels.

2.2.1 Stage en développement logiciel

Un stage réalisé au sein de l'équipe de développement logiciel de l'entreprise 3DSemantix® durant une certaine période au début des recherches a contribué à l'acquisition de connaissances dans le domaine des logiciels CAO, spécifiquement en comparaison géométrique, en commandant la résolution de problèmes concrets. Les prototypes logiciels auxquels les travaux de cette thèse ont contribué sont 3DPartFinder®, un engin de recherche basé sur la géométrie au sein de bases de données CAO, ainsi que 3DComparator, un outil pour la comparaison explicite de deux modèles CAO et l'identification de leurs différences géométriques.

La réalisation de ce stage met en évidence les réalités suivantes en lien avec la dimension pratique de la comparaison géométrique et des logiciels qui l'implémentent :

- la recherche de l'efficacité pour les méthodes de calcul, comme pour la comparaison géométrique dans un objectif de recherche au sein d'une base de données;
- le développement des interfaces avec les systèmes CAO, autant graphiques qu'applicatives;
- le contrôle de la précision des mesures extraites depuis les géométries, à savoir de connaître et maîtriser tous les facteurs qui l'influencent, spécialement en contexte de comparaison où une même métrique mesurée selon des précisions différentes peut mener à de faux positifs;
- La mesure de la différence, comme dans le cas de la recherche basée sur la géométrie, est un concept trop abstrait pour revêtir une signification concrète dans un contexte d'ingénierie, autre que de fournir un ordre de grandeur qui, en plus, peut parfois varier selon l'échantillon comparé.

2.2.2 Évaluation de l'offre logicielle

La seconde partie de l'étude pratique est l'essai et l'évaluation d'un échantillon représentatif des outils logiciels disponibles sur le marché possédant des capacités de comparaison des modèles CAO et d'identification des différences géométriques. Cette étude se base sur l'inventaire des solutions logicielles réalisé à la phase méthodologique précédente et publié dans le premier article de cette thèse (Brière-Côté, Rivest et Maranzana, 2012a). Au sein de cet inventaire, les outils logiciels sont catégorisés en quatre familles :

- les systèmes CAO comprenant une ou plusieurs fonctions de comparaison des modèles à même l'environnement de modélisation;

- les logiciels de visualisation et d'annotation, offrant la possibilité à un public plus large que celui regroupant les utilisateurs de CAO d'effectuer des comparaisons simples de modèles CAO allégés;
- les logiciels de validation géométrique permettant de vérifier la qualité et la cohérence des modèles CAO ayant subi une traduction de format dans des contextes d'échange de données géométriques ou de migration, par exemple.
- tout autre outil logiciel présentant la capacité de comparer des modèles géométriques en trois dimensions.

L'échantillon est constitué d'un total de quinze (15) logiciels étant disponibles sans frais au moment de l'étude et dont les manipulations pouvaient être réalisées soi-même sans intermédiaires (i.e., aucune démonstration, aucun essai réalisé par un représentant, etc.). En plus de permettre la comparaison des géométries explicites, trois de ces logiciels possèdent également la capacité de comparaison de modèles CAO représentés de manière procédurale; ils font l'objet d'une série distincte de tests étant donné les particularités des méthodes selon qu'elles comparent la représentation explicite ou procédurale des modèles CAO. Les cinq critères définis et utilisés pour l'évaluation de ces logiciels dans l'identification des différences géométriques sont :

- le taux de rappel, qualifiant la capacité du logiciel de repérer toutes les différences géométriques présentes entre deux modèles;
- la précision de calcul, qualifiant la capacité du logiciel à ne pas repérer des différences qui n'en sont pas;
- la richesse de la représentation, qualifiant le niveau de détail et la quantité d'information fournie dans la description des différences localisées;
- l'exactitude de la représentation, qualifiant la justesse des informations fournies; et
- l'intelligibilité des moyens de visualisation des différences et de leurs descriptions.

Les trois principales observations en lien avec l'objectif de cette thèse découlant des essais et de l'étude des résultats sont :

- Comparer les représentations procédurales des modèles CAO permet certes de mieux discerner les origines des différences géométriques puisque celles-ci sont exprimées sous la forme de caractéristiques de formes modifiées, mais les essais montrent que les différences au niveau d'un arbre de construction ne sont pas en directe corrélation avec le résultat géométrique;
- Le recours aux géométries approximées par facettisation ou décomposition spatiale est déconseillé lorsqu'il est question de caractériser le delta distinguant deux versions d'une même géométrie, car les mesures obtenues présentent inévitablement un certain niveau d'erreur lors de leur évaluation et qu'il peut devenir difficile de distinguer les différences significatives des différences dues à l'approximation des formes.
- La représentation graphique des différences doit forcément être complémentée par d'autres moyens de représenter et visualiser les résultats, tels des nomenclatures et des rapports, pour éviter que des différences détectées par l'outil logiciel ne soient pas remarquées par l'utilisateur lors de l'inspection des résultats, omissions pouvant avoir de graves conséquences en propagation et transposition de l'évolution.

Le compte-rendu de cette évaluation de l'offre logiciel prend la forme d'un second article scientifique intitulé « *3D CAD model comparison: An evaluation of model similarity and difference assessment technologies* » (Brière-Côté, Rivest et Maranzana, 2013). Le contenu de ce second article est présenté au chapitre 4 de cette thèse. Des informations complémentaires sur la publication de cet article dans la revue « *Computer-Aided Design and Applications* » sont présentées à l'annexe I.

2.3 Phase 3 : Représentation des scénarios d'application de la comparaison géométrique

Les scénarios d'application de la comparaison géométrique des modèles CAO sont multiples. Les résultats de la première phase le démontrent, notamment par la catégorisation de ces scénarios en six domaines d'application distincts :

- lors de la traduction/réédition des données CAO,
- en réutilisation des données produit,

- en gestion de l’évolution en ingénierie,
- pour le maintien des procédures de modélisation CAO,
- pour la standardisation des pièces et des produits, et
- pour la création des modèles d’analyse.

Cette première catégorisation permet de situer globalement le scénario de la transposition du changement entre modèles géométriques, central à cette thèse, par rapport aux autres scénarios d’application et, du même coup, par rapport aux solutions existantes. En effet, la transposition du changement constitue une problématique propre à la gestion de l’évolution en ingénierie, car elle englobe notamment l’analyse de l’impact d’un delta sur un modèle cible et la propagation cohérente de ce delta aux données expertes d’un tel modèle.

Conséquemment, la troisième phase méthodologique de cette thèse vise à approfondir cette première catégorisation des scénarios par la proposition d’un cadre théorique pour la représentation de ces scénarios. En plus de fournir un outil supplémentaire à la compréhension de la problématique sous ses diverses formes déjà documentées, l’usage d’un formalisme dédié doit permettre l’identification et l’organisation des quelques facteurs déterminants qui soit associent soit distinguent ces scénarios les uns par aux autres et ainsi orientent le choix d’une solution.

Avec comme objet d’études de la comparaison le modèle CAO lui-même, les concepts à la base du cadre théorique proposé pour cette phase proviennent de la théorie de la méta-modélisation, soit celle présentée par Caplat (2008). Par méta-modélisation, on doit comprendre ici la modélisation des modèles, i.e., leur représentation selon un niveau d’abstraction plus élevé afin d’en comprendre leurs contextes, leur composition, leurs rôles, etc. Ainsi, le sujet modélisé, le point de vue porté sur celui-ci et le formalisme dans lequel le modèle est exprimé caractérisent fondamentalement le modèle de spécification qu’est le modèle CAO et, conséquemment, le scénario dans lequel il est comparé à un autre.

En somme, trois facteurs déterminants provenant de l'analyse du scénario lui-même et des modèles comparés permettent de caractériser significativement ce scénario :

1. la fonction de la comparaison ou, en quelque sorte, la question à laquelle la comparaison de deux modèles CAO doit répondre, ce qui prescrit la nature et la forme attendues des résultats;
2. la composition des modèles comparés et les relations conceptuelles pouvant être établies entre les deux structures, soulevant du coup les similitudes apparentes à ce niveau; et
3. les disciplines impliquées, soit celle qui commande la comparaison pour ses besoins d'analyse et celle dont le ou les modèles sont comparés, établissant ainsi l'adéquation entre les besoins en information et la nature de celle disponible.

Le cadre pour la représentation des scénarios d'application proposé lors de cette troisième phase a été présenté lors d'une conférence internationale et constitue le sujet d'un article de conférence intitulé « *A three-step approach to structuring 3D CAD model comparison scenarios* » (Brière-Côté, Rivest et Maranzana, 2011). Le contenu de cet article de conférence compose le cinquième chapitre de thèse.

2.4 Phase 4 : Modèle de représentation des différences

L'interprétation et l'assimilation de l'évolution de la définition du produit mécanique représentée au sein de modèles CAO sont ici capitales. Une meilleure caractérisation du delta géométrique entre deux modèles réside donc avant tout dans la représentation de celui-ci, puis dans la précision de son calcul. Tout résultat mathématique ou géométrique incarnant la différence entre deux formes doit pouvoir transmettre davantage que l'issu d'un calcul : par le choix adéquat d'attributs et d'expressions pour les communiquer, ces résultats doivent porter une signification pour les disciplines cibles. Il en va de soi pour la mise à niveau efficace et cohérente des modèles cibles dans le processus de propagation du changement.

Les lacunes décelées au sein de la revue de la littérature et de l'expérimentation logicielle le démontrent : très peu de modes de représentation des différences entre modèles CAO portent une sémantique adéquate pour stimuler le raisonnement sur la définition du produit dans un contexte d'ingénierie. Afin de la rapprocher le plus possible de l'édition du modèle cible, la différence doit être exprimée en fonction de la discipline, i.e., selon un langage ou un formalisme qui lui est familier, qui lui permet d'exprimer le plus fidèlement possible son intention. Pertinemment, la quatrième phase méthodologique de cette thèse consiste à identifier le formalisme qui convient à la fois à l'expression naturelle des intentions des ingénieurs de conception et de fabrication – vers l'intelligibilité de la différence – et à la représentation plus formelle des différences – vers la fonctionnalité de la différence.

La représentation de la différence ne peut donc être envisagée globalement : pour chaque discipline, un delta géométrique aura une signification propre, un impact particulier sur les modèles en place. C'est ce que la phase méthodologique précédente aura permis entre autres d'exposer : une intention de conception ne peut être exprimée que dans les limites du formalisme choisi pour la représenter. En ce sens, les formalismes traditionnellement exploités en conception mécanique, en conception assistée par ordinateur de surcroit, sont tout indiqués.

Afin de faire ainsi le pont entre la géométrie, son évolution et l'intention du concepteur, des points de vue des études et des méthodes conjointement, cette thèse adopte le formalisme du dimensionnement géométrique, normalisé notamment au sein des normes ASME Y14.5-2009 (American Society of Mechanical Engineers, 2009) et Y14.41-2012 (American Society of Mechanical Engineers, 2012), pour l'expression des différences géométriques entre modèles, mais plus particulièrement exploité en CAO sous la forme des contraintes géométriques appliquées en conception. On prend ainsi exemple sur la pratique industrielle traditionnelle telle que rapportée par Quintana (Quintana, 2011; Quintana, Rivest et Pellerin, 2012) où, notamment, l'évolution d'une pièce ou d'un ensemble est d'abord exprimée par des annotations manuscrites appliquées aux cotes d'un dessin de définition, annotations respectant parfois un modèle et une symbolique bien précise (exemple : l'utilisation de tampons d'annotations sur des dessins imprimés sur papier).

Ce formalisme est ensuite implémenté sous la forme d'un modèle de représentation des données incarnant les différences géométriques de manière à ce que celles-ci puissent être intégrées, exploitées et sauvegardée. Cette thèse prend alors exemple dans la réalisation de cette phase méthodologique sur des travaux de génie logiciel (Cicchetti, Di Ruscio et Pierantonio, 2008; Del Fabro, Bézivin et Valduriez, 2006; Kolovos *et al.*, 2009). Ce domaine de recherche en particulier propose déjà plusieurs solutions dans la représentation des différences entres code sources et, plus récemment, entre modèles conceptuels des logiciels. Le formalisme utilisé pour représenter le modèle proposé provient également du monde de la conception logicielle, soit le langage UML.

Le modèle de représentation des différences basée sur les contraintes géométriques proposé lors de cette quatrième phase constitue le sujet d'un article scientifique en processus de soumission intitulé « *Identification of shape differences between 3D CAD models using geometric constraints. Part I: Difference modeling* ». Le contenu de cet article compose le sixième chapitre de thèse.

2.5 Phase 5 : Procédure de calcul des différences basée sur le modèle proposé

La cinquième phase méthodologique s'appuie sur les connaissances acquises, lors des deux premières phases, sur la nature et les capacités des méthodes de comparaison géométrique existantes. Il s'agit de développer une nouvelle méthode de calcul des différences adaptée à l'exploitation du modèle de représentation des différences par les contraintes géométriques.

Le calcul des différences entre les modèles CAO doit d'abord être géométrique : selon l'un des principaux postulats émis, les données communes entre les deux modèles comparés sont de nature strictement géométrique et non sémantique du point de vue d'un ingénieur mécanicien. La désignation d'une méthode de calcul des différences géométriques entre deux modèles permettant le repérage d'un delta géométrique avec certitude et précision constitue la première partie de la procédure recherchée durant cette phase méthodologique.

La seconde partie de la procédure de calcul des différences doit ensuite élever le niveau sémantique de la représentation des différences depuis le niveau géométrique primaire. Le modèle de représentation des différences établi à la phase méthodologique précédente définit la nature des informations exploitées dans la description de l'évolution; il définit la richesse de la représentation. La procédure de calcul recherchée, quant à elle, doit permettre de trouver ces informations à partir des modèles comparés et de la représentation géométrique des différences; elle garantit la pertinence sémantique de la représentation des différences.

La procédure de calcul proposée est décomposée à la figure 2.2. La comparaison des modèles CAO via leurs représentations par les frontières (B-Rep) est d'abord identifiée à partir des résultats des phases précédentes comme étant la méthode de calcul des différences géométriques à exploiter pour la procédure proposée. En plus de produire des résultats précis, cette méthode permet de produire les associations entre les entités topologiques des modèles comparés.

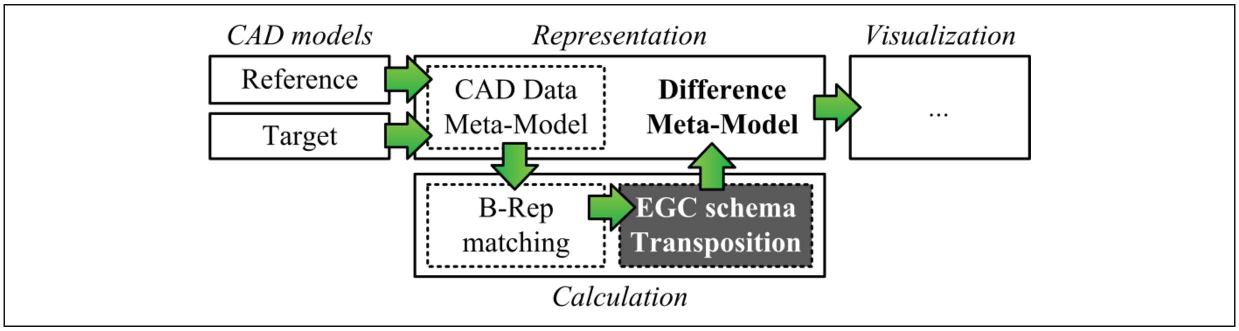


Figure 2.2 Décomposition de la procédure de calcul des différences basées sur les contraintes géométriques

Ces associations sont ensuite nécessaires à la représentation des différences au sein du modèle de représentation proposé et constituent l'information exigée pour la transposition des contraintes géométriques explicites – *explicit geometric constraints* (EGC) *schema transposition*. Le principe de la transposition des contraintes géométriques explicites d'un modèle de référence vers un modèle cible est illustré à la figure 2.3. Les contraintes géométriques explicites sont elles-mêmes des associations entre entités topologiques d'un même modèle incarnant un raisonnement de conception ou de fabrication. En exploitant les associations calculées au niveau géométrique entre les modèles comparés, les contraintes géométriques d'un modèle de référence peuvent alors être reproduites au sein d'un modèle cible différent qui en est dépourvu. Les résultats de cette transposition, i.e., les conditions selon lesquelles le schéma de contraintes a pu être reproduit ou adapté en fonction de la géométrie cible différente, constituent la représentation des différences entre modèles CAO proposée par cette thèse.

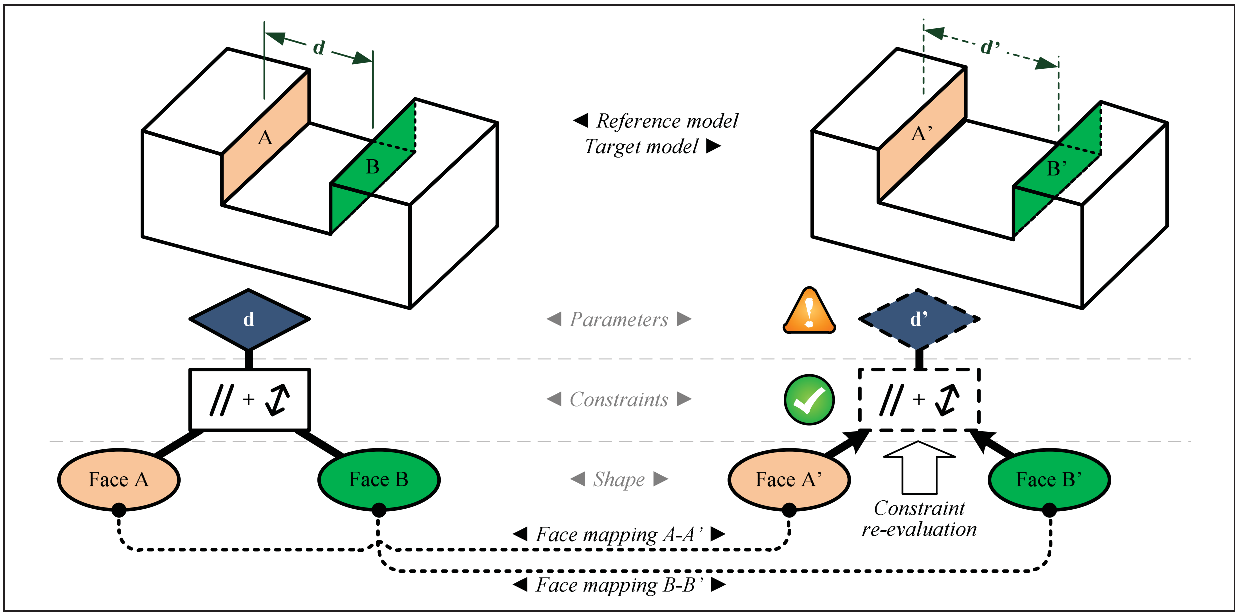


Figure 2.3 Principe de calcul des différences basées sur les contraintes géométriques
Tirée de Brière-Côté, Rivest et Maranzana (2012b)

La procédure de calcul des différences basé sur les contraintes géométriques explicites proposée lors de cette cinquième phase constitue également le sujet d'un article scientifique en processus de soumission intitulé « *Identification of shape differences between 3D CAD models using geometric constraints. Part II: Explicit geometric constraint schema transposition* ». Le contenu de cet article compose le septième chapitre de thèse. L'annexe II présente quant à elle des extraits de schémas ayant servis à la validation des procédures de calcul proposées.

2.6 Phase 6 : Transposition du modèle de représentation au contexte de réutilisation

En tant que dernière phase méthodologique, la contribution de cette thèse au scénario de la représentation et la propagation de l'évolution de la définition géométrique d'un produit mécanique est transposée à un autre scénario impliquant la comparaison géométrique des modèles. Ainsi, le modèle de représentation des différences basée sur les contraintes géométriques est appliqué au problème de la réutilisation du design de pièces et d'ensembles en contexte de développement de nouveaux produits. Ces pièces et ensembles étant en prémisse modélisés au sein de modèles CAO en trois dimensions, on explore l'utilisation de l'approche de comparaison proposée dans le but de trouver des pièces existantes similaires à une référence et d'en caractériser la différence afin d'appuyer les prises de décision subséquentes.

Le but de cette transposition est de valider la pertinence et le potentiel de l'approche proposée. On démontre que celle-ci ne permet pas seulement de résoudre le cas spécifique pris en exemple dans cette thèse, soit la caractérisation de l'évolution de la définition géométrique provenant d'un modèle de conception afin de mettre à jour un modèle cible de fabrication. La généralité de l'approche proposée ouvre la porte à d'autres mises en application et favorise le développement de nouvelles solutions à la problématique de l'identification des différences.

Le scénario de la réutilisation des pièces existantes en développement de nouveaux produits peut impliquer deux sous-scénarios de comparaison des modèles CAO :

- Il y a d'abord la recherche des pièces existantes présentant des caractéristiques géométriques similaires à celles qui sont recherchées pour le nouveau produit. On exploite alors les modèles CAO de ces pièces existantes afin d'effectuer une recherche basée sur la similarité géométrique parmi une large base de données CAO. Étant donné le volume des comparaisons géométriques effectuées lors d'une telle recherche, un compromis est nécessaire au niveau de la représentation des similarités afin d'accélérer les procédures de calcul. Généralement, la similarité est évaluée afin tout simplement d’établir un ordre de grandeur de similarité parmi les nombreux spécimens de l’échantillon de recherche et identifier les quelques meilleurs candidats à la réutilisation.

- Ensuite, afin d’identifier le meilleur candidat parmi le nouvel échantillon plus contingenté, il importe de délaisser les mesures de la similarité relative obtenues lors de la première comparaison pour aller vers la caractérisation des différences. Il ne s’agit plus ici d’une recherche, mais plutôt d’une sélection basée sur ce qui différencie exactement les différents candidats à la réutilisation. Le volume de comparaison étant beaucoup plus petit, la richesse et la précision de la représentation des différences géométriques peuvent être favorisées au détriment de la rapidité du calcul.

Il est démontré que l’approche de comparaison géométrique proposée par cette thèse s’applique très bien au deuxième sous-scénario de comparaison impliqué en réutilisation de pièces. Ce sous-scénario et celui examiné dans nos travaux comportent des similarités importantes :

- La comparaison doit avant tout être géométrique, comme le stipule le second postulat de ces travaux, afin de pouvoir comparer des modèles de provenances diverses et donc potentiellement représentés sous des formats CAO hétérogènes au sein de la base de données CAO.
- Les modèles comparés – le modèle de référence et le modèle candidat – sont détachés, i.e., aucun lien formel et/ou détaillé n’existe entre les modèles avant la comparaison autre que le résultat de la recherche géométrique précédente, ce qui respecte le troisième postulat de cette thèse.
- Il est avantageux de décrire la géométrie du modèle candidat du point de vue spécifique du schéma de cotation du modèle de référence qui constitue la cible à atteindre en termes de spécifications géométriques à retrouver.

Au final, le calcul et la représentation des différences basés sur les contraintes géométriques permet en effet de repérer et de représenter avec précision et de manière détaillée les différences des modèles des meilleurs candidats à la réutilisation par rapport au modèle de référence. Les résultats détaillés et sémantiques du point de vue du concepteur ainsi obtenus permettent d'analyser l'impact de la réutilisation de chacune des pièces candidates dans le développement du nouveau produit – un peu comme on évalue l'impact d'une modification sur le processus de conception d'une pièce – et de prendre de meilleures décisions.

La transposition de l'approche de comparaison des modèles CAO proposée dans ces travaux au scénario de la réutilisation des pièces et ensembles existants durant le développement de nouveaux produits réalisée lors de cette sixième et dernière phase a été présentée lors d'une conférence internationale. Elle constitue le sujet d'un article de conférence intitulé « *Integration of explicit geometric constraints in the comparison of 3D CAD models for part design reuse* »; le contenu de cet article compose le huitième chapitre de cette thèse.

CHAPITRE 3

COMPARING 3D CAD MODELS: USES, METHODS, TOOLS AND PERSPECTIVES

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article publié dans la revue « *Computer-Aided Design and Applications* »
en novembre 2012.

3.1 Abstract

With the advancements of 3D modeling software, the use of 3D CAD in mechanical product design has become a standard practice. Methods and tools are continually being developed to improve designers' efficiency in the creation, modification and analysis of 3D CAD models. Among other advantages, comparing 3D CAD models to assess their relative shape similarity or to identify their differences leads to benefits in various CAD- and PLM-related areas such as design reuse, engineering change management and data exchange. As 3D data continues to be more frequently and intensively shared and used in the mechanical product development process (PDP), this paper describes the subject of 3D CAD model comparison from three related points of view. First, it organizes the wide variety of use cases for 3D CAD model comparison into specific application domains. Difference calculation methods and approaches are compared, identifying their key characteristics and limitations. Then, it presents an inventory of commercially available software tools that perform 3D CAD model difference identification (MDI). Finally, some research perspectives for 3D CAD model comparison applied to shape change transposition are contemplated.

3.2 Introduction

Today’s product lifecycle management (PLM) solutions address the contemporary challenges of collaborative and integrated product development. The management, storage and distribution of the geometric definition of a product relies on 3D data, data that is being more frequently and intensively shared and used than ever before. For example, 3D CAD models are increasingly used as inputs to retrieve and compare products, parts and related information from PLM vaults to ultimately enable product data reuse, and/or to otherwise leverage the knowledge associated with this type of document and the data it encloses.

Three-dimensional (3D) CAD model comparison is defined here as the process of calculating and representing the differences or similarities between 3D CAD models embodying the geometric definition of mechanical parts. 3D CAD model comparison has been the focus of several advancements in the last decade, notably in the field of 3D shape-based retrieval (Cardone, Gupta et Karnik, 2003; Iyer *et al.*, 2005; Li, Liu et Ramani, 2004; Tangelder et Veltkamp, 2004; Yang, Lin et Zhang, 2007). In contrast, developments regarding the pair-wise comparison of 3D CAD models designed for the location and documentation of differences have remained sparse, mostly originating from standardization schemes (CAx Implementor Forum, 2008; Frechette, 1996) or 3D CAD software developments (CoCreate Software GmbH et Gutierrez, 3 juin 2010). However, the process of comparing 3D CAD models does bring a variety of benefits to multiple scenarios in the development of mechanical products.

This paper falls within the framework of a research project that addresses the subject of shape change transposition between heterogeneously formatted 3D CAD models. For example, in Figure 3.1, an initial reference model released as a STEP file from Design Engineering defines a part’s original geometry. An initial target CAD/CAM model is created by Manufacturing Engineering as per the initial reference model in a format deemed appropriate for manufacturing planning (e.g. procedural modeling such as NX® (Siemens PLM Software inc., 2009a)). Then, an engineering change order (ECO) calls for the release of a modified reference model, derived from the initial reference model. To derive a modified target CAD/CAM model from the initial target model, Manufacturing Engineering must therefore identify the exact shape change through model comparison and transpose this shape change in the manufacturing domain.

The global objective of this research project is to develop a 3D CAD model comparison method for shape change transposition, in which the representation of the differences is intended to optimize their interpretation and their integration by the process responsible for updating the initial target model and, thus, deriving the modified target model. Previous work on the identification and representation of 3D CAD model comparison scenarios like this one is described by Brière-Côté, Rivest and Maranzana (2011).

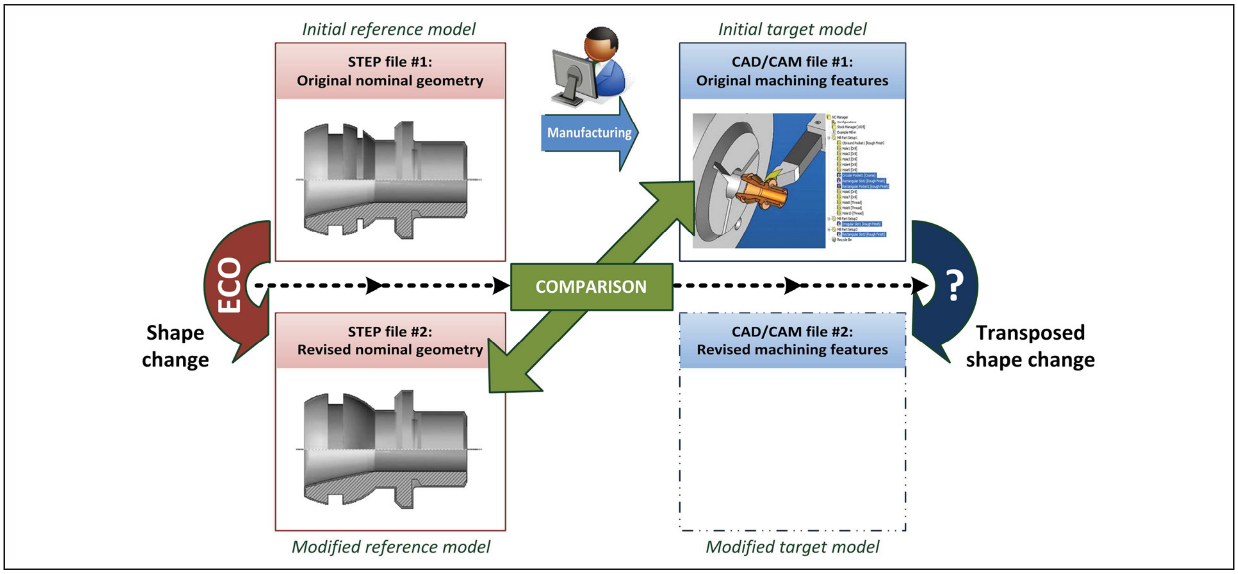


Figure 3.1 Sample scenario of shape change transposition: Updating a CAD/CAM model based on revised geometry

With the specific objective of identifying and organizing existing literature and recent developments in this promising CAD- and PLM-related domain and, thereby, of laying the groundwork for our research project, this paper presents a comprehensive review of current research and developments in 3D CAD model comparison from three perspectives: the uses, the software tools and the methods. Section 3.3 presents a survey of 3D CAD model comparison scenarios, categorizing different use cases. Difference calculation methods are characterized in Section 3.4. Pair-wise 3D CAD model comparison is then the focus, as an inventory of existing software tools implementing 3D CAD model difference identification (MDI) is described in Section 3.5. Finally, research perspectives are contemplated in Section 3.6.

3.3 Use cases of 3D CAD model comparison

As the first part of this article, an exploratory survey makes a broad inventory of scenarios involving the comparison of 3D CAD models as they are exposed in CAD- and PLM-related documentation and research literature. As summarized in Table 3.1, 3D CAD model comparison use cases can be organized into six major application domains and related to three solution domains: (1) shape-based retrieval, (2) equivalence/similarity assessment, and (3) difference identification.

Tableau 3.1 Application domains, use cases and solution domains for 3D CAD model comparison

		Solution domain		
Application domain	*Use case*	*Shape-based retrieval*	*Equivalence/similarity assessment*	*Difference identification*
Product information reuse	▪Design concept ▪Manufacturing process ▪Simulation/analysis data ▪Pricing information ▪Sourcing information ▪Qualification tests results	X		X
Product rationalization and standardization	▪Eliminate duplicate parts ▪Improve sourcing ▪Form part/product families ▪Find interchangeable parts ▪Identify common platforms ▪Identify differentiation enablers ▪Verify interchangeability	X	X	X
CAD modeling management	▪Prevent model duplication ▪Promote modeling best-practices	X		
CAD data translation/remastering	▪CAD migration ▪CAD data exchange ▪CAD interoperability ▪Long term archival		X	
CAx models authoring	▪CAM models ▪FEA models		X	
Engineering change management	▪Change documentation ▪Impact analysis ▪Change transposition ▪Change propagation ▪Evolution control			X

3.3.1 Identifying the solution domains

As pictured in Figure 3.2, the three solution domains for the inventoried use cases were determined, based on the two key aspects that characterize 3D CAD model comparison problems:

- *Cardinality* – a reference model may either be compared to one single target model (1:1, or pair-wise) or to many models (1:n) usually from large sets; and
- *Level of detail* – the amount of information expected from the comparison, which will vary according to the intended use, ranging from a simple “Yes-No” or “Passed-Failed” diagnosis to detailed measures of the differences between the compared models.

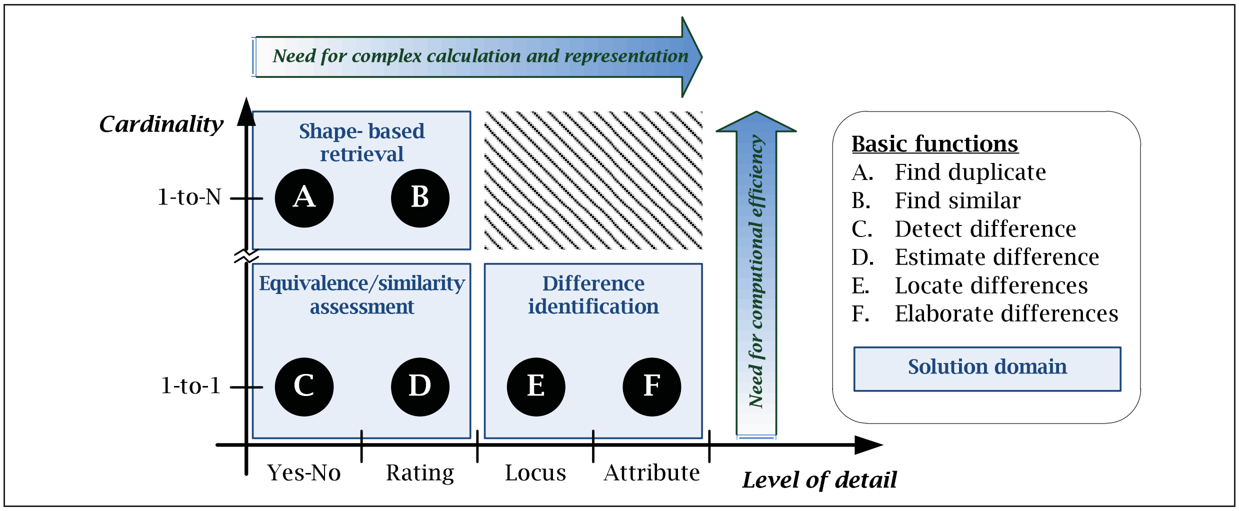


Figure 3.2 Solution domains and basic functions as a relation between the required level of detail and cardinality

Higher cardinalities require higher computational efficiency, since it regulates the quantity of comparisons to be performed in a single operation. Similarly, higher levels of details understandably call for more detailed difference calculation algorithms and, consequently, more complex difference representation schemes. Accordingly, we defined a relation between the level of detail and the cardinality, allowing us to identify six different basic functions for comparing 3D CAD models (see Table 3.2). Each basic function relates to an elementary question that the 3D CAD models comparison is expected to answer, and provides insight on what type of result is expected.

Tableau 3.2 Basic functions of 3D CAD model comparison

	Basic function	*Basic question*	*Expected result*
A.	Find duplicate	Which models are equivalent?	Finite sets of objects
B.	Find similar	Which models are similar?	Ordered, scale-based distributions
C.	Detect difference	Are the models different?	Binary results (Yes/No, Pass/Fail, etc.)
D.	Estimate difference	How different are the models?	Qualitative, global, scale-based measures
E.	Locate differences	Where are the differences?	Graphical reports, loci, regions
F.	Elaborate differences	What are the differences?	Classifications, local measures, descriptions

The three solution domains organize these basic functions. Verifying two models' equivalency according to some explicit criteria or estimating their relative similarity with use of a metric, i.e. providing a qualitative appraisal of how close or different they are from each other, involves equivalence or similarity assessment. Shape-based retrieval achieves finding duplicate or similar models with the use of the comparable similarity measures, but within large sets of 3D CAD models simultaneously. Finally, to distinguish individual differences between two models, providing their respective locus in relation with the modeled shapes or, furthermore, a detailed description of their characteristics, we refer to the identification of the models' differences.

3.3.2 Product information reuse

To achieve the desired goal of reducing costs and delays while lowering risk, one aspect of PLM features the retrieval and reuse of parts, products and associated information. 3D CAD model comparison is key to overcoming the challenge, as shape itself can now be perceived as a neutral and effective language to represent and retrieve product data in PLM vaults (Li, Liu et Ramani, 2004).

It is now possible to quickly compare quite large numbers of 3D shapes to each other through the use of lightweight pre-computed shape descriptors or signatures, as many research and survey papers have so indicated (Cardone, Gupta et Karnik, 2003; Iyer *et al.*, 2005; Li, Liu et Ramani, 2004; Tangelder et Veltkamp, 2004; Yang, Lin et Zhang, 2007). A 3D CAD model selected as a search key can be compared to other models in order to assess their similarity via a given metric which, in turn, will be used to identify, rate and sort a subset of the compared models that can be considered similar. Particularly, feature-based techniques for shape-based retrieval, such as those described by Cicirello and Regli (2001), Bai et al. (2009) or Chu and Hsu (2006), enable part and/or model function to be involved in the similarity assessment of candidate models, which is a considerable aspect in product information reuse.

Most of the published works present the reuse of existing product information as the main application of their respective approaches to shape-based retrieval. A benchmark on product design reuse conducted by the Aberdeen Group (Jackson et Buxton, 2007) validates this trend by revealing that, while 46% of the surveyed manufacturing organizations identify information retrieval as a challenge, the best-in-class organizations in terms of reuse are three times more likely to have made use of shape-based searches. Msaaf, Maranzana and Rivest (2007) reported on the potential uses for shape-based retrieval tools for product information reuse, as confirmed by industrial users. Examples include:

- Searching for existing parts to be reused as-is or with minor modifications in a new design;
- Searching for existing designs to reuse their simulation/analysis contents in the development of a new product;
- Searching for parts that are similar to a new design in order to reuse/adapt their manufacturing processes; and
- Searching for similar parts to compile comparative data on manufacturing costs and thereby make better estimates.

In many cases, the solution for better product information reuse is not limited to the use of shape-based retrieval tools. As Msaaf, Maranzana and Rivest (2007) point out, the problem must be divided into two steps: (1) a search for similar parts, and (2) an identification of the differences between the retrieved similar parts. The purpose of shape-based retrieval is generally limited to identifying sets of CAD models that can be considered as candidates for product information reuse in numerous scenarios. To determine if two CAD models can be identified as similar in a particular use context – e.g., if the design of an existing similar part can successfully be reused in a new project – a pair-wise comparison of each retrieved candidate model with the original search key is mandatory.

For instance, as prescribed in the Cessna Aircraft Company's Supplier Guidelines and Requirements for Engineering Certification Projects (2010), a supplier may reuse the qualification data of an approved similar design to support the qualification of a new design submitted to the OEM's engineering department. This so-called "Qualification by Similarity" procedure, however, requires that "a detailed comparison of the two parts shall be provided that identifies the specific differences and the justification for why the differences meet the requirements of the specification and allow the determination of compliance".

Furthermore, in the case of manufacturing processes' reuse, Huang, Tian and Zhou (2004) proposed an approach to facilitate the interoperation between new part designs and existing manufacturing processes. Two constructive solid geometry (CSG) models, one defining a part's geometry in the design space and the other defining the capability envelope set of a parametric machining process, are compared to find their equivalent parameter domains.

3.3.3 Product rationalization and standardization

Product rationalization and standardization can benefit from the use of shape-based retrieval tools such as in developing families of parts and products (Wei et Yuanjun, 2008). Identical parts from separate projects can be located and regrouped to reduce the management and manufacturing costs, particularly by manufacturing in larger batches (Msaaf, Maranzana et Rivest, 2007). Sourcing can also be improved by subcontracting regrouped similar parts to a reduced number of suppliers.

Use of similarity measures like those used in shape-based retrieval have also been reported for low-cardinality comparison in the composition of product families. Viswanathan, Chowdhury and Siddique (2008) presented a commonality measure for component pairs within similar product models based on feature-pair's dimensions and positions. Chowdhury and Siddique (2009) then implemented that commonality measure to identify a common platform from 3D CAD models of vacuum cleaners. Accordingly, following the example of product information reuse, product standardization can be considered as a two-step process. After candidate parts or products to be grouped into families have been identified, one further step to this process is to identify and validate the new product family's common platform and differentiation enablers through pair-wise comparison.

3.3.4 CAD modeling management

Shape-based retrieval can also be applied in the management of CAD modeling within a manufacturing organization. Msaaf, Maranzana and Rivest (2007) described the use of shape-based retrieval tools in the promotion of modeling best-practices. By locating existing design models similar to a 3D draft, the reuse of best modeling practices ensures continuity in CAD model construction methods. The Parametric Technology Corporation (23 septembre 2003) introduced a method for comparing CAD models to prevent model duplication. As new models are stored in the PLM vault, existing models with near-identical shape properties are retrieved and designers alerted as to potential duplication.

3.3.5 CAD data translation/remastering

A fourth group of 3D CAD model comparison scenarios involves the use of comparison tools to support the geometric validation of translated or remastered 3D CAD models. The objective of the comparison is to verify the geometrical equivalency of two closely related 3D CAD models, i.e. models intended to represent and communicate the same geometric definition of a part.

From the perspective of 3D CAD model comparison, 3D CAD data translation and data remastering can be regarded as the same process. In both cases, a representation of product data, mostly concerning the shape, is fully or partially transferred from a source data format to a target data format, with the possibility for product data degradation during such transfers. They simply differ on how they are executed – translation being automated while remastering is manual – and thus on the sources of degradation/difference between the source and target 3D CAD files. The purpose of the ensuing validation process remains the same: to ensure that the authority data from the source file is accurately accounted for in the target CAD file.

The validation of 3D CAD data translation/remastering by comparing source and target CAD files has become a fundamental procedure in industry when matters of 3D CAD data migration, interoperability or long-term archival arise. Working groups such as AutoSTEP (automotive) (Frechette, 1996), LOTAR International (aerospace) (Zuray et Delaunay, 2009) and the CAx Implementor Forum (software) (2008) have either issued recommendations for best practices regarding geometric validation or developed auditable processes in which 3D CAD data translation validation is key and involve geometric comparison. As a result, provisions for the geometric validation of translated CAD data have been included in recent versions of application protocols AP203 (International Organisation for Standardization, 2011) and AP214 (International Organisation for Standardization, 2010) of the ISO STEP standard for product data exchange.

3.3.6 CAx model authoring

Another validation process benefiting from 3D model comparison is the validation of downstream 3D CAx models, whose construction often relies on a master 3D CAD model. For example, the modeled geometry of a part in a 3D computer-aided manufacturing (CAM) model has to be equivalent or consistent with the geometry of the originating 3D CAD model, and comparing both models' geometries constitutes a means to validate the 3D CAM model for use in downstream processes.

Three-dimensional finite-element analysis (FEA) models offer a similar case, in which the initial geometry of a part is regularly simplified for computation purposes. Li and Liu (2002) proposed a methodology for abstracting detailed features of a 3D solid model and using dissimilarity metrics between the original model and the computed simplified model to assess the level of simplification. Lockett and Guenov (2008) described a similar situation where two similarity measures can be used to evaluate the quality of a mid-surface model – used for the analysis of thin-walled parts – by comparing it to a solid model of the same part.

3.3.7 Engineering Change Management

The sixth and last application domain for 3D CAD model comparison is engineering change management. As emphasized by 3D CAD model comparison software editors (e.g. CT CoreTechnologies Group (2008) , ITI TranscenData (2010a), CapVidia NV (2010), Kubotek USA (2010)) and demonstrated by the research of Chatelain, Maranzana and St-Martin (2002) and Al-Sabeh (2004), 3D CAD model comparison can be used to identify and document modifications applied to a part or product model between revisions, and by doing so, support the elaboration of engineering change orders (ECO). The impact of a part or product models' evolution on downstream models, such as process plans, NC programs or simulation models, can thus be determined more accurately and dealt with more efficiently. Likewise, unauthorized engineering changes can be properly detected and managed.

The recognition of a part's evolution through the comparison of its 3D CAD model's versions also opens the way to the computer-assisted transposition of shape changes from the definition model to the different downstream CAx models. An example of shape change transposition is the remeshing of FEA models following the modification of the master 3D CAD model's geometry. François and Cuillière (2000) presented a 3D automatic remeshing algorithm based on the preliminary location of geometric modifications between the original mesh and the revised 3D CAD model by means of octree structures. Cuillière *et al.* (2009) and Souaissa *et al.* (2010) presented a similar remeshing algorithm in which the comparison of the two 3D CAD models is performed via the tensor-based matching of boundary representation (B-Rep) entities.

Sheffer and Ungor (2001) presented a different approach where geometric modifications are expressed in the form of parametric changes between parametric procedural representations of the 3D CAD models. Similarly, Sypkens Smit and Bronsvoort (2007; 2009a) envisioned a way to describe the difference between two models from the point of view of each shape feature to reuse subparts of an original mesh more efficiently. Their approach constructs such a description through the use of cellular representations of the part models to store the persistence qualifications for each feature.

3.3.8 Synthesis

By organizing the various inventoried use cases for 3D CAD model comparison in six distinct application domains and in close relation with the three identified solution domains, we are able to draw some useful insights:

- As opposed to shape-based retrieval, which has been the focus of numerous research papers and reviews, developments regarding pair-wise 3D CAD model comparison have remained sparse. Even so, several use cases can be identified, and some are complementary to shape-based retrieval initiatives.
- Applications for 3D CAD model comparison in relation to engineering change management require a significant level of detail in terms of the description of 3D CAD model differences; therefore, model difference identification solutions represent a most promising avenue with which to address our concerns about shape change transposition.

For this latter reason, the remainder of this paper focuses primarily on the solution domain of 3D CAD model difference identification (MDI). Accordingly, the next sections describe the specific problem of calculating and representing geometric differences between two 3D CAD models from two other perspectives: the methods used to locate and measure the geometric differences between pairs of 3D CAD models and the tools implementing MDI capabilities.

3.4 Characterizing 3D CAD difference calculation methods

The problem of determining 3D CAD model differences is intrinsically complex. The overall problem can be separated into three phases (Kolovos *et al.*, 2009):

- *calculation*, a procedure, method or algorithm able to compare two distinct 3D CAD models, i.e. identifying the mappings and, then, the differences between them;

- *representation*, the outcome of the calculation must be represented in some form that is amenable to further manipulations; and
- *visualization*, model differences often need to be presented according to a specific need or scope, highlighting those pieces of information that are relevant only for the prescribed goal.

Calculation and representation are the central ingredients for any pair-wise comparison solution. In this section, we focus on the former aspect by describing and categorizing several difference calculation approaches. As for difference representation, it is highly dependent on the calculation method and, therefore, on the 3D CAD representation scheme used for comparison, as they define the type of data to be represented and manipulated.

3.4.1 3D CAD representations used for comparison

First, we briefly summarize the different 3D CAD representation schemes that are either implicit to or that can be derived from 3D CAD models. The difference calculation methods presented in this section operate on particular representations of 3D CAD data and are therefore fundamentally affected by their aspects, such as applicability, efficiency and accuracy.

3.4.1.1 Procedural representation

Most of today's 3D CAD systems implement the feature-based parametric solid modeling paradigm. Modeling operations are stored as features and organized sequentially in a tree, maintaining a parent/child relationship. Creating a model implies instantiating a data structure comparable to a program, and the expected geometry is thereby represented implicitly. Everything related to the development of the model, including 2D sketches, 2D parameters on the sketches, 3D operations and the parameters related to the 3D operations, is recorded. Some edits are done by accessing one of the previous operations and adjusting a sketch or a parameter, while other modifications involve adding, removing, reordering or replacing one or several previous operations. After adjustments have been made, the "program" can be replayed to evaluate a different geometric model.

However, issues related to the non-uniqueness of modeling sequences in representing solids, as illustrated in Figure 3.3, greatly reduce (beforehand) the relevancy of difference calculation methods based on this particular representation scheme (see Table 3.3) in uses other than those related to version comparison for a given 3D CAD model.

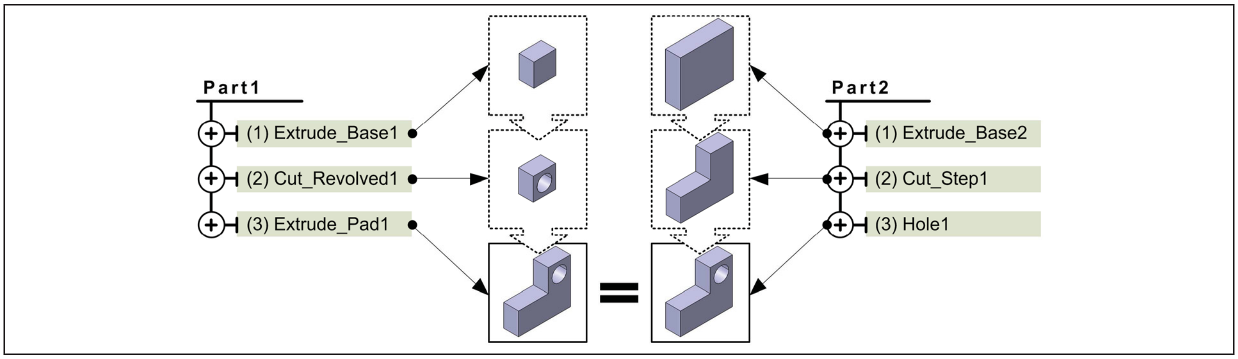


Figure 3.3 Non-uniqueness of the procedural representation of solids

3.4.1.2 Boundary representation (B-Rep)

Along with the construction history, the parametric feature-based solid modeling paradigm also relies on the boundary representation of solids, more specifically to represent the resulting explicit geometry. The B-Rep data structure stores basic elements composing the boundary of a solid, such as the vertices, the edges, and the faces, with the information about how they are connected. The data in any B-Rep data structure can be classified as either (Lee, 1999, p. 119):

- *geometry information*, which generally refers to information about surface equations (faces), curve equations (edges) or point coordinates (vertices); or
- *topology information*, which refers to the interrelationships among faces, edges, and vertices.

3.4.1.3 Planar facet tessellations

Planar facet tessellations are a specialized type of boundary representation in which points, straight lines and planes are the only geometric entity types allowed for vertices, edges and faces, respectively. Consequently, only point coordinates need to be explicitly specified and stored in the data structure as line and plane equations are computed from neighboring point data. The result is a computationally lightweight representation of 3D shape data broadly used in computer graphics and, thus, for 3D CAD visualization. Digital mock-ups (DMU) typically use this representation scheme for the visualization and the spatial analysis of large assemblies of parts (Lee, 1999).

The evaluation of a planar facet tessellation from a 3D CAD dataset involves the approximation of the explicit modeled shape by means of triangular planar facets. Adjustable parameters regulating the shape approximation, such as the chord tolerance, need to be monitored to ensure a reasonable level of accuracy for the ensuing processes.

3.4.1.4 Decomposition models

Characterized as auxiliary representations of 3D CAD models, decomposition models approximate 3D shapes by means of their spatial occupancy in the 3D space. For example, voxels are uniformly-sized cubic cells obtained via the 3D rasterization of a given space encompassing a solid. The size of the voxels determines how closely the voxel representation approximates the modeled solid.

Since the memory space required to store the voxel representation increases dramatically as the size of the voxels decreases, one may opt instead for the *octree* representation. An *octree* representation is similar to a voxel representation in that it represents a solid as an aggregate of hexahedra, but it reduces the memory requirement considerably by dividing the space differently and representing it in a navigable tree-like data structure. Cellular representation is also a method of representing a solid as an aggregate of simple cells. However, it does not impose a strict restriction on the allowable shape of the cells to be used (Lee, 1999).

3.4.2 Geometric comparison

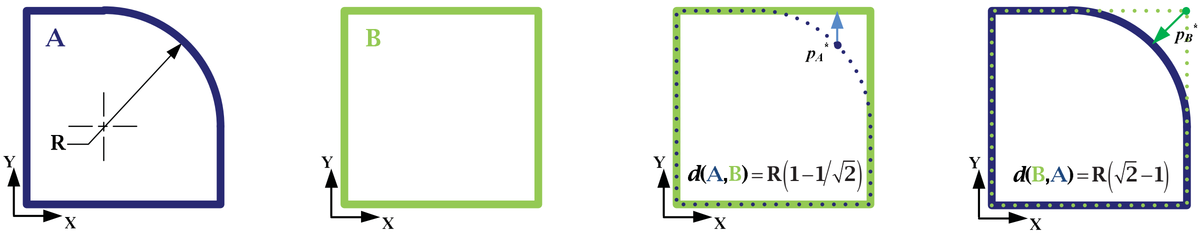
3.4.2.1 Global geometric properties

The comparison of global geometric properties such as 3D CAD models' volumes, surface areas, moments of inertia, etc., allows for the quick and straightforward validation of two models' geometric equivalency. For such metrics, tolerances are required in order to establish a threshold on the acceptable difference expressed in percentages between the compared models. However, no information on the nature and locus of the differences between the compared shapes can be provided, nor do the differences offer much significance in terms of functional part features, such as measuring the distance between two faces, for example.

Only results with a low level of details (yes-no, rating) can therefore be achieved using this particular approach. However, the very low computation cost of these difference calculation algorithms enables their use in scenarios such as detecting duplicate 3D CAD models in large PLM vaults (Parametric Technology Corporation, 23 septembre 2003). Also, global geometric properties are being implemented as "Geometric Validation Properties" (GVP) (CAx Implementor Forum, 2008) in application protocols AP203 (International Organisation for Standardization, 2011) and AP214 (International Organisation for Standardization, 2010) of the ISO STEP standard for product data exchange.

3.4.2.2 Point-to-part deviation

Also called the "Cloud Of PointS" (COPS) mechanism (CAx Implementor Forum, 2008), the point-to-part difference calculation method is based on the evaluation of the *Hausdorff* metric (Aspert, Santa-Cruz et Ebrahimi, 2002) which, fundamentally, measures how far two subsets of a metric space are from each other. The surface of the first 3D CAD model is discretized by a set of sampling points. For each sampling point, a deviation is calculated as the smallest distance separating it from the surface of the second 3D CAD model. The highest deviation from all sampling points denotes the *forward Hausdorff* distance. The *backward Hausdorff* distance is calculated the same way, i.e. with the sampling points lying on the surface of the second 3D CAD model, since both distances will not always equate ($d(A,B) \neq d(B,A)$), as demonstrated in two dimensions by Figure 3.4. The highest value between the two directed Hausdorff distances determines the Hausdorff metric; accordingly, the corresponding sampling point identifies the locus of the global maximal deviation on the models' surfaces.



(a) (b) (c) (d)

(a) Part A with a round feature;
(b) Part B with sharp corners;
(c) Forward Hausdorff distance calculated between sampling points from the boundary of part A and the boundary of part B;
(d) Backward Hausdorff distance calculated between sampling points from the boundary of part B and the boundary of part A.

Figure 3.4 Illustrating the Hausdorff metric calculated between two parts' boundaries (in 2D)

The sampling domains, i.e. the 3D CAD model surfaces, may be partitioned into a number of smaller subdomains prior to the distance computation to enable the location of multiple local deviation maxima on the models' surfaces. Typically, face entities from a B-Rep data structure are used as sampling subdomains (CapVidia NV, 2010; ITI TranscenData, 2010a). Heuristics may also be used to distinguish several local deviation maxima and their neighboring regions among the sampling points after the distance computations (Lattice Technology Co., 2010).

A tolerance must be applied to the calculated deviations in order to identify the point-to-part deviations sizeable enough to be considered as relevant geometric differences. Color scales are frequently used for the visualization of point-to-part deviation results on the surface of the modeled 3D shapes (CADCAM-E.Com, 2010; CT CoreTechonologies Group, 2008), following the example of point-cloud-based inspection software tools (InnovMetric Software Inc., 2009). Point-to-part deviation calculation is typically a pair-wise comparison method, as distance computation is computationally expensive, and one of the few capable of measuring multiple differences.

Rapid distance computation constitutes a problem on its own, as taken up by the related research field of computational geometry. When transposed in a 3D CAD setting, some proposed distance computation solutions introduce novel uses for decomposition models. For example, Tsai (2002) presents two rapid and simple algorithms for approximating the distance function for given isolated points on uniform grids that can be derived from voxel representations. Also, Pottman, Leopoldseder and Zhao (2003) propose the *d^2-tree*, an octree data structure which stores, in each of its cells, a local quadratic approximant of the squared-distance function of a geometric object. Bearing in mind that these distance computation algorithms yield approximate results, their respective use in an MDI context requires the complementary use of refining algorithms.

3.4.2.3 Spatial occupancy comparison

As 3D CAD models fundamentally embody the definition of 3D geometric shapes, i.e. subsets of the 3D Euclidean space, some difference calculation methods focus on spatial occupancy. The calculation of the regularized Boolean operations from set theory between the two modeled solids constitutes another approach to compare two 3D CAD models. Given two subsets A and B representing the modeled solids from the reference and target 3D CAD models, respectively, the corresponding difference calculation methods essentially aim at evaluating the three mutually exclusive subsets given by the regularized difference operations ($A -^* B$) and ($B -^* A$), and by the regularized intersection operation ($A \cap^* B$), as pictured in the Venn diagram of Figure 3.5. In the context of mechanical CAD model comparison, these three subsets symbolize regions of material removal, material addition (with respect to the reference part), and material common to the two parts, respectively. Accordingly, spatial occupancy comparison is typically used to mainly locate and visualize differences.

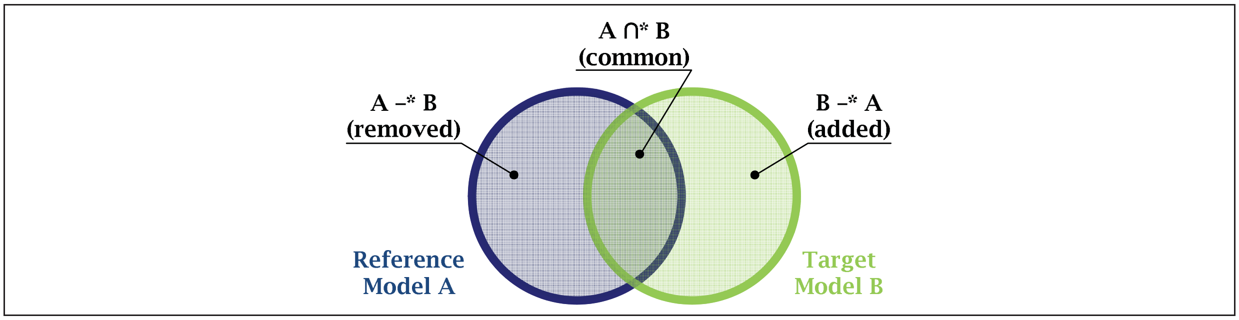


Figure 3.5 Illustration of the calculation of regularized Boolean operations for comparing 3D CAD models

Most 3D CAD systems implement a geometric modeling kernel which includes the basic Boolean operations that perform the regularized union, intersection and difference of 3D objects. These operators may either be used as-is or through automation to compare the volumes of two modeled parts and identify regions of material removal, material addition and common material.

Dassault Systèmes SolidWorks® (2010) provides an automated implementation of such a calculation method. The outcome of the operation is three sets of editable solids, each resulting from the calculation of either of the two directed set differences or of the set intersection of the two input solids. The robustness of this method is directly related to the modeling kernel's robustness in the calculation of set operations for compared solids displaying issue-prone features like coincident boundaries or boundaries intersecting at vanishing angles, which are commonly encountered between similar parts.

The use of auxiliary representations, such as decomposition models, to obtain the result of regularized Boolean operations on two solids can increase computational efficiency. For example, representing both solids simultaneously by means of voxels enables the calculation of the Boolean operations on the integer values of 1 (overlapping solid) or 0 (empty) for the corresponding voxels of the two solids (Dassault Systèmes, 2007; François et Cuillière, 2000).

The superposition of 3D shapes constitutes a faster and more straightforward approach to locate regions of material removal, material addition and common material between two similar 3D CAD models. DMU analysis tools usually implement 3D shape superposition to compare parts, since the corresponding algorithms are similar to those used to verify part mating and clearance and detect part interference in assemblies. Different colors are used to identify common intersecting and differentiating regions on the modeled parts, as shown in Figure 3.6. Examples of software tools that implement the superposition approach are CATIA® V5 (Dassault Systèmes, 2007) and some 3D CAD visualization tools such as Oracle AutoVue® (Oracle Corporation, 2010), Lattice Technology XVL Studio® (Lattice Technology Co., 2010) and Actify SpinFireTM (Actify inc., 2010).

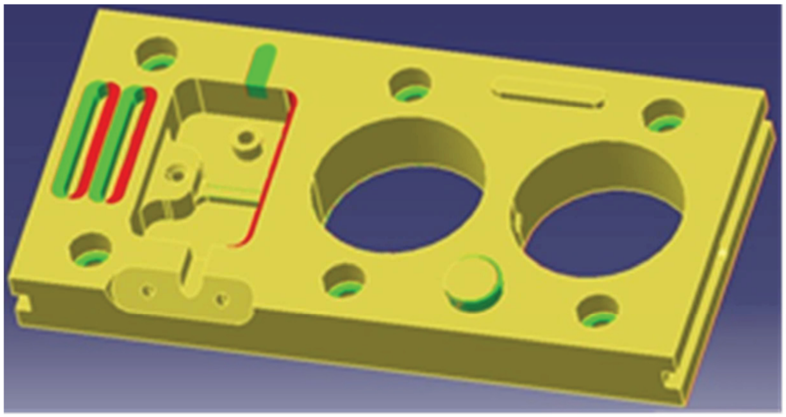


Figure 3.6 Superposition of 3D shapes using CATIA® V5 *DMU Space Analysis* workbench. From Dassault Systèmes (2007)

3.4.3 Data structure matching and comparison

A second 3D CAD difference calculation approach consists of comparing 3D CAD models by means of their implicit data structures. Data structure matching and comparison distinguishes itself from geometric comparison since it fundamentally processes representation data structures instead of computed geometric results. Any CAD representation scheme can be the subject of this type of matching algorithm: CSG (e.g. Chatelain, Maranzana et St-Martin (2002)), B-Rep (e.g. CoCreate Software GmbH et

Gutierrez (3 juin 2010)), parametric feature-based (e.g. Parametric Technology Corporation (2009)), cellular representation (e.g. Sypkens Smit et Bronsvoort (2007)), etc.

For example, comparing two 3D CAD models' procedural representations or "programs", will locate differences in the nature, the quantity and the order of the operations used to build the models, along with the differences at the operations' parametric level. Difference calculation between B-Rep data structures of two 3D CAD models essentially involves the matching of boundary entities with regard to their respective explicit geometry and/or topological information. Entities are then classified according to their own and their neighboring entities' matching status as per specified criteria. For instance, a pair of matched faces having the same surface equation but different boundary edges may be classified as "relimited" or "affected" faces.

The most intricate part of the calculation task is the matching, or mapping, of specific model elements. Following the example of model matching approaches used in Model-Driven Software Engineering (MDSE) (Kolovos *et al.*, 2009), four matching approaches for 3D CAD data structures have been identified.

3.4.3.1 Static identity-based matching

Assuming that each model element has a persistent and unique identifier that is assigned to it upon creation, matching model elements are identified based on their corresponding identities. This approach must involve 3D CAD models that were constructed dependently of each other for persistent identifier to be available in both models. Also, it cannot be applied to 3D CAD formats that do not support maintenance of persistent unique identifiers.

PTC's Pro/ENGINEER® (Parametric Technology Corporation, 2009) offers MDI functionalities for comparing proprietary procedural models which perform static identity-based matching of modeling operations and parameters. Al-Sabeh (2004) presents a similar algorithm using persistent identifiers of modeling features only for the comparison of CATIA V5 (Dassault Systèmes, 2007) part models. Sypkens Smit and Bronsvoort (2007) exploit the mapping of persistent features between two versions of a feature model. The cellular representation of the feature models enables them to compare each feature pair's spatial occupancy independently from their interaction with other features.

3.4.3.2 Signature-based matching

The identity of each model element on which their true/false matching relies on is no longer static; rather, it is a signature assembled from the values of its attributes and properties, or calculated dynamically by means of a pre-defined function. For 3D CAD model elements, geometric data may be used to calculate the signature just as much as descriptive data (e.g. names of feature instances). Signature-based matching can therefore be used to compare isolated models, but first, a series of functions to create the signatures of different types of model elements needs to be specified.

One of Dassault Systèmes' SolidWorks® (2010) available comparison functions uses the signature-based approach, matching modeling operations between two construction histories by their (editable) names. The 3DComparator tool described by Msaaf, Maranzana et Rivest (2007) extracts geometric signatures for faces and matches them to distinguish equivalent faces from different faces between two B-Rep models.

3.4.3.3 Similarity-based matching

As opposed to treating the problem of model matching as true/false identity, the 3D CAD models' data structures are treated as typed attributed graphs and matching elements are identified based on the aggregated similarity of their attributes and geometric properties. As not all characteristics of model elements are equally important for model matching, similarity-based algorithms typically need to be provided with relative weights for each characteristic, and thus, must often undergo an empirical trial and error fine-tuning process.

For example, Cuillière *et al.* (2009) and Souaissa *et al.* (2010) presented a model comparison algorithm that computes the metric tensor, inertia tensor and barycenter of each face's and each edge's respective set of control points from two NURBS-based B-Rep models. Faces and edges from the two models are then matched gradually by verifying the equivalency of their computed properties in a particular order – which equates to giving each property a different weight in the overall similarity measure function.

Chatelain, Maranzana and St-Martin (2002) proceed likewise to compare CSG models. Primitive solids are characterized by their type, transformation matrix, dimensional parameters and position index in the CSG tree, respectively. Then, they are recurrently matched between models with respect to reducing subsets of the characteristics, resulting in decreasing degrees of similarity between matched primitive solids.

3.4.3.4 Syntax-specific matching

Syntax-specific matching algorithms incorporate the semantics of the target 3D CAD representation scheme or format, thereby providing more accurate results and also drastically reducing the search space. For example, in comparing B-Rep data structures, Pan *et al.* (2011) incorporate the knowledge that it only makes sense to compare two edges if the faces they are adjacent to are already known to match; thus reducing the number of model element comparisons that need to be performed.

Similarly, PTC's CoCreate® Modeling (CoCreate Software GmbH, 2008; CoCreate Software GmbH et Gutierrez, 3 juin 2010) implements a B-Rep difference calculation algorithm in which vertex, edge and face mappings are computed progressively using geometric attributes and the mappings from lower-order topological entities simultaneously – i.e. edges are matched using vertex mappings and faces are matched using edge mappings. Accordingly, the syntax-specific matching approach not only exploits intrinsic characteristics of model elements as compared to the previous three approaches, but also uses specific information on how they relate to other model elements in their respective data structure.

3.4.4 Pose registration

All of the difference calculation approaches described here require that the shapes being compared are positioned and oriented consistently in their respective coordinate systems beforehand; that is, they have to fit appropriately on top of each other. In any given pair-wise comparison scenario, this preliminary step, referred to as pose registration (Yang, Lin et Zhang, 2007), may either be executed automatically or manually since it only needs to be executed once before pair-wise model comparisons, as opposed to shape-based retrieval (1-to-N) problems. Many methods for the automatic pose registration of explicit geometric models are available (e.g. Pottmann, Leopoldseder et Hofer (2004), Tarbox, Gottschlich et Gerhardt (1993)); a popular one being the principle component analysis (PCA) transformation (Yang, Lin et Zhang, 2007).

Remarkably, a few difference calculation algorithms manage to implicitly perform the pose registration of the compared models. The algorithms proposed by Msaaf, Maranzana and Rivest (2007) via their *3DComparator*, and by Cuillière *et al.* (2009) and Souaissa *et al.* (2010), notably, use pose-independent geometric signatures to match B-Rep entities. Consequently, pose registration is performed as part of the algorithms by using the mappings determined by their respective matching sub-algorithms.

3.4.5 About comparing B-Rep models

Using B-Rep difference calculation methods will benefit to some comparison scenarios in which ensuring the topological equivalency of 3D CAD models' boundary representations becomes critical. For instance, information about a surface to be machined and its boundary edges is necessary for the calculation of NC tool paths. If the topological structure of a reference 3D CAD model was to be altered in some way, such as via 3D CAD data translation as shown in Figure 3.7, the ensuing NC toolpaths' calculation process may produce detrimental results, such as excessively longer machining times, or even fail.

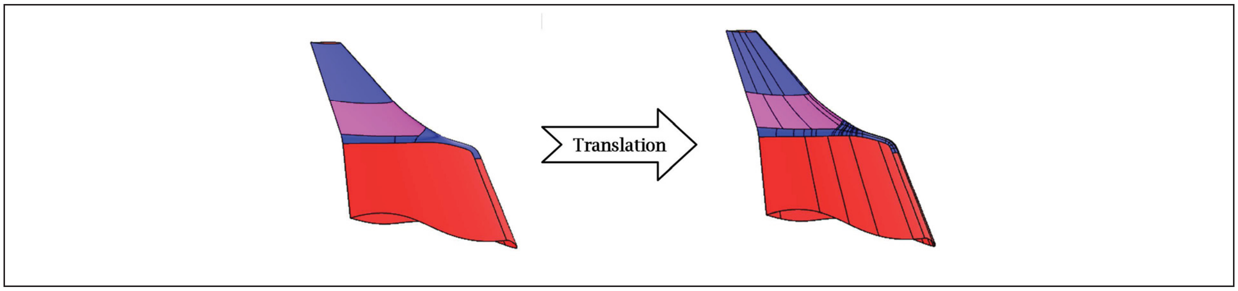


Figure 3.7 Examples of two geometrically equivalent yet topologically different 3D CAD models of a wingtip
Courtesy of CapVidia®

3.4.6 Relevance in a MDI problem

As described in Section 3.3, MDI problems comprise the pair-wise comparison of two 3D CAD models and require a significant level of detail regarding the calculation and representation of the differences. Appropriate difference calculation approaches are bound to provide enough information leading at least to the distinction of each single difference within the parts' geometric definitions. Methods using global geometric properties should therefore be dismissed.

Moreover, the trade-off between the accuracy of the calculated results and the computational efficiency of the difference calculation, specifically identified in approaches using auxiliary approximate geometric representations, may lose its bearing. MDI solutions can focus primarily, but not exclusively, on the accurate calculation and representation of the differences, which is their basic function, rather than on the computational efficiency of an algorithm executed singly.

3.5 Inventorying the current MDI-capable software tools

In order to go deeper into the description of the state of the art about 3D CAD model difference identification (MDI), this next section reports on the variety of existing software tools capable of locating and detailing geometric similarities and/or differences between two given 3D CAD models. Aside from their sought-after functionalities, tools were identified and inventoried based on their availability to the general public. Consequently, commercially available software tools make up the entirety of the compiled inventory, as no publically-available instance of other types of implementations, such as research prototypes, were found. Still, this particular review contributes to our purpose as no comprehensive inventory of MDI tools had been made to date. For some examples of shape-based retrieval software, the reader may refer to Msaaf, Maranzana and Rivest (2007).

The following inventory was divided into four categories based on the software tools' primary function: (1) 3D CAD systems, (2) 3D CAD validation tools, (3) 3D CAD visualization and collaboration tools and (4) other miscellaneous tools with MDI capabilities. This categorization allows us to distinguish some issues that are characteristic of the MDI problem.

3.5.1 3D CAD systems

MDI capabilities are usually available in 3D CAD systems amongst many others peripheral model analysis functionalities. Table 3.3 lists and describes seven (7) different CAD systems comprising such capabilities.

Tableau 3.3 3D CAD systems with MDI capabilities

3D CAD system	*Ref.*	*Compared representation*		*Additional details*
		Explicit	*Procedural*	
CATIA® V5	(Dassault Systèmes, 2007)	X		From *DMU Space Analysis* workbench.
CoCreate®	(CoCreate Software GmbH, 2008)	X		Proprietary method described in CoCreate Software GmbH and Gutierrez (3 juin 2010).
NX®	(Siemens PLM Software inc., 2009a)	X	X	
Pro/ ENGINEER®	(Parametric Technology Corporation, 2009)	X	X	
SolidWorks®	(Dassault Systèmes, 2010)	X	X	Available in the *SolidWorks Utilities* add-in; Developed in part by Geometric Ltd (2010).
SpaceClaim®	(SpaceClaim Corporation, 2011)	X		
TopSolid®	(Missler Software, 2008)	X		

According to their respective documentation, the 3D CAD systems' main purpose in providing pair-wise model comparison capabilities is to allow users to compare successive versions of a model and, ultimately, to manage change in 3D CAD models. Notably, 3D CAD systems are the only software tools that allow the comparison of the models' procedural representations, i.e. the sequence of modeling operations and their corresponding parameters (also referred to as feature trees, specification trees, construction or modeling histories, etc.), provided that the compared models are both proprietary models. The comparison of imported models is also possible, but is heavily dependent on the 3D CAD system's importation capabilities, which are likely to induce some level of data degradation during the translation process (e.g. loss of modeling features, topological alterations, etc.).

3.5.2 3D CAD validation tools

The second category comprises current, commercially available MDI-capable software tools that support the geometric validation process of 3D CAD data translation/remastering. Although initially designed for geometric validation applications, all of the inventoried validation tools can be used as MDI tools, as they provide an applicable level of detail about the located differences in those cases where the validation fails. All but one of the tools listed in Table 3.4 are integrated in 3D CAD feature-based translation software.

Tableau 3.4 3D CAD validation software tools with MDI capabilities

Editor	*Software tool*	*Ref.*	*Interoperability mode*	*Additional details*
CapVidia	CompareVidia®	(CapVidia NV, 2010)	Licensed libraries	
CT CoreTechnologie	3D_Evolution©	(CT CoreTechonologies Group, 2008)	Licensed libraries	
Elysium	CADdoctor®	(Elysium Co., 2010)	Licensed libraries	
ITI TranscenData	CADIQ®	(ITI TranscenData, 2010a)	CAD systems' API	
Kubotek	Kubotek Validation Tool™	(Kubotek USA, 2010)	Licensed libraries	
Theorem Solutions	Theorem Process Manager© (TPM)	(Theorem Solutions, 2010)	Licensed libraries	
Translation Technologies	Mirror Model Comparator© (MMC)	(Translation Technologies Inc., 2010)	CAD systems' API	Proprietary method described in Translation Technologies Inc. (12 décembre 2006).

As validation tools, the key result these software tools are primarily designed to provide is a straightforward pass/fail diagnosis regarding the compared models' explicit geometric equivalency. The compared models' equivalency is usually based on selectable and/or adjustable criteria. When these criteria are not completely fulfilled, the validation tools may indicate where these discrepancies are located in relation to the pre-set criteria.

The 3D CAD data translation/remastering process these tools are designed to validate inevitably requires them to compare 3D CAD models represented in heterogeneous formats. Two opposite approaches to this interoperability issue can be identified. First, a validation tool may use licensed libraries from the major 3D geometric kernel editors to read native files and import the data into its own 3D geometric kernel. Such an approach benefits the comparison process, as the validation tool does not require CAD systems to be installed on the computer. However, it does involve the translation of the native data, which may induce some undesirable degradation unmanaged by the comparison tool.

Alternatively, a validation tool may use a 3D CAD system's application programming interfaces (API) to analyze the native data as it was originally represented. While the risk of data degradation due to the translation process is minimized, this second approach requires the two compared models' source 3D CAD systems to be installed and their respective API's to be made available in order for the comparison to take place.

3.5.3 3D CAD visualization and collaboration tools

As a key part of PLM, 3D CAD visualization and collaboration tools are designed to enable the viewing, manipulation, annotation and sharing of engineering 3D models throughout the different functions of an extended enterprise, without resorting to the native 3D CAD systems. Some of those tools, as listed in Table 3.5, offer MDI capabilities as an analysis function that can support, among others, the dissemination of engineering change applied to part models.

Tableau 3.5 3D CAD visualization and collaboration software tools with MDI capabilities

Editor	*Software tool*	*Ref.*	*Additional details*
Actify	SpinFire™	(Actify inc., 2010)	
Adobe Systems	Acrobat® Pro	(Adobe Systems inc., 2008)	Available in *3D Reviewer*©.
C4W	3D Shop ModelScan©	(C4W, 2008)	
CADCAM-E.com	CCE EnSuite©	(CADCAM-E.Com, 2010)	
Lattice Technology	XVL Studio© Professional	(Lattice Technology Co., 2010)	
Oracle	AutoVue© Electro-Mechanical Professional	(Oracle Corporation, 2010)	
Synergis Software	Adept©	(Synergis Software, 2010)	MDI capabilities provided by Oracle Corporation (2010).

On the matter of 3D CAD interoperability, visualization tools purposely resort to licensed libraries to read native files in order to circumvent the use of source 3D CAD systems. Accordingly, degradation of native data is predictable as collaboration tools commonly exploit lightweight 3D geometric formats to facilitate 3D CAD data sharing, such as on the web (Vezzetti, 2009). For example, none of the tools listed in Table 3.5 preserves the procedural representation, i.e. features and parameters, of geometric data imported from native files. Furthermore, lightweight formats commonly approximate exact geometry with planar facet tessellations, raising issues regarding the accuracy of the difference calculation process between explicit geometries.

3.5.4 Other software

When it comes to inventorying existing software tools, the sought-after 3D CAD MDI capabilities may not be explicitly detailed as one of the software's key features or labeled functions, but rather be deducible with some ingenuity. A broader view of the state of the art is thus possible. Nevertheless, it must be kept in mind that these implicit capabilities might inevitably be restricted in their use since they may not have been initially designed for the specific purpose of MDI.

Point-cloud-based inspection software tools, such as InnovMetric Software's PolyWorks® (2009), constitute a first type of software that can be used to compare 3D CAD models. An automated inspection system such as the one described by Prieto *et al.* (1999) will take in an unordered cloud of the 3D points of an actual part obtained from a high-resolution 3D range sensor, along with its 3D CAD model. The cloud is then segmented by computing the minimal distance and comparing some local geometric properties between the 3D points and the CAD model's surface.

Then again, two 3D CAD models can be compared by substituting the point cloud obtained from the 3D range sensor with a point cloud obtained from the surface of another similar 3D CAD model by means of macro programming. Such practice basically reproduces a variant of the point-to-part method for calculating geometric differences, which is detailed in section 3.4.

Autodesk® (2010b) released a technology preview of Autodesk® Inventor® Fusion, a history-free CAD modeling software. When used concurrently with the parametric feature-based CAD software Autodesk® Inventor® (Autodesk inc., 2010a), this technology preview includes a complementary module called the Inventor® Fusion Change Manager. The module is designed to propagate direct-modeling modifications applied to a 3D CAD model in a history-free environment to a feature-based parameterized version of the same model in a history-based environment. The propagation process begins with the comparison of the new history-free version with the earlier feature-based version of the CAD model in order to locate the modifications and compute their representation as new parameterized feature instances. At this prototype stage, the comparison capabilities of the Change Manager module are strictly limited to successive versions of Inventor® 3D CAD models, the newer version having been edited in Inventor® Fusion.

3.6 Perspectives

By describing and analyzing the uses, the tools and the methods related to the comparison of 3D CAD models, this paper allows us to outline the research perspectives into this promising CAD- and PLM-related domain; more specifically, in relation to the MDI-related problem of shape change transposition. The process of transposing shape change between 3D CAD models may be divided into two fundamental steps (see Figure 3.1): (1) the location and measurement of geometric differences between a modified reference model and an initial target model, each formatted differently yet related (e.g. the initial target model embodies an earlier version of the part modeled by the modified reference model) and (2) the modification of the initial target model based on an adapted representation of these geometric differences. We describe three research perspectives we intend to pursue in future work.

3.6.1 Difference representation

Our survey of 3D CAD model difference calculation approaches and methods was key to this paper, as it outlines one of the central ingredients for any MDI solution, the other component being difference representation. Whereas a number of difference calculation methods and algorithms could be reported on and categorized, developments regarding difference representation have remained of secondary importance. The representation of such differences is highly dependent on their calculation. However, the requirements regarding what information must be provided by the comparison influence the selection of the proper calculation method.

In the context of shape change transposition, the intended update of the target model determines the outcome of the geometric difference calculation step which is to be represented in some form that is amenable to further manipulations, whether they are to be performed manually or automatically. New emphasis should therefore be put on the adequate representation of 3D CAD model geometric differences, such as outlining the minimal set of requirements that should be taken into account in order to define a suitable representation technique for the problem at hand.

3.6.2 Search for representation completeness

Due to their higher level of semantics, the comparison of the 3D CAD models' feature-based procedural representations is expected to provide more relevant information on geometric differences from the point of view of design engineering than the other difference calculation and representation approaches. However, this particular approach inherently lacks flexibility, being strictly limited to the comparison of 3D CAD models directly derived from one to another. Also, since the compared shapes are represented implicitly, adequate calculation precision and recall for explicit geometric differences cannot be guaranteed.

A promising avenue for an MDI solution to efficiently support the scenario of shape change transposition would be to develop a difference calculation and representation approach that combines the calculation precision and recall of explicit geometry comparison with the representation completeness of procedural representation comparison. One key challenge of such an avenue would reside in overcoming the issue of comparing heterogeneously-represented 3D CAD models.

3.6.3 Parametric representation of geometric differences

Intuitively, dimensions and other geometric constraints, referred to here as parametric data, fundamentally constitute the first level of engineering semantics that distinguish a 3D CAD model embodying the geometric definition of a mechanical part from any given 3D shape model. Systematically represented in both the feature-based parametric and the direct (explicit) solid modeling paradigms implemented by most modern 3D CAD systems, parametric data can also be seen as the primary object of engineering change applied to the geometric definition of a part.

Unfortunately, despite the variety of MDI tools and difference calculation approaches surveyed in this paper, very few developments have been aimed at representing 3D CAD geometric differences via their parametric representation. The comparison of procedural representations involves locating parametric differences between matched modeling operations, even though such parametric data remains implicit with respect to the geometry. Whether it is through difference calculation, difference representation or some combination of the two, the objective is to work towards an MDI solution for shape change transposition that takes advantage of the compared 3D CAD models' parametric representation of engineering data.

3.7 Conclusion

This paper reviewed recent developments in the domain of 3D CAD model comparison. The subject was outlined, in a comprehensive approach, from three related points of view, providing the groundwork for our projects addressing the problem of shape change transposition. First, we described and analyzed the possible uses for 3D CAD comparison and identified 3D CAD model difference identification (MDI) as the most promising solution domain. Existing MDI-capable software tools were inventoried and categorized. A survey of various difference calculation methods was also presented. Finally, in light of the review, research perspectives towards innovative MDI methods in shape change transposition scenarios were then contemplated.

Our research clearly shows that pair-comparison of 3D CAD models does not draw as much attention in the CAD and PLM research communities as shape-based retrieval. By concurrently reporting on the scenarios, the tools and methods for 3D CAD model comparison, the goal is to highlight the potential of these tools to address the contemporary challenges of PLM and, thus, bolster new developments in the comparison of 3D CAD models. Moreover, as software developments have been made in recent years, comparing available solutions constitutes a complex and delicate task that remains to be performed.

CHAPITRE 4

3D CAD MODEL COMPARISON: AN EVALUTION OF MODEL DIFFERENCE IDENTIFICATION TECHNOLOGIES

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article publié dans la revue « *Computer-Aided Design and Applications* » en mars 2013.

4.1 Abstract

The use of 3D CAD in mechanical product design has become a standard practice. Consequently, methods and tools are continually being developed to improve designers' efficiency in the creation, modification and analysis of 3D CAD models. Recent software developments had led to the emergence of multiple tools capable of comparing 3D CAD models to locate shape similarities or differences, leading to benefits in various CAD- and PLM-related application domains such as design reuse, data exchange and engineering change management. This paper describes evaluation trials that were performed on sets of commercially available 3D CAD model comparison tools. The goal was to evaluate their capacity to efficiently calculate, represent and display 3D CAD model differences in shape change transposition scenarios where shape modifications must be precisely located and elaborated in order to be consistently propagated between application-specific models of a product. First, some basic concepts of 3D CAD model comparison are presented. Then, simulated shape change assessment scenarios are defined to pilot two series of evaluation trials intended for existing software tools capable of comparing 3D CAD procedural and explicit models, respectively. The results are summarized and conclusions are drawn.

4.2 Introduction

The development of complex products depends on engineering processes through which a product's definition evolves systematically, driven by the concurrent work of many specialists from distributed teams producing and modifying the product model, making decisions and taking actions accordingly. In numerous cases, those decisions and actions highly depend on the proper identification, representation and integration of what differentiates the new version of the product model from its previous versions. Accordingly, innovative methods and tools are sought after to enable the fast, accurate and comprehensive identification and representation of the differences between models.

Three-dimensional (3D) CAD model comparison has been the focus of several advancements in the last decade, notably as part of product lifecycle management (PLM) initiatives towards better product information reuse (Cardone, Gupta et Karnik, 2003; Jackson et Buxton, 2007) and product data exchange (CAx Implementor Forum, 2008). Recent commercial software developments focused on the pair-wise comparison of 3D CAD models, mostly enabling the detection of model differences for validation purposes (Brière-Côté, Rivest et Maranzana, 2012a). The relevance of such comparison tools in engineering change management scenarios remains however to be demonstrated, as few were expressly designed for the detailed assessment and representation of shape changes between versions of a mechanical part.

This paper falls within the framework of a research project addressing the subject of product shape change transposition between heterogeneously formatted 3D CAD models. The objective is to develop a novel 3D CAD model comparison mechanism providing application-specific representations of shape differences between models of an evolving mechanical part, thus optimizing their identification, their interpretation and their integration by the individual engineering processes responsible for updating target application-specific models.

Figure 4.1 illustrates a sample case of shape change transposition. An initial reference model, released by Design Engineering in the form of a STEP file, defines a part's original geometry. An initial target CAD/CAM model is created by Manufacturing Engineering as per the initial reference model in a format deemed appropriate for manufacturing planning (e.g. procedural modeling). Eventually, an engineering change order (ECO) calls for the release of a modified reference model, derived from the initial reference model. To derive a modified target CAD/CAM model from the initial target model, Manufacturing Engineering must therefore identify the exact shape change through model comparison and transpose the description of this shape change in the manufacturing domain.

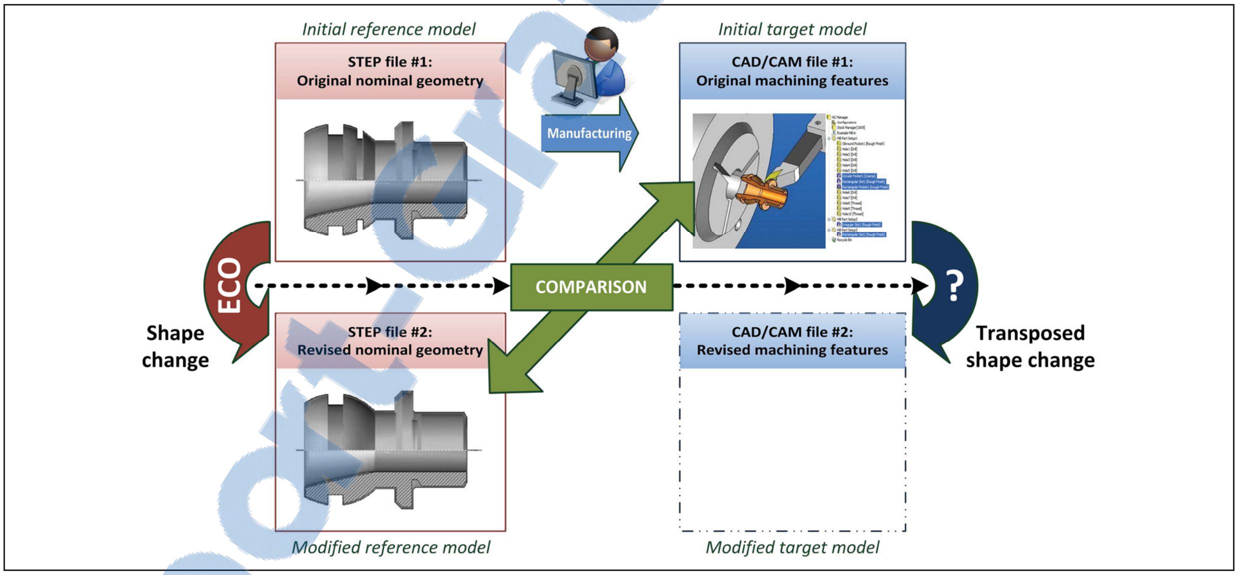


Figure 4.1 Sample scenario of shape change transposition: Modifying a CAD/CAM model based on revised geometry

A preceding paper by the same authors (Brière-Côté, Rivest et Maranzana, 2012a) identified and organized existing literature and recent developments relating to 3D CAD model comparison, including an inventory of commercially available comparison software tools. While such review focused mainly on the applicative and theoretical aspects of 3D CAD model comparison, this paper explores its practical aspect by describing two series of evaluation trials that were performed on a representative sampling of commercially available comparison software tools. The objective of the evaluation trials was to assess if and how much these specific tools, implementing the various difference calculation methods reported previously, could effectively contribute to the transposition of a shape change between application-specific models that must be kept consistent. Results of this investigation would then set the groundwork for the development of a novel 3D CAD model comparison mechanism.

The structure of this paper is as follows. Section 4.3 summarizes some of the theoretical aspects of 3D CAD model comparison that were introduced in Brière-Côté, Rivest and Maranzana (2012a) and are relevant to the current investigation, such as the basic functions and solution domains, the composition of the model comparison problem and an overview of current difference calculation methods. In Section 4.4, we give a brief description of our approach in devising and performing the evaluation trials, including the definition of the set of criteria that were used to evaluate the software tools' relevance in two simulated shape change assessment scenarios. Sections 4.5 and 4.6 detail the first and second series of evaluation trials on software tools performing 3D CAD model comparison via the models' procedural and explicit geometric representations, respectively, along with results and other observations. Conclusions and future contributions are described in Section 4.7.

4.3 An overview of 3D CAD model comparison

Prior to the evaluation trials, this first section briefly describes the topic of 3D CAD model comparison as it is exposed in current CAD- and PLM-related documentation and research literature. We refer to 3D CAD model comparison as the general process of comparing two or more CAD models in order to yield a statement or an estimate of their geometric similarities and/or differences and, consequently, to support decision making in any given product lifecycle phase. For a more detailed review on this specific topic, the interested reader can refer to a previous work by the same authors (Brière-Côté, Rivest et Maranzana, 2012a).

4.3.1 Application and solution domains

Six (6) 3D CAD model comparison application domains were identified to categorize the numerous and varied use cases for comparison during the different phases of the product lifecycle:

- *Product information reuse* – achieving one of PLM's key aspects towards reducing costs and delays, by using a product's shape to retrieve and assess reusable product data such as manufacturing processes (Huang, Tian et Zhou, 2004), sourcing and pricing information (Msaaf, Maranzana et Rivest, 2007), qualification tests results (Cessna Aircraft Company, 2010), etc.;
- *Product rationalization and standardization* – eliminating duplicates and grouping similar existing parts and products into new families (Chowdhury et Siddique, 2009) or optimized manufacturing batches for more efficient outsourcing (Msaaf, Maranzana et Rivest, 2007);
- *CAD modeling management* – preventing model duplication and promoting modeling best-practices on the basis of geometric comparison between existing and new 3D CAD models (Parametric Technology Corporation, 23 septembre 2003).
- *CAD data translation and remastering* – monitoring the possible lost or degradation of 3D CAD shape data rendered automatically (translation) or manually (remastering) in formats different than the one it originates from (e.g. CAx Implementor Forum (2008));
- *CAx models authoring* – verifying the geometric consistency of intermediate or analysis models with respect to the master model they are derived from (e.g. Lockett and Guenov (2008)); and
- *Engineering change management* – identifying and assessing the impact of ordered shape changes on a part's definition and downstream models, such as process plans, NC programs, analysis and simulation models (e.g. Souaissa *et al.* (2010), Sypkens Smit and Bronsvoort (2007)). Shape change transposition cases, like the one pictured in Figure 4.1, belong to this particular application domain.

Current 3D CAD model comparison solutions cannot contribute interchangeably to any application domains. Depending on the use case, the basic function of the comparison process varies with regard to two key factors which are the comparison's cardinality – comparing two (1:1) or many models (1:N) – and the level of details expected from the comparison – ranging from a simple "Yes-No" diagnosis to elaborated descriptions of the differences. Accordingly, as pictured in Figure 4.2, three distinct solution domains were identified to categorize 3D CAD model comparison solutions implementing similar basic functions.

Shape-based retrieval collates model comparison solutions implementing basic functions such as "finding duplicate" or "finding similar" models. These solutions usually relate to use cases from the product information reuse and product rationalization and standardization application domains where large sets of models are compared (1:N cardinality) and where simple results are expected (finite sets of equivalent models or scale-based distributions of similar models). Examples of shape-based retrieval solutions were largely reviewed in many previous works (e.g. Cardone, Gupta and Karnik (2003), Iyer *et al.* (2005)). Higher cardinalities require higher computational efficiency, leading a vast majority of shape-based retrieval solutions to use lightweight pre-computed shape signatures to aggregate shape and other model features for fast, yet coarse comparison.

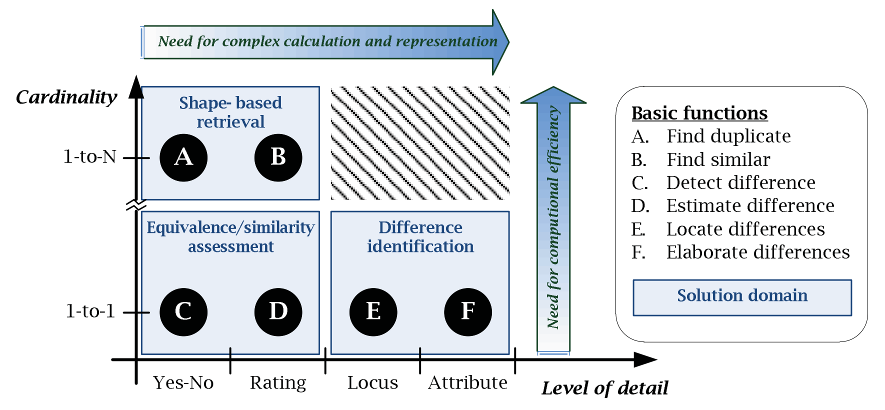


Figure 4.2 Solution domains and basic functions as a relation between the required level of detail and cardinality

The equivalence/similarity assessment solution domain comprises model comparison solutions implementing the “detect difference” and “estimate difference” basic functions. These solutions are used to check if two models (1:1 cardinality) are equivalent according to some explicit criteria or to estimate their relative similarity, i.e. providing a qualitative appraisal of how close or different they are from each other (few to no details about the similarities/differences). Application domains such as CAD data translation/remastering (e.g. CAx Implementor Forum (2008)) and CAx models authoring (e.g. Lockett et Guenov (2008)) benefit mainly from equivalence/similarity assessment solutions.

Then, model difference identification (MDI) constitutes the third solution domain, organizing model comparison solutions implementing basic functions such as “locating differences” and “elaborating differences” between pairs (1:1) of 3D CAD models. As high levels of details are required in use cases such as those from the product information reuse, the product rationalization and standardization and the engineering change management application domains, MDI solutions involve complex mechanisms for the calculation and the representation of model differences (e.g. CoCreate Software GmbH et Gutierrez (3 juin 2010)). Accordingly, their use is better suited for low-cardinality or pair-wise model comparison problems.

This paper focuses on pair-wise model comparison solutions, most specifically on MDI solutions, as they can be applied to engineering change management use cases, such as the transposition of shape changes between two models. Consequently, shape-based retrieval and equivalence/similarity assessment solutions are considered out of scope in the present work.

4.3.2 Components of model difference identification (MDI)

The process of identifying similarities and differences between two 3D CAD models is intrinsically complex. The model difference identification process can be separated into three major components (Kolovos *et al.*, 2009):

- *Calculation* – a procedure, method or algorithm able to compare two distinct 3D CAD models, i.e. identifying mappings between model elements and, then, similarities and differences;
- *Representation* - the outcome of the calculation must be described and represented in some form that is amenable to further analysis or manipulations; and
- *Visualization* - model differences often need to be presented according to a specific need or scope, highlighting those pieces of information that are relevant only for the prescribed goal.

Calculation and representation are the central ingredients for any MDI comparison solution, but all three components tend to overlap. Difference representation is highly dependent on the calculation method and, therefore, on the representation of CAD data used for comparison, as they define the type of data to be represented and manipulated. Therefore, the effectiveness of difference representation is often compromised by factors such as the calculation method or the scope of the difference. As for visualization, it is realized by specifying a concrete syntax which renders the abstract representation of differences. Accordingly, in some solutions, both the representation and the visualization of model difference are rendered with the same notation (e.g. model comparison by means of graphical representations).

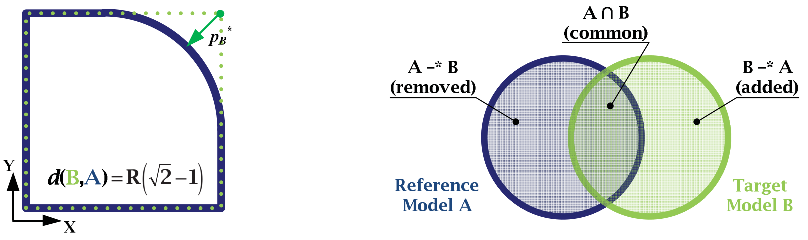
4.3.3 Difference calculation methods

Two approaches for calculating 3D CAD model differences between pairs of models have been identified among current model difference calculation methods: (1) explicit geometric comparison and (2) model data structure matching and comparison. Explicit geometric comparison operates on the explicit representation of geometric objects, like solids, surfaces or point sets, for which geometric properties can be evaluated, e.g. volume, area, distances, positions, etc. Conversely, model data structure matching and comparison operates on CAD data structures, model data elements, like B-Rep entities or modeling operations, and their attributes.

Generally, whatever the difference calculation method, compared shapes must be positioned and oriented consistently in their respective coordinate systems beforehand; that is, they have to fit appropriately on top of each other in 3D space. Such preliminary operation is known as *pose registration* (Yang, Lin et Zhang, 2007). Some particular algorithms are able to perform pose registration implicitly by manipulating pose-independent geometric properties (e.g. Msaaf, Maranzana et Rivest (2007), Souaissa *et al.* (2010)).

Examples of explicit geometric comparison methods are the comparison of global geometric properties, the point-to-part deviation calculation and the spatial occupancy comparison. The comparison of global geometric properties such as the 3D CAD models' volumes, total surface areas and moments of inertia, among others, enables the quick assessment of two models' geometric equivalency. Given its low computation cost and the availability of the compared metrics, such method has been used in CAD modeling management systems (Parametric Technology Corporation, 23 septembre 2003) and geometric validation mechanisms (CAx Implementor Forum, 2008; Frechette, 1996).

Also called the “Cloud Of PointS” (COPS) mechanism (CAx Implementor Forum, 2008), the point-to-part difference calculation method is based on the evaluation of the Hausdorff metric between the models' surfaces. As pictured in Figure 4.3(a), it measures the distance between two subsets of the 3D space, with one of the subsets being discretized into a point set. The location and evaluation of multiple local deviation maxima can be computed by dividing the models' surfaces into smaller subdomains like faces (e.g. CapVidia NV (2010)) or local neighboring regions (e.g. Lattice Technology Co. (2010)).



(a) (b)

(a) Point-to-part deviation calculated between sampling points from part B and the boundary of part A;
(b) Identifying regions of material removal and addition between parts in the spatial occupancy comparison method;

Figure 4.3 Illustrating explicit geometric comparison methods

Figure 4.3(b) illustrates the basic principle on which relies the spatial occupancy comparison. Regions of material removal and material addition are distinguished from the common space occupied by both part models. The superposition of 3D shapes is the more straightforward way to compare spatial occupancy. Algorithms similar to those used to verify part mating, clearance and interference in DMU environments (e.g. Dassault Systèmes (2007)) are usually applied here. The calculation of the regularized Boolean operations from set theory between two solids can also be performed by a geometric modeling kernel (e.g. Dassault Systèmes (2010)). Likewise, the use of decomposition representations, such as voxels (e.g. Dassault Systèmes (2007)) and octree structures (e.g. François et Cuillière (2000)), to compare spatial occupancy can increase the method's computational efficiency.

In the second approach for calculating model differences, i.e. comparing the models' data structures, the most intricate part of the calculation task is the matching of equivalent model elements between data structures. Four (4) matching methods for model elements have been identified (Kolovos *et al.*, 2009):

- *Static identity-based matching* – matching elements via their persistent and unique identifier assigned upon creation and maintained through modification;
- *Signature-based matching* – using a specific subset of an element's attributes or properties to find an equivalent counterpart;
- *Similarity-based matching* – associating model elements based on the measured similarity of their aggregated geometric and/or descriptive features;

- *Syntax-specific matching* – incorporating the semantics of the compared 3D CAD models' data representation scheme in the matching of model elements; e.g. specific relationships between model elements.

Fundamentally, CAD shape data is either represented following the procedural or the explicit modeling approach. A procedural model is described in terms of the operation of a sequence of procedures (which may include the solution of constraint sets), as opposed to an explicit or evaluated model whose full details are immediately available without the need for any form of calculation (International Organisation for Standardization, 2005). Procedural models may be evaluated in order to be compared to explicit models. The opposite is however unfeasible, as procedural models are not unique for a given explicit geometry. Common 3D CAD parametric feature-based modeling systems are known to produce hybrid models, as model histories or construction trees are procedural shape representations while 2D sketches used as input for modeling operations, among others, are explicit shape representations (Kim *et al.*, 2008).

4.4 Evaluation methodology

This section presents an outline of the methodology used to perform the two series of evaluation trials on a set of existing 3D CAD model difference identification software tools. The details and results of each series are presented in the next two sections. In Section 4.5, the first series involves tools implementing the comparison of 3D CAD models by means of their respective procedural representations, i.e. via implicitly represented geometry. Then, the second series focuses on tools implementing the comparison of 3D CAD models via the explicit representations of the modeled shapes and is detailed in Section 4.6.

4.4.1 Engineering change management scenario

Simulated 3D CAD model comparison scenarios are defined for each series of evaluation trials. As stated earlier, the objective of the evaluation trials is to assess if and how much existing MDI-capable software tools could efficiently contribute to the transposition of a shape change between two application-specific models. Consequently, both scenarios involve the elaborated description of a shape change distinguishing two successive versions of a mechanical part's 3D CAD model as it would occur, for example, during the impact analysis of an engineering change proposal.

The 3D CAD models' reference original versions were inspired from geometric model available in the Engineering Shape Benchmark (Purdue Center for Information Systems in Engineering, 2007). Modified versions were then created according to a predetermined list of discrete modifications reproducing specific model difference calculation situations. How each tool processed these situations contributed to their evaluation. The lists and descriptions of the modifications are given for each series in the following sections. A repository of the models used for both series of evaluation trials is available online[2] for download.

4.4.2 Evaluated comparison tools

The selection of the 3D CAD model comparison software tools to be evaluated was based on the inventory and categorization of existing MDI-capable software tools presented in Brière-Côté, Rivest et Maranzana (2012a). Tools were selected based on their availability, free of charge, for direct examination and manipulation, i.e. all tests were to be executed locally by the authors; not as part of demonstration sessions or executed by representatives from software editors. Accordingly, evaluated comparison tools were either fully licensed versions, or provisional trial license graciously supplied by software editors.

Selected tools were initially divided into four categories based on the software tools' primary function (Brière-Côté, Rivest et Maranzana, 2012a):

- Three-dimensional (3D) CAD systems, enabling the creation, modification, analysis and optimization of 3D part and product models;

---

[2] http://profs.etsmtl.ca/lrivest/Recherche/index.html

- Three-dimensional (3D) CAD visualization and collaboration tools, applied in the efficient viewing, manipulation, annotation and sharing of native or lightweight 3D CAD models;
- Three-dimensional (3D) CAD validation tools, originally designed to support the geometric validation of translated or remastered 3D CAD data; and
- Miscellaneous tools with some MDI capabilities.

4.4.3 Preprocessing and configurable settings

When made available, support documentation for the evaluated comparison tools was examined to retrieve appropriate procedures to follow for the comparison of the 3D CAD models. The need for any preprocessing steps, such as pose registration, and the sets of configurable settings were also surveyed.

4.4.4 Evaluation criteria and rating scale

Given the simulated shape change assessment scenarios, model differences identified between the 3D CAD models' versions need not only to be calculated, but also suitably represented and visualized. To that intent, inspired by shape-based retrieval literature (Iyer *et al.*, 2005), we defined five criteria on which to evaluate the results provided by the different MDI-capable software tools and their respective pair-wise comparison functions:

EC#1. *calculation recall* – the ability of the difference calculation algorithm to locate all relevant differences without omissions (false negatives);

EC#2. *calculation precision* – the quality of a difference calculation algorithm able to precisely locate relevant differences while avoiding false positives;

EC#3. *representation range* – a quality that characterizes the level of detail and the amount of information provided in the description of the differences; and

EC#4. *representation accuracy* – the quality of a function that accurately identifies and describes the nature of the located differences as per the corresponding representation range;

EC#5. *visualization discernability* – the quality of a difference visualization mechanism that makes the identification and the perception of differences a relatively simple process.

Based on the compiled results and observations, software tools were rated according to the following 3-grade rating scale to specify to what extent each of them met the considered evaluation criteria:

- Good (● ● ●), highlighting the tool's reliability with respect to the examined criterion in the identification and analysis of model differences;
- Fair (● ●), indicating an acceptable performance level for the evaluated tool, despite some minor or generalized shortcomings/omissions; or
- Poor (●), denoting the observation of major flaws or shortcomings, rendering the tools unfit for the assessment of engineering change.

4.5 Comparing procedural representations of 3D CAD models

Software tools providing MDI capabilities by means of procedural representation comparison were the subject of a first and distinct series of evaluation trials for two main reasons:

- The implemented difference calculation approach distinctively compares 3D CAD models represented implicitly via the tree-like structure of their geometric modeling operations instead of via explicit 3D geometry; and
- The applicability of this approach is strictly limited to MDI scenarios involving related 3D CAD models embodying versions of a given part.

4.5.1 Scenario

The shape change assessment scenario used as the setting for this first series involves two 3D CAD models embodying via procedural representation the successive versions of a mechanical part. Figure 4.4 presents the geometry of both the original and modified versions of the part. The original procedural model comprises 55 modeling operations, more than half of which are sketch-based extrusion features (35). Simple modeling operations, which also include holes, mirror patterns and datum planes, were used to build the models' modeling histories so that they could be easily retrieved and instantiated whatever the 3D CAD format the models were to be expressed in. The resulting explicit geometries purposely excluded complex geometric entities (e.g. NURBS curves and surface) to ensure the geometric reproducibility of the models when processed by different geometric modeling kernels.

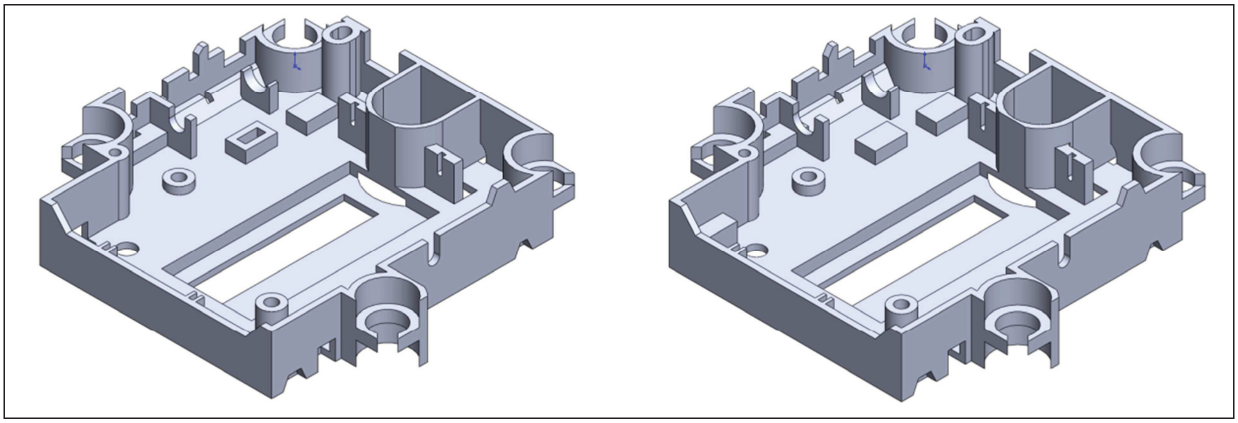


Figure 4.4 Resulting part geometry of the original (left) and modified (right) 3D CAD models used for the evaluation of tools implementing the comparison of procedural representations

Since the comparison of procedural representations is only possible between homogeneously formatted models, each trial had to use a distinct pair of reference and target models. Target procedural models representing the part's modified geometry were generated by copying their respective reference models and by editing the copied models' modeling histories. The modifications differentiating the models to be compared are as follow:

- Thirteen (13) modeling operations were directly modified at the parametric level, i.e. one or more of their parameters had their value changed. One particular parametric change did not lead to any change in the resulting explicit geometry.
- Two (2) modeling operations were removed and two (2) modeling operations were added from the modeling sequence. Two of the removed/added operations had no geometric outcome. Also, two (2) modeling operations were reordered in the modeling sequence; only one led to a change in the target geometry.
- Two (2) modeling operations underwent metadata modifications, i.e. one had its identifying name modified, while the other had its related sketch's name modified.

In this first scenario, the outcome of the 3D CAD models' comparison was mainly to locate the differences and provide as much details as possible on their representation via modeling operations and related parameters. The comparison results were expected to enable an appropriate assessment of the engineering change represented by the two versions.

4.5.2 Evaluated comparison tools

Among all the inventoried software tools from Brière-Côté, Rivest et Maranzana (2012a), only three (3) implement the comparison of 3D CAD procedural representations. All three are 3D CAD systems and are listed in Table 4.1. For a systematic evaluation, a reference model and a target model were built with each 3D CAD system using the predetermined sequence of modeling operations. Minor differences between the reference models' modeling histories had to be tolerated due to the inherent heterogeneity of the modeling operations provided by the different 3D CAD systems.

Tableau 4.1 3D CAD systems evaluated for their procedural model comparison capabilities

Trial	*3D CAD system*	*Release*	*Reference*
1	Dassault Systèmes SolidWorks®	2010 SP4.0	(Dassault Systèmes, 2010)
2	Siemens PLM NX®	7.0 MP1	(Siemens PLM Software inc., 2009a)
3	PTC Pro/ENGINEER®	Wildfire 5.0	(Parametric Technology Corporation, 2009)

4.5.3 Preprocessing and configurable settings

As observed in their respective support documentation, all three evaluated software tools account for the fact that the applicability of 3D CAD MDI via implicitly represented geometry is strictly limited to MDI scenarios involving related 3D CAD models embodying versions of a given part, i.e. models derived from one to another. Hence, the need for pose registration prior to the comparison was overlooked for these first trials since each pair of models was originally defined in the same coordinate system. Also, no configurable settings were to be adjusted prior to the comparison.

4.5.4 Results

Detailed results about the evaluation trials were compiled and are summarized in Table 4.2. Modeling operation counts for each of the six models' operation sequences took only first level parent operations into account, i.e. sketches or other child operations were not considered. In the cases where differences were specifically located in sketches or other child operations by the software tools, corresponding parent operations were implicitly considered as being different in the compiled results.

Tableau 4.2 Results from evaluation trials of history-based model comparison software tools.

	Scenario	*Trial #1*	*Trial #2*	*Trial #3*
Software tool	--	SolidWorks®	NX®	Pro/ENGINEER®
*Modeling operation counts**				
- Reference model	55	55	55	56
- Target model	55	55	55	56
Differences between histories				
- Modified operations	13 (24%)	22 (40%)	36 (65%)	22 (39%)
- Removed operations	2 (4%)	4 (7%)	2 (4%)	2 (4%)
- Added operations	2 (4%)	4 (7%)	2 (4%)	2 (4%)
- Reordered operations	2 (4%)	--	--	2 (4%)
- Renamed objects	2 (4%)	--	--	2 (4%)
- Operations not compared	--	--	10 (18%)**	--
Changes w/o geometric outcome				
- Parametric modification	--	Located	Inconclusive	Located
- Removed/added operations	--	Located	Located	Located
- Reordered operation	--	Not located	Not located	Located
Representation & visualization				
- Changed parameter values	--	Partial	No	Yes
- Reporting	--	Yes	Yes	Yes
- 3D interaction	--	Yes	No	Yes
Evaluation criteria				
#1. Calculation recall	--	● ●	● ●	● ● ●
#2. Calculation precision	--	● ●	●	● ●
#3. Representation range	--	● ●	●	● ● ●
#4. Representation accuracy	--	● ●	●	● ● ●
#5. Visualization discernability	--	● ●	●	● ●

* Subordinate operations such as sketches or patterned features are omitted.

** Three (3) hole operations were simultaneously identified has “changed” and “not compared”.

For all three systems, more modeling operations (22) were identified as being parametrically different than was originally intended in the scenario (13) – one of the systems locating nearly three times more modified modeling operations (36). This issue is mainly due to each system having its own set of rules regarding how to identify local parametric modifications and their ensuing impact on the overall target operation sequence and/or explicit geometry. For instance, in this particular scenario, later modeling operations impacted by modifications on preceding operations, such as mirror patterns referencing a modified seed operation, were unevenly identified as differences by the systems. The same behavior was observed for operations with sketches referencing modified explicit geometry. Thus, the counts for modified operations diverged, indicating variance in calculation precision (EC#2).

Removed/added modeling operations were all correctly located and accurately represented between the compared modeling histories. However, this was not the case for reordered operations and renamed objects. Only Pro/ENGINEER® explicitly identified two differences in the modeling sequences and two differences in the modeling trees' metadata. A major issue was also observed in trial #2 (NX®) regarding overall accuracy (EC#2 and EC#4), as some modeling operations were concurrently identified as both 'modified operations' and 'operations not compared'.

Except for reordered operations, modifications having no geometric outcome when applied to modeling histories were also generally detected as expected, denoting that the implemented difference calculation algorithms actually compares the 3D CAD models' modeling sequences and not their explicit geometric data. On the matter of representation accuracy (EC#4), results corresponding to parametric modifications not bearing a geometric outcome were inconclusive in trial #2 because no distinction could be made in the representation of the differences between those with a geometric outcome on the modeled part and those without.

As for the representation and visualization of comparison data, all three systems provided reporting functionalities to summarize and distribute the results. Interaction with both comparison and 3D geometric data was also possible in two out of three systems, enabling users to expand their analysis of the modified part. Nonetheless, 3D annotations locating the outcome of parametric modifications directly on the evaluated shapes would have improved discernability (EC#5). Relating to representation range (EC#3), the semantic value of 3D CAD procedural representations allows for detailed and significant comparison results as model differences were expressed in terms of, for example, an added blend radius, an increased wall thickness or displaced holes with respect to local references, etc. However, changed values for modeling operation parameters were not systematically specified as part of the comparison results. While Pro/ENGINEER® provided original and modified values for parameters from both modeling operations and sketches (e.g. dimensions), SolidWorks® provided similar data for only first-level modified modeling operations, excluding sketches.

4.6 Comparing explicit geometry

Explicit geometry comparison algorithms, whether they compare geometric data structures such as B-Rep data structures or other explicit geometric representations, are the type most widely implemented by 3D CAD MDI-capable software tools. As opposed to procedural representation comparison, corresponding difference calculation methods are various, having a direct influence on the representation and visualization of comparison results. Hence, the second series of evaluation trials focused primarily on the quality of the different comparison functions implemented by the inventoried software tools to locate and elaborate on basic geometrical and topological differences between two 3D CAD models.

4.6.1 Scenario

This second shape change assessment scenario involved the comparison of the original and modified versions of a mechanical part embodied by a reference and a target ISO STEP file (International Organisation for Standardization, 2011), respectively. All trials were to be performed using a single pair of STEP files comprising only explicit geometric data (B-Rep). No procedural representations were available for comparison.

Surface types now included planar, cylindrical, conic and B-spline surfaces. Figure 4.5 shows the geometry of both the original and the modified modeled parts, the latter appended with labels identifying the modified regions. Table 4.3 gives an outline of the thirteen (13) modifications applied to these specific regions, specifying if changes were to be expected in either the target model’s topology, geometry or both, and providing the resulting maximum deviation for each geometric difference.

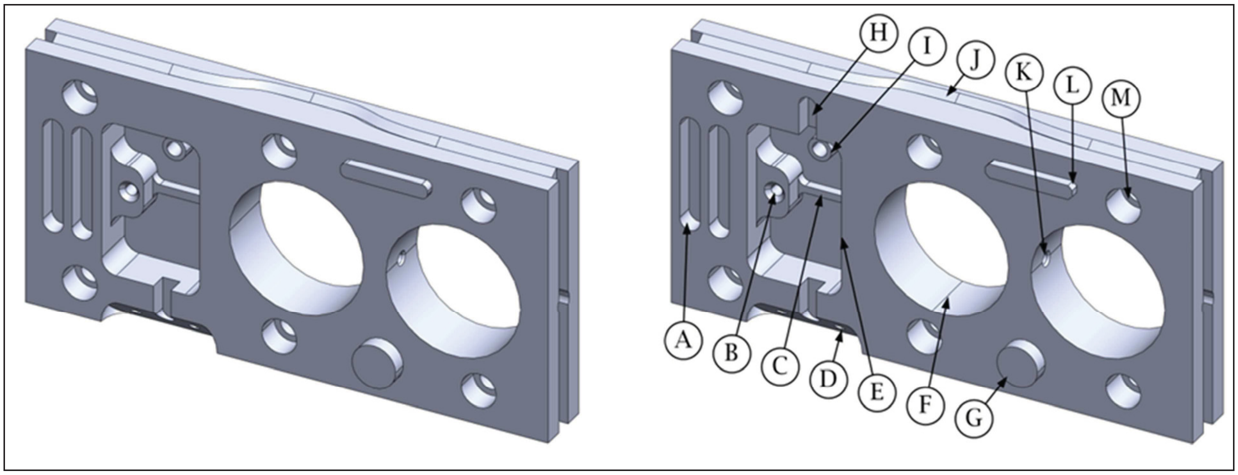


Figure 4.5 Geometry of the original (left) and modified (right, with the modified regions indicated) modeled parts used for the evaluation of software tools implementing an explicit geometry comparison for MDI purposes

Tableau 4.3 Outline of the modifications differentiating the reference and target 3D CAD models

		Type(s) of model change				*Estimated maximum*
Label	*Modification*	*Size*	*Location*	*Geometric*	*Topological*	*deviation*
A	Two-slot pattern, moved laterally		X			3.8 mm
B	Countersunk hole, diameter reduced	X				0.07 mm
C	Rib, width reduced	X			X	0.3 mm
D	Blind holes' bottoms, changed from conic to flat			X	X	0.8 mm
E	Pocket, width reduced (from labeled side)	X				1.6 mm
F	Through hole, inner faces split				X	--
G	Circular boss, moved and diameter reduced	X	X			1.3 mm
H	Blind slot, added			X	X	7.6 mm
I	Circular boss, tapered side			X		0.2 mm
J	Swept cut, modified trajectory			X		0.6 mm
K	Chamfer, surface equation changed			X	X	0.08 mm
L	Rectangular boss, rounds replaced by chamfers			X		0.2 mm
M	Six-hole pattern, counterbore's depth increased	X				2.5 mm

The purpose of this second shape change assessment scenario was primarily to locate the geometric differences between the two modeled shapes. The location of simple topological differences was also evaluated, as the topological equivalency of 3D CAD models' boundary representations may become critical in some given cases (e.g. NC tool paths calculation). A tolerance value of 0.05 mm was to be set for all trials to filter infinitesimal and/or irrelevant geometric deviations from the calculated results. Additional information on the located differences, such as counts, difference regions and deviation measurements, among others, was also to be investigated as such elaboration furthers the assessment of the engineering change between the two versions.

4.6.2 Evaluated software tools

The list of software tools that were evaluated within the second series of trials is presented in Table 4.4 and organized according to the different categories of 3D CAD model comparison tools proposed in Brière-Côté, Rivest et Maranzana (2012a). In some cases, more than one comparison function was available for trial; the names of these functions are also listed, making up for a total of twenty (20) trials.

Tableau 4.4 Software tools and functions evaluated for their explicit geometry comparison capabilities

Software tool	*Release*	*Evaluation license*	*Ref.*	*Trial*	*Evaluated functions*
3D CAD systems					
Autodesk® Inventor® Fusion	0.3		(Autodesk inc., 2010b)	1-	Change Manager
Dassault Systèmes CATIA® V5	R18 SP8		(Dassault Systèmes, 2007)	2- 3-	Graphical comparison Geometric comparison
Dassault Systèmes SolidWorks®	2010 SP4.0		(Dassault Systèmes, 2010)	4- 5-	Compare volumes Compare faces
Missler Software TopSolid®	2008		(Missler Software, 2008)	6-	Shape compare
PTC CoCreate® Modeling PE	2.0		(CoCreate Software GmbH, 2008)	7-	Compare parts
PTC Pro/ENGINEER®	Wildfire 5.0		(Parametric Technology Corporation, 2009)	8-	Compare by geometry
Siemens PLM NX®	7.0 MP1		(Siemens PLM Software inc., 2009a)	9- 10-	Model comparison Geometric comparison
SpaceClaim® Engineer	2011.1 SP1	X	(SpaceClaim Corporation, 2011)	11-	Color modified faces
3D CAD visualization and collaboration tools					
Actify SpinFire™	9.0	X	(Actify inc., 2010)	12-	Model Compare
Adobe Systems Acrobat® Pro 3D Reviewer®	9.4 SP4		(Adobe Systems inc., 2008)	13-	Compare
C4W 3D Shop ModelScan	2.8.4	X	(C4W, 2008)	14-	Compare
Lattice Technology XVL Studio© Professional	9.1a	X	(Lattice Technology Co., 2010)	15- 16- 17-	Simple shape compare View shape compare Difference detection
Oracle AutoVue® Electro-Mechanical Professional	20.0	X	(Oracle Corporation, 2010)	18-	Compare
3D CAD validation tools					
CapVidia CompareVidia	1.0	X	(CapVidia NV, 2010)	19-	Compare
Other 3D software tools					
InnovMetric Software PolyWorks®	11.0		(InnovMetric Software Inc., 2009)	20-	Inspect

Very few to no details were available to investigate the exact nature of each difference calculation approach employed, which is understandable in a commercial context. Since only deductions could be made from the representation and visualization of the comparison results for most of the evaluated tools, it was considered better not to identify the difference calculation methods systematically. As a substitute, brief descriptions of the graphical presentation of the comparison results are provided for each trial.

Trial #19 exceptionally involved a 3D CAD model comparison tool, CapVidia CompareVidia (CapVidia NV, 2010), primarily designed for CAD data translation validation. Though it does not naturally belong to the MDI solution domain, the focus of the evaluation trials, this particular software tool was still included in the trials as it does implement MDI capabilities to warrant validation results.

Also, trial #20 involved PolyWorks® from InnovMetric Software Inc. (2009), a computer-assisted inspection software capable of comparing sets of points representing actual measured parts to reference 3D CAD models. To exploit the software's comparison capabilities for MDI purposes, the target STEP model was first converted into a point cloud using the CATIA® V5 (Dassault Systèmes, 2007) STL Rapid Prototyping workbench and, then, compared to the reference STEP model.

4.6.3 Preprocessing and configurable settings

Some variance was observed among the evaluated software tools regarding the availability and/or the condition of the user-adjustable settings regulating difference calculation. For example, four (4) functions did not allow a tolerance value to be set by the user prior to being executed. Neither the software tools' user interfaces nor their respective help documentation provided evidence of the existence of such a parameter or of other parameters that could serve the same purpose. For the evaluation trials where no tolerance settings were available, it was therefore predicted that geometric deviations less than 0.05 mm, as per specified in other trials, could possibly be detected.

The computing accuracies of the various comparison functions were also regulated differently. Six (6) of the experimented functions provide parameters to regulate shape approximation or surface point sampling specifically for comparison. Eight (8) other functions relied indirectly on either their respective tools' display accuracy setting or on some previously-specified data importation settings. As for the six (6) remaining functions, the computing accuracy was not user-adjustable. Nonetheless, trials were conducted with computing accuracies set at their respective default setting. In the event of trials exhibiting issues regarding difference calculation recall (EC#1) for the given modifications, arbitrary computing accuracies would have been identified as a possible cause.

As a preliminary operation for the comparison process, all the tested functions required that the compared shapes undergo pose registration. However, this preprocessing was determined to be unnecessary since the reference and target models embodied versions of the same part and, therefore, were defined beforehand in the same coordinate system. Still, if pose registration had to be performed, fourteen (14) of the evaluated functions provided means to register the poses of the models prior to the comparison. Nine (9) functions provided means to align geometries according to either pre-defined (4) or transient custom-defined (5) coordinate systems, while five (5) functions allowed manual pose registration via external DMU editing operations.

4.6.4 Results

To begin with, two (2) of the model comparison tools listed in Table 4.4, along with their corresponding trials, presented critical issues related to their application in the simulated shape change assessment scenario. Essentially, both could not compare STEP files exclusively, as detailed here:

- Trial #1 on the Autodesk® Inventor® Fusion (Autodesk inc., 2010b) software could not be performed using the original pair of reference and target STEP files. The software's Change Manager module is designed to identify and convert explicit modifications applied to an evaluated (explicit) model back to a reference procedural representation, which cannot be provided via a STEP file.
- Trial #11 could not be performed on SpaceClaim® Engineer (SpaceClaim Corporation, 2011) either, because the tool's comparison functionalities can only be used on related versions of the same native-formatted 3D CAD model. Such particularity is typical of a static identity-based matching algorithm used for calculating shape differences. In

> the simulated scenario, the persistent identifiers exploited by this type of algorithm were inherently inexistent in the outsourced STEP files.

Table 4.5 presents a summary of the evaluation results for the twenty (20) trials, including trials #1 and #11 for which supplemental tests were performed and are discussed in Section 4.6.5. Besides each trial's evaluation regarding the five (5) evaluation criteria, the table includes brief descriptions of the graphical output for each comparison functions, identifies corresponding types of utility functions performing pose registration and, in specific cases, highlights which difference metric is provided.

Tableau 4.5 Results from evaluation trials of explicit geometric model comparison software tools

Software tool	*Trial*	*Graphical output*	*Pose registration*	*Metric*	*EC#1*	*EC#2*	*EC#3*	*EC#4*	*EC#5*
3D CAD Systems									
Autodesk® Inventor® Fusion	1-	Procedural changes*	--	Parameters	●●	●●●	●●●	●	●●
Dassault Systèmes CATIA® V5	2-	Superposed shapes	CSYS align	--	●	●	●	●●	●
	3-	Colored outlying voxels	CSYS align	--	●●●	●	●	●●	●●
Dassault Systèmes SolidWorks®	4-	Solids models	CSYS align	--	--	--	--	--	--
	5-	Classified faces	CSYS align	--	●●	●●●	●●	●●●	●●
Missler Software TopSolid®	6-	New faces	Custom align	--	●●	●●●	●	●●●	●●
PTC CoCreate® Modeling PE	7-	Classified faces	Custom align	Deviations	●●	●●●	●●●	●●	●●●
PTC Pro/ENGINEER®	8-	Out-of-tol. facets	--	--	●●	●●●	●	●●●	●
Siemens PLM NX®	9-	Classified faces	Custom align	--	●●●	●●●	●●	●●●	●●
	10-	Classified faces	Custom align	--	●●	●●●	●●●	●●	●●
SpaceClaim® Engineer	11-	Classified faces*	--	Dimensions	●●●	●●●	●●●	●●●	●●
3D CAD visualization and collaboration tools									
Actify SpinFire™	12-	Superposed shapes	Manual	--	●	●	●	●●	●
Adobe Acrobat® Pro 3D Reviewer	13-	Out-of-tol. facets	Manual	--	●●	●●●	●	●●●	●
C4W 3D Shop ModelScan	14-	Unique faces	--	--	●●●	●●	●	●●●	●
Lattice Technology XVL Studio® Professional	15-	Unique faces	Manual	--	●●●	●●	●	●	●
	16-	Superposed shapes	Manual	--	●	●	●	●●	●
	17-	Facets w/ color scale	Manual	Deviations	●●●	●●	●●●	●●	●●●
Oracle AutoVue® Electro-Mechanical Professional	18-	Superposed shapes	--	--	●	●	●	●●	●
3D CAD validation tools									
CapVidia CompareVidia	19-	Facets w/ color scale	--	Deviations	●●●	●●	●●●	●	●●●
Other 3D software tools									
InnovMetric Software PolyWorks®	20-	Out-of-tol. points	Custom align	Deviations	●●	●●●	●●	●●●	●

* Results from additional evaluation trials require to overcome incapability to compare STEP files (see Section 5.5).

One particular function failed to execute properly and, thus, no result are available for the corresponding trial (#4). It was observed that the regularized Boolean operations performed on the compared solids resulted in new solids with infinitesimal boundary entities that the processing geometric modeling kernel ultimately could not handle. Further investigation identified blended/rounded regions as one of the causes of the failure.

4.6.4.1 Calculation recall and precision (EC#1 & EC#2)

The aspect of difference calculation refers to the capacity of the software tools and their respective comparison functions to locate all (EC#1) and only (EC#2) relevant shape differences between the compared STEP files. Accordingly, seven (7) trials were rated with good calculation recall (EC#1) and ten (10) trials were rated with good calculation precision (EC#2).

Minor issues relating to calculation recall (EC#1) mostly concerned the location of modification 'A' and 'F'. Modification 'A' involved the leftward translation of the two-slot pattern by a distance equal to the width of the resulting rib between the slots. As pictured in Figure 4.6, this ultimately led to two interior planar faces from the slot features, one from the reference model and one from the target model, to possess equivalent boundary edges and equivalent surface geometric definition, except solely for their respective normal vectors, which were opposed.

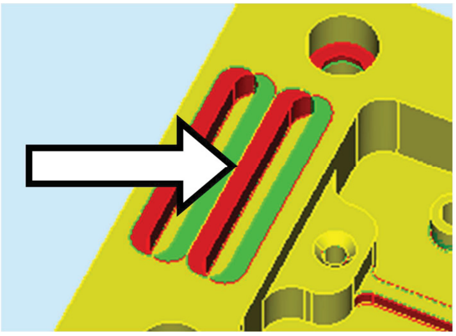


Figure 4.6 Coincident planar faces with opposed normal vectors from the two-slot patterns

This particular geometric singularity affected seven (7) trials in which the comparison functions inaccurately matched the two coincident yet unrelated planar faces and, thus, did not completely locate the difference. These particular functions calculate differences via the B-Rep entities of the compared models.

Modification ‘F’, which involved only a topological difference in the subdivision of a cylindrical face, was simply not located in eight (8) trials. For some of those trials, it was observed that the software tools altered the models’ topology prior to comparison. As a result of their respective STEP file importation processes, modification ‘F’ was systematically removed, cancelling beforehand the topological difference that could then not be located. This issue could be resolved by appropriately configuring, when possible, the tools’ STEP file importation modules in order to preserve the STEP models’ topology during the importation process. Still, the observed behavior highlights the possible loss and/or degradation of original 3D CAD data when submitted to a translation process. When comparing 3D CAD models, such data degradation originating from the model comparison tool’s processing of the CAD data would logically overthrow the process that it is precisely designed to perform.

Poor calculation recall was attributed in four (4) trials as the corresponding functions overlooked differences bearing small deviations (e.g. modifications ‘B’, ‘I’, ‘K’ and ‘L’). No tolerance value could be set prior to comparison for these specific functions, which led to conclude that the default settings were greater than the preset 0.05 mm for all other trials.

Concerning calculation precision (EC#2), no clear distinction could be made in some trials between the geometric differences due to the explicit modifications described in Table 6.3 and those resulting indirectly from parent or neighboring modified features. In a scenario involving the detailed assessment of engineering change, such a distinction is key. However, the most significant case of poor calculation precision (EC#2) involved modification ‘F’ and five (5) comparison functions relying on approximated shape (via tessellations or voxels) to calculate differences. As pictured in Figure 4.7, the comparison of two locally equivalent geometries, which were approximated differently due to the inherent topological difference, notably led to the visual location of false geometric deviations within the central large opening of the target model. In a few trials, differences were also located on some other cylindrical faces that did not present any topological differences.

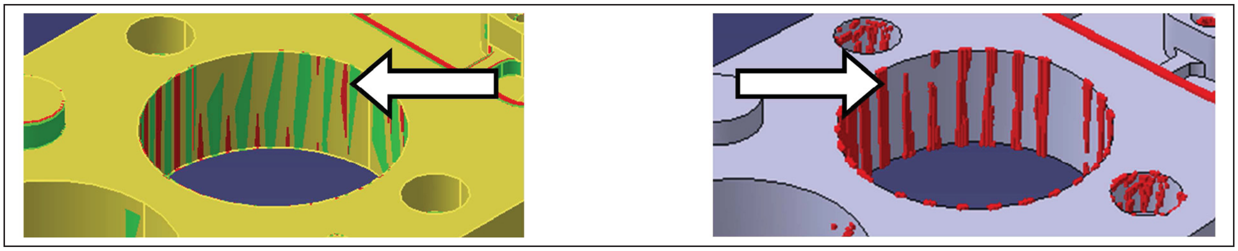


Figure 4.7 Representations of erroneous geometric deviations due to shape approximation: (left) Superposed 3D tessellated shapes, (right) Voxel approximation

Additional trials focused on these specific functions showed that this problematic behavior could be avoided by increasing or “loosening” the difference tolerance value. However, the downside of such a work-around is that other small yet critical differences will be wrongfully filtered out and thus overlooked. Increasing the computing accuracies of these functions prior to comparison, when possible, reduced the number of false geometric deviations, but also increased computing time, especially in the case of voxel approximation.

4.6.4.2 Representation range and accuracy (EC#3 & EC#4)

The range of difference representation (EC#3) relates to the amount and quality of information provided by a comparison function in the description of the located differences, while the accuracy (EC#4) characterizes its capability to rightly describe such differences. Generally, it was observed that simple, but poor representation ranges, e.g., unchanged/changed or added/removed classifications, were mostly associated with accurate representation capabilities. Conversely, comparison functions providing more elaborated differences classifications and descriptions were found less accurate in identifying the correct description of the differences.

Good representation range was notably attributed to comparison functions promptly providing metrics in the form of parametric or dimensional differences, or in the form of local geometric deviations. In many cases, such measures could be obtained using external measuring utility functions, but there were not taken into account in this particular evaluation due to the fact that they were not part of the actual comparison functions.

Minor issues leading to some trials' fair representation accuracy (EC#3) ratings include side effects of poor calculation precision, i.e., the description of false differences, and inaccuracies in faces classifications. For example, large geometric deviations between corresponding faces from reference and target models (e.g., modifications 'A' and 'M') seldom prevented them to be matched and accurately described as modified. Poor representation accuracy (EC#3) involved more significant disorganization in face mapping and classification.

The representation range and accuracy greatly affect the application of a giving function in a shape change assessment scenario. For example, modifications 'C' and 'E' involved the perpendicular translation of planar faces in the large pocket area. As a result, the blended faces from related round features were also identified as being geometrically different, in some cases even topologically, even though no modifications were applied directly to the round features. Since not enough additional information is provided, it is impossible to identify key details such as whether the radii of the round features were actually modified or if they were simply moved along with a tangent face.

4.6.4.3 Visualization discernability (EC#5)

Good visualization discernability was attributed to trials in which the graphical output of the comparison could be easily explored and inspected by means of selectable difference lists, elaborated viewing filters or even cross-sections. Those at least providing separate synchronized views of the reference and target models were given a 'fair' rating.

The nine (9) remaining trials presented the comparison results via the visualization of superposed 3D shapes, which notably rendered the visual examination of modification ‘A’ impossible, as shown in Figure 4.6. Accordingly, comparison functions relying solely on 3D shape superposition presented visualization discernability (EC#5) issues when it came to locating small yet pertinent geometric differences. In the specific cases of modifications ‘B’ and ‘L’, differences were actually correctly located. However, since the comparison results were exclusively visual, the superposed models’ views had to be zoomed in considerably to be observable. This is obviously impractical, because the exact location of the differences cannot be known beforehand. The risk of potentially overlooking relevant differences in cases where the comparison takes place between larger and more complex 3D shapes is therefore significant.

4.6.5 Additional evaluation trials

Additional trials were performed on both Autodesk® Inventor® Fusion’s Change Manager (Autodesk inc., 2010b) and SpaceClaim® Engineer (SpaceClaim Corporation, 2011) to round out the second series of evaluation trials. The goal was again to evaluate the tools’ MDI-capabilities, but without regard to their inability to compare two outsourced models. In both trials, the shape change assessment scenario was altered to involve the same two versions of the part, but in native formats instead.

On aspects such as calculation recall (EC#1) and precision (EC#2), the additional trials lead to satisfying results, with at least all twelve (12) modifications bearing a geometric outcome on the modeled shape being exclusively and accurately located. Also, good results were generally observed relating to representation range (EC#3) and accuracy (EC#4), as compared to the results of the eighteen previous trials. For example, SpaceClaim® Engineer provides a useful dimensioning function to measure user-defined part dimensions simultaneously on both the reference and target shapes, helping users to further elaborate on the description of a shape change.

By relating explicit geometric differences back to a reference procedural representation of the model, Autodesk® Inventor® Fusion's Change Manager enables some shape differences to be elaborated in terms of revised modeling operations and corresponding parametric modifications. A distinction between source and impacted geometric differences, such as in the cases of the translated blended faces induced by modifications 'C' and 'E', is thus possible. However, the description of differences according to this elaborated representation range is far from accurate. Since converting explicitly represented modifications into procedural form is not a simple task, only simple isolated modifications are accurately represented, the remaining being insufficiently converted into low-level boundary edits.

4.7 Conclusion

Any decision regarding the selection of the proper model comparison tool must be influenced primarily by the problem at hand and the product lifecycle process to provide for. The results presented in this paper from the two series of evaluation trials performed on commercially available 3D CAD model comparison tools confirm such assertion. As it was intended to assess how these software tools could efficiently contribute to the shape change transposition problem, the significant variance observed in the model difference calculation algorithms implemented and, consequently, in the comparison results presented, leads us to discern that no MDI technology can be applied conveniently whatever the model comparison scenario.

Five evaluation criteria were defined to symbolize the basic requirements of two simulated shape change assessment scenarios, emphasizing on the need for detailed and significant comparison results in the engineering change management application domain. As far as geometric differences could acceptably be located (EC#1) between versions of either procedural or explicit 3D CAD models, major concerns arisen relating to other criteria:

- Comparing procedural models will provide more insight on the rationale for a shape change as model differences are located and measured in terms of semantically valued modeling operations, upholding the representation range criterion (EC#3). Yet, the first series of evaluation trials revealed concerns for both difference calculation precision (EC#2) and representation accuracy (EC#4) for the evaluated comparison tools, as no clear distinction could be made between modified operations and those geometrically or chronologically impacted by the modifications.

- The use of shape approximations to calculate model differences between explicit 3D CAD models must be avoided in shape change assessment scenarios. On top of providing approximate measures of the located differences (EC#4), it generated geometric “noise” that could not be distinguished from small yet possibly significant geometric deviations given the context (EC#2). The use of exact shape representations, such as B-Rep, must therefore be preferred.
- MDI tools relying exclusively on the graphical representation and visualization of model differences are exposed to critical discernability issues (EC#5). The absence of simple reporting functions in some comparison tools, usually provided to support the graphical visualization of differences, renders some accurately located shape differences unnoticeable by the common user, unless he knows beforehand where to look, which is impractical in a shape change assessment scenario.

Shape change transposition – i.e., enabling rapid and reliable decision making in a prescribed use context via the adequate representation of 3D CAD model differences – ultimately calls for precise difference calculation and elaborated difference representation. In the light of the observations stemming from the two series of evaluation trials and the conclusions drawn from our previous review (Brière-Côté, Rivest et Maranzana, 2012a), the next generation of MDI solutions capable of supporting shape change transposition ought to integrate the precision and flexibility of existing explicit B-Rep-based calculation methods with a representation range comparable to the one of procedural CAD model comparison. Notably, description of model differences at the level of actual engineering semantics must be made possible for the comparison of application-specific, thus heterogeneously formatted, 3D CAD models.

Furthermore, difference representation and visualization constitute two parts of the MDI problem which must be addressed separately as much as possible, as it was observed that the joint graphical approach leads to precision and discernability issues. Representation of 3D CAD model differences should not be considered as a transient stage between the calculation and the visualization of comparison results, but as the keystone of efficient difference analysis and subsequent manipulation. It is the authors' opinion that a proper representation of calculated model differences is at the basis of any good visualization scheme, whatever the application for 3D CAD model comparison.

CHAPITRE 5

A THREE-STEP APPROACH FOR STRUCTURING 3D CAD MODEL COMPARISON SCENARIOS

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article présenté lors de la « *8th International Conference on Product Lifecycle Management* »
ayant eu lieu du 11 au 13 juillet 2011 à Eindhoven au Pays-Bas.

5.1 Abstract

One aspect of PLM involves the search and reuse of products, parts and information to reduce costs and delays while lowering risks. Since 3D CAD tools and models have become prevalent in the mechanical product lifecycle, using 3D CAD models as a key to searching and comparing objects from the PLM vault is a promising avenue, and numerous scenarios will benefit from such a capability. For instance, comparing 3D CAD models of mechanical components to assess their relative shape difference leads to benefits in areas such as design reuse, sourcing, engineering change management and data interoperability. Fundamentally, depending on the scenario, the function of the comparison will vary: engineering change management entails documenting the differences, while data sharing implies ensuring equivalency. Hence, this paper presents a three-step approach for structuring 3D CAD model comparison scenarios and analyzing the characteristics that are pivotal to the selection or the design of an appropriate solution approach. The goal is to ultimately represent a scenario's defining factors through the use of a meta-model designated here as PDMF4C. As an illustrative example, this paper details the use of the proposed approach to describe the recognizable 3D CAD model comparison scenario of design reuse via shape-based retrieval.

5.2 Introduction

In the present state of market globalization, product data authors and product data consumers face the challenges of collaborative and integrated product development. Today's PLM solutions need to manage, store and distribute the definition of a product, as 3D data is being shared and used more frequently and intensively than ever before. For example, 3D CAD models are increasingly used as inputs to retrieve and compare products, parts and related information from PLM vaults to ultimately enable product data reuse. Innovative PLM supports for the creation, editing, exchange and manipulation of 3D data are increasingly relied upon to stimulate fast and reliable decision making throughout a product's lifecycle.

A comprehensive survey of 3D CAD model comparison scenarios was completed as part of a broader project on the problem of engineering change transposition between differently formatted 3D CAD models. It revealed that the generic process of comparing 3D CAD models can bring a variety of benefits to multiple scenarios in the development of mechanical products such as design reuse, sourcing, engineering change management and data interoperability. However, a concurrent survey on 3D CAD model comparison methods and available tools revealed that, whereas some specific scenarios such as the validation of translated 3D CAD data have been addressed with purpose-built tools (e.g. Kubotek USA (2010), CAx Implementor Forum (2008), Translation Technologies Inc. (12 décembre 2006)), appropriate solutions for many of the surveyed scenarios are still to be achieved.

Moreover, the heterogeneity of 3D CAD model comparison scenarios have led to the development of methods and tools addressing very specific subsets of scenarios sharing common distinctive traits. Such defining factors need to be systematically identified and organized prior to one's decision to opt for one of the existing solutions or, ultimately, to develop a custom-built comparison tool.

This paper presents a three-step approach designed to provide a structured representation of a scenario's defining factors: (1) the basic function of the comparison, (2) the compared models' respective forms, contents and relationships, and (3) the inquiring/inquired engineering processes. Accordingly, this approach relies on the proposed Product-Definition-Model-Formalism for Comparison (PDMF4C) meta-model. Inspired by Caplat's theory on generic meta-modeling (2008), it is intended to capture the set of concepts, characteristics and links that characterizes a 3D CAD model comparison scenario. This approach could benefit, for example, engineering managers planning the migration of 3D CAD data or promoting design reuse within their organizations.

This paper is organized as follows. Section 5.3 categorizes different 3D CAD model comparison scenarios that have already been described and/or addressed. Section 5.4 introduces the proposed structuring approach and related PDMF4C meta-model by describing a comparison scenario's three defining factors. Finally, section 5.5 exemplifies the use of the proposed meta-model on the typical 3D CAD model comparison scenario of design reuse and on the corresponding solution involving shape-based retrieval of similar parts.

5.3 Review of 3D CAD model comparison scenarios

Numerous scenarios can be found that take advantage of comparisons between 3D CAD models, as much in CAD- and PLM-related scientific literature as in commercial documentation. Overall, whether the comparison is done approximately or in detail, it is generally intended to support the work of specialists during a particular engineering process via the assessment of the two modeled 3D shapes' similarity or difference. We classify 3D CAD model comparison scenarios into six *application domains*:

- CAD data translation/remastering,
- Product information reuse,
- Engineering change management,
- CAD modeling management,
- Product rationalization and standardization, and
- CAx model authoring.

We present the first three application domains, as they include the scenarios that are the most frequently referred to in the literature.

5.3.1 CAD data translation/remastering

A first application domain comprises scenarios involving 3D CAD model comparison as the key for the geometric validation of CAD data translation and CAD data remastering. The purpose of the comparison is to verify the geometric equivalency of two related CAD models; i.e. to ensure that the authoritative 3D data from a source file was adequately and accurately reproduced in a new target file (CAx Implementor Forum, 2008; Frechette, 1996). This type of validation is required because of the possible loss or degradation of authority data during the translation (automated) or the remastering (manual) processes. The differences thus located are treated as detrimental and will normally cause the validation to fail.

3D shape comparison methods used for geometric validation are usually metric-based. The compared metrics may be (CAx Implementor Forum, 2008; Frechette, 1996):

- global properties, such as the shapes' volume, surface area or centroid;
- entity counts, such as the number of faces, edges or surface types; or
- local measurements, such as the maximum deviation between the models' boundaries (also known as the points cloud or "point-to-part" method).

Metric-based comparison methods provide pass/fail diagnosis on geometric equivalency between heterogeneously formatted models. Some are notably being implemented as "Geometric Validation Properties" (GVP) in application protocols AP203 (International Organisation for Standardization, 2011) and AP214 (International Organisation for Standardization, 2010) of the ISO STEP standard for product data exchange.

A family of software tools has emerged to address the specific purpose of geometric validation following 3D CAD data translation by means of 3D CAD model comparison. These geometric validation functionalities are generally complementary to 3D CAD data translation engines. They may also be paired with product data quality (PDQ) validation functionalities that check source and target models beforehand for geometric defects that often cause CAD data translation and, thus, geometric validation to fail.

In addition to being capable of reading multiple CAD data formats, geometric validation tools are required to maintain the two compared data sets' integrity prior to their being compared. Any loss or degradation of data that would originate from the validation tool's data processing itself would overthrow the process it is designed to perform. The approaches used to address this issue are either to process the compared data using licensed libraries published by the major CAD systems' editors (e.g. CapVidia NV (2010), CT CoreTechonologies Group (2008)), or to operate the originating CAD systems via their API to access and process the compared data (e.g. ITI TranscenData (2010a), Translation Technologies Inc. (2010)). While the former approach benefits from not requiring costly 3D CAD system installations and licensed seats for the comparison tool to operate, the latter ensures that the CAD data is fully and accurately read by the originating systems, as licensed libraries may be voluntarily left incomplete by issuing parties for competitive motives.

5.3.2 Product information reuse

As a solution for reducing costs and delays while lowering risks, one aspect of PLM features the retrieval and reuse of parts, products and associated information. 3D CAD model comparison is key to overcoming the challenge as shape can now be perceived as a neutral and effective language to represent and retrieve product data (Li, Liu et Ramani, 2004). The problem of retrieving product data based on shape is divided into two steps, each with distinct objectives (Msaaf, Maranzana et Rivest, 2007).

The first step is to locate similar parts in the PLM vault, which may contain thousands or hundreds of thousands of parts represented by their 3D CAD models. A model selected as a search key will be compared to other models in order to evaluate their similarity via a given metric which, in turn, will be used to identify and sort a subset of the compared models that can be considered as similar.

The typical shape-based retrieval approach does not involve the comparison of the 3D CAD models themselves. Instead, lightweight shape signatures are being extracted beforehand or “off-line” for each 3D CAD model and stored as meta-data. Large amounts of 3D shapes can therefore be compared quickly to each other and qualitative similarity measures can be computed efficiently, since only the shape signatures are compared.

Many shape signatures, along with extraction and similarity measure algorithms, have been proposed in recent years and duly reviewed (Cardone, Gupta et Karnik, 2003; Iyer *et al.*, 2005). Shape signatures are high-level abstractions of 3D shapes and, therefore, possess reduced discrimination capabilities. Unless it is expressly so designed, it is difficult to predict how well a particular shape signature will perform in a given scenario. For instance, machining feature-based shape signatures (Cicirello et Regli, 2001; Ramesh, Yip-Hoi et Dutta, 2001) will suit shape-based retrieval tasks better than other shape signatures when the reuse of existing machining processes is the objective.

The second step of product information reuse is to detail the differences between the CAD models considered to be similar as a result of the first step. The previously computed shape signatures, given their reduced discrimination capabilities, are discarded and pair-wise comparisons of each retrieved candidate model with the search key are performed. The goal is to determine if the retrieved pairs of CAD models can rightfully be identified as similar as per the particular reuse objective.

For instance, to determine if a retrieved existing sourced part can appropriately replace another one in a design, sufficient evidence of their interchangeability must be compiled. For that purpose, pair-wise 3D CAD model comparison enables the identification of a common base between two designs, validating that features and/or dimensions, regarded as critical from the viewpoint of the inquiring engineering process, are present in both.

5.3.3 Engineering change management

Engineering change management represents another domain of PLM that benefits from 3D CAD model comparison. Numerous scenarios requiring a detailed assessment of the modifications applied to a part or a product model can be identified; typical ones include:

- the execution of engineering change orders (ECO) (Al-Sabeh, 2004; St-Martin, 2001),
- the impact analysis of a design's evolution on downstream models (Sypkens Smit et Bronsvoort, 2007),
- the automated propagation to downstream models of a change applied to a part's 3D geometric definition (François et Cuillière, 2000), and
- the detection of unauthorized or detrimental modifications applied to 3D CAD or other 3D CAx models (ITI TranscenData, 2010a; Kubotek USA, 2010).

Leading 3D CAD systems such as Dassault Systèmes' SolidWorks® (Dassault Systèmes, 2010) or PTC's Pro/ENGINEER® (Parametric Technology Corporation, 2009) offer model comparison functionalities specifically designed to support tasks related to engineering change management. Available feature-based comparison functions enable the location and the evaluation of differences at the parametric level between two CAD models as represented by their respective modeling histories or feature trees. However, the issue of the non-uniqueness of part representations restricts the applicability of such functions to scenarios involving the comparison of related CAD models representing versions of a single part's evolving definition.

3D CAD visualization and collaboration tools also incorporate some model comparison functionalities specifically to support the diffusion of engineering changes throughout an extended organization (e.g. Adobe Systems inc. (2008), Oracle Corporation (2010)). Accordingly, comparison results can only be visualized. Capabilities regarding the characterization of localized differences remain slight because only lightweight or approximated shape models can be compared. As for geometric validation tools, engineering change management scenarios will sometimes be included in their respective sets of possible uses; but, since it is not their primary purpose, their relevance in such scenarios remains limited.

5.4 Structuring 3D CAD model comparison scenarios

Due to the heterogeneity of scenarios, the selection or the development of an implementation approach for 3D CAD model comparison must be addressed piece-wise. Defining factors such as the purpose, the nature and the composition of the compared data, as well as the inquiring engineering process are key and highly interrelated.

This section presents the three steps of the proposed approach to structuring 3D CAD model comparison scenarios. It also presents the Product-Definition-Model-Formalism for Comparison (PDMF4C) meta-model for representing pre-existing commonality and relationships between compared CAD models.

A high-level UML formalism is used to represent the PDMF4C meta-model. Hence, although set at the conceptual level, the information captured by this meta-model should be rich enough to represent any of the scenarios despite their heterogeneity. It should then provide insight on what should be accomplished next, whether it involves selecting an existing solution (e.g. commercially available tools), or defining the requirements for the development of a new solution approach.

5.4.1 Identifying the basic function of the comparison

The particular function of a comparison between two models will significantly vary depending on the scenario. It constitutes the first defining factor for any 3D CAD model comparison problems as it enables the following preliminary identification of a solution domain and of some basic requirements to assimilate.

Six basic functions for 3D CAD model comparison problems have been identified. As summarized by Table 5.1, each basic function relates to an elementary question that the process of comparing 3D CAD models is expected to answer. Identifying the basic function also provides insight on how the expected result should be expressed.

The identification of the appropriate basic function can be achieved by characterizing two specific aspects of a 3D CAD model comparison problem:

- *cardinality* – a reference model may either be compared to one single target model (1-to-1, or pair-wise) or to many models (1-to-N) coming from usually large sets; and
- *level of detail* – the amount and the accuracy of information expected from the comparison will vary according to the intended use, ranging from simple "Yes-No" or "Pass-Fail" diagnosis to detailed measures of the located differences.

Tableau 5.1 Basic functions of model comparison

Basic function	**Basic question**	**Expected result**
A. Find duplicate	Which models are equivalent?	Finite sets of objects
B. Find similar	Which models are similar?	Ordered, scale-based distributions
C. Detect difference	Are the models different?	Binary (Yes/No, Pass/Fail, etc.)
D. Estimate difference	How different are the models?	Qualitative, scale-based values
E. Locate differences	Where are the differences?	Graphical reports describing regions, loci
F. Measure differences	What are the differences?	Classifications, measures, detailed descriptions

Accordingly, relating both the level of detail and the cardinality allows us to distinguish each of the six basic functions for comparing 3D CAD models, depicted in Figure 5.1. Each of these two aspects has an important influence over the problem's solution domain. As it regulates the quantity of comparisons performed in a single occurrence of the scenario, higher cardinalities justify a solution boasting good computational efficiency. On the other hand, high levels of detail understandably command far more methodical difference calculation algorithms and, consequently, more complex difference representation systems.

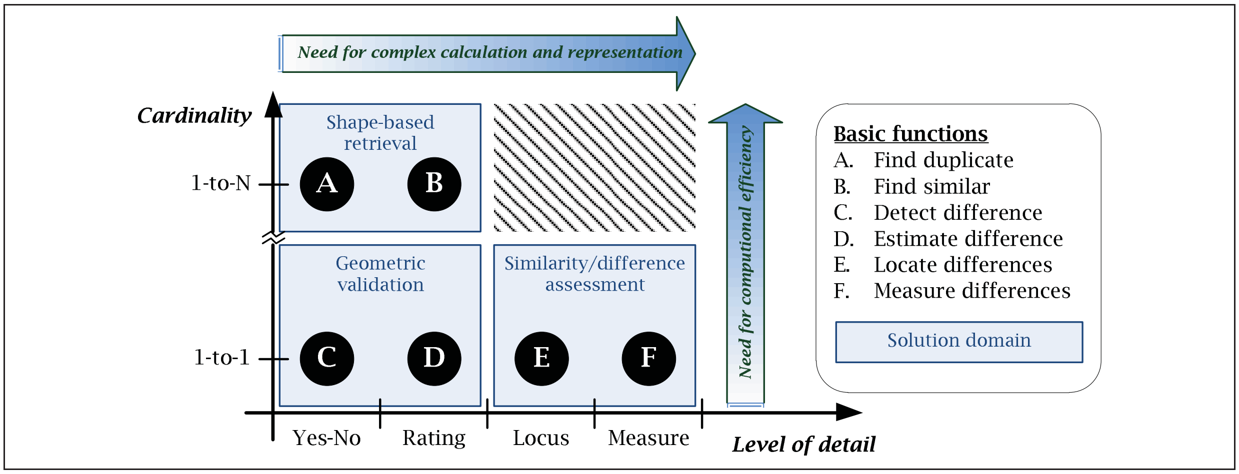


Figure 5.1 Basic functions and their relation to the required level of detail and cardinality

Three solution domains are identified and related to the six basic functions on Figure 5.1. Hence, finding duplicate or similar models within large sets of 3D CAD models relates to shape-based retrieval, the corresponding similarity measures being of qualitative nature, at most. Comparing two models to either check their equivalency according to some given context-specific criteria or to estimate their relative difference with reference to specific characteristics, i.e. to provide a qualitative appraisal of how close or far they are from each other, involves geometric validation. Then, when distinguishing differences between two models, providing their respective locus in relation to the modeled shapes and, furthermore, when categorizing them or measuring their geometric magnitude, we refer to the detailed assessment of the models' similarities or differences.

5.4.2 Developing the reference/target relationship

Fundamentally, a relationship must already exist or must be defined between two CAD models for them to become the subjects of a comparison. Informally, such a relationship is established when one of the models is identified as the reference model; for instance, the query shape in shape-based retrieval or the source file in the geometric validation of a CAD data translation task.

Conceptually, we introduce a representation of this relationship within the PDMF4C meta-model through the definition of the **Comparison** association that relates two **Model** instances. As represented in Figure 5.2, the attributes of the **Comparison** association are defined in order to allow the capture of the comparison scenario's first defining elements, i.e. the basic function of the comparison and the related aspects of level of detail and cardinality.

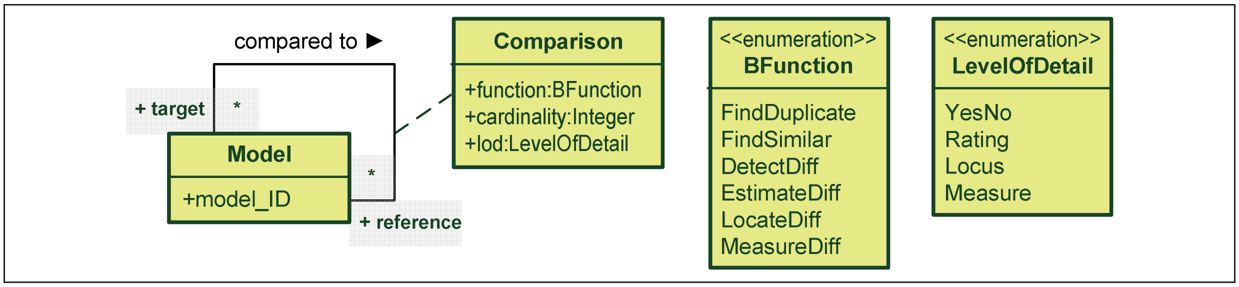


Figure 5.2 Representation of the comparison association

The reference/target relationship between the compared models is obviously central to the comparison problem. Still, it is assumed that a more elaborate representation of a 3D CAD model comparison scenario via the PDMF4C will add to the input and significance of such a relationship in the development of an implementation approach. Besides providing directionality to the comparison, this meta-model may also reveal preexisting elements of commonality between the models – for example, in how they are represented, what information they convey and to what intent.

5.4.2.1 The four core concepts of modeling

To organize this facet of the 3D CAD model comparison problem, we were inspired by Caplat's theory on meta-modeling (Caplat, 2008). In Caplat's theory, four core concepts involved in fundamental modeling are identified and related, as depicted in Figure 5.3: the *subject*, the *language*, the *point of view* and the *model* itself.

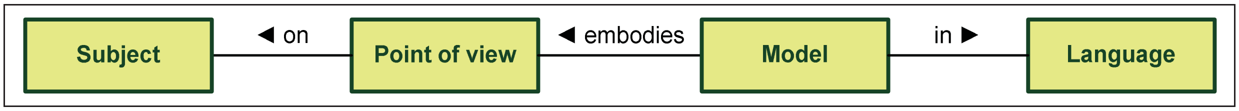


Figure 5.3 The four core concepts involved in modeling according to Caplat (2008)

The *subject* represents objects, individuals or situations that form a set because they relate through recognizable common traits as per the *point of view*. The *model*, itself a constructed object, embodies a *point of view* in accordance with the rules imposed by the *language* it is expressed in – rules which are a priori independent of the *subject*. Basically, every model is a form representing content – a concrete and analyzable representation conveying a meaning, a means of sharing knowledge.

In the specific context of CAD and, correspondingly, of the proposed PDMF4C meta-model, the **Product** constitutes the modeling subject. Accordingly, the CAD **Model** embodies a temporarily and contextually set **Definition**, a particular point of view on the expected features and functions of the **Product** and **Product_Instances** projected to be manufactured, by detailing a finite set of **Specification** instances. It is expressed in a given and sometimes tool-specific CAD **Formalism**. From the perspectives of engineering and product development, the CAD model is a communication tool for the different actors in the design, manufacturing and other processes of the product throughout its lifecycle.

The four CAD-specific core concepts of **Product**, **Definition**, **Model** and **Formalism**, represented in Figure 5.4, constitute the core elements of the PDMF4C meta-model for representing pre-existing commonality and relationships between compared CAD models. Accordingly, the development and refinement of the informal reference/target relationship is to be approached from two perspectives: via the models' respective product definitions that they embody or via the formalisms they are respectively expressed in.

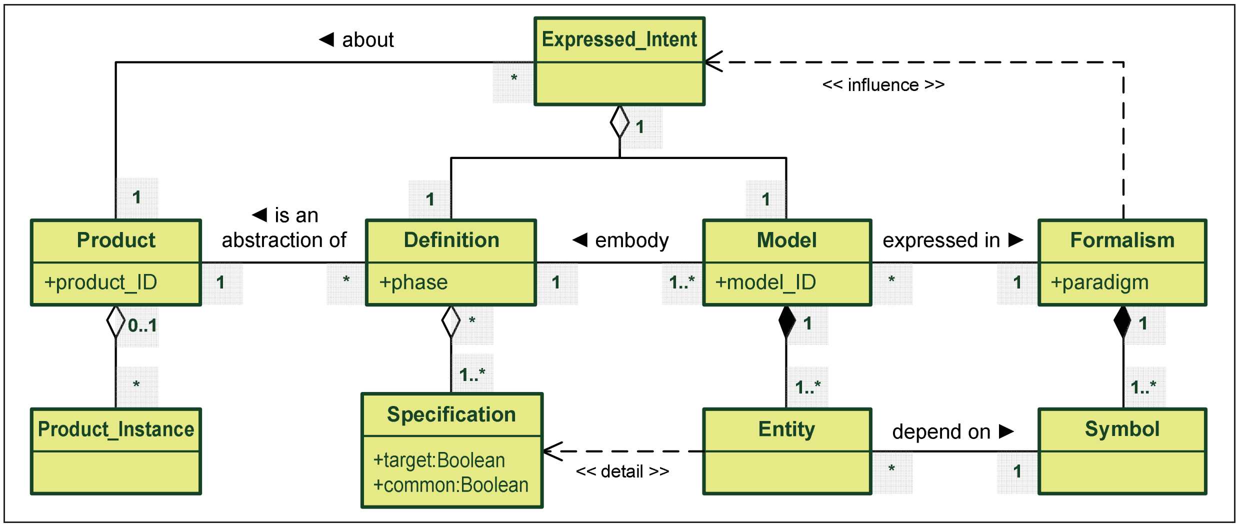


Figure 5.4 Representation of the PDMF4C meta-model core concepts

5.4.2.2 Relating the compared model's definitions

Developed from a finite set of **Specification** instances, a **Definition** is regarded as the subjective and flexible element of product modeling. For instance, at the beginning of the conceptual design phase, the definition may only include initial requirements about the product, which implies a high level of abstraction. It must, however, evolve considerably to reach the status of detailed and released design, which implies a much lower level of abstraction. Even though they are abstractions of the same product, a product's conceptual and detailed definitions will be represented as distinct **Definition** instances in the meta-model. They represent distinct sets of specifications, even if the latter encompasses the former.

As an outcome of a 3D CAD model comparison scenario, common subsets of specifications, from the single instance to an entire product definition, will normally be either validated or identified. For instance, in the geometric validation of translated 3D CAD data, comparison of the source and target models is normally required to validate that both models systematically embody the same geometric definition of a product – a typical subset of specifications embodied by 3D CAD models.

Clearly identifying target specifications through an appropriate level of aggregation is advisable. In the PDMF4C meta-model represented in Figure 5.4, **Specification** instances are characterized by two attributes: the ‘target’ attribute aims at identifying **Specification** instances that represent the objects of the comparison, while the ‘common’ attribute aims at identifying the specifications that are expected to be common between the two models.

5.4.2.3 Relating the compared model’s formalisms

To embody a **Definition**, a **Model** instance must be expressed in a particular **Formalism**, which stands for the core modeling concept of *language*. In the PDMF4C meta-model, an element of a model is called an **Entity**. It is based on a particular **Symbol** from the **Formalism** the **Model** is expressed in, and it is structured according to a defined grammar. Formalisms have a determining influence on the capability to express intent (**Expressed_Intent**) about a product. That is, symbols available to express a product’s definition and its specifications via the CAD model are fixed at the moment a formalism is used, both in their form and in their semantic use.

3D CAD file formats constitute examples of distinct formalisms. Modeling paradigms, represented here by the ‘paradigm’ attribute of the **Formalism** concept, define the foundations – a coherent set of modeling object classes, their respective features, their relationships, the rules of use, the constraints, etc. – on which one or many formalisms and one or many methodological approaches to modeling rely. In 3D CAD, boundary representation (B-Rep), constructive solid geometry (CSG) and tessellation represent examples of modeling paradigms. Accordingly, the STEP AP203 formalism (International Organisation for Standardization, 2011) is based on the B-Rep paradigm, as is the Parasolid® formalism. Graphical representations of 3D CAD models must be considered as related, yet distinct models, since they are expressed in formalisms based on a different paradigm (tessellation).

Figure 5.5 details how 3D CAD formalisms may relate via the representation of a third section of the PDMF4C meta-model. 3D CAD formalisms are implemented or processed by

Software_Tool instances. Depending on the function they implement, they can be classified into two subtypes of tools relevant to 3D CAD model comparison scenarios: **Translator** instances or **Editor** instances. **Software_Tool** instances are usually integrated into **System** instances such as 3D CAD systems.

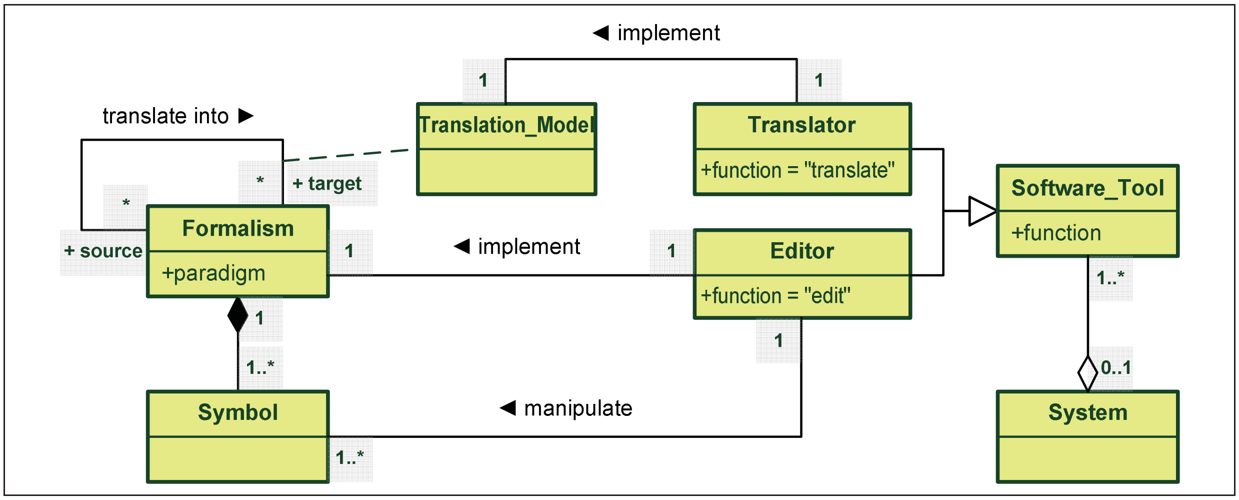


Figure 5.5 Representation of the **Formalism** and **Software_Tool** concepts

The function of a software tool classified as an **Editor** is to enable the creation and modification of a model by instantiation and manipulation of a particular formalism; consequently, it manipulates the symbols belonging to the formalism it implements in accordance with its rules and grammar. Likewise, as depicted in Figure 5.5, a **Translator** translates 3D CAD models by implementing a **Translation_Model** that associates elements of a source formalism (symbols, rules, grammars) to a target formalism.

Translation models are not unique. Many translators may be available to perform, with divergent results, a single CAD data translation task. It is also acknowledged that these models may not be optimal, either. Symbols may be abstracted or left unprocessed due to the source and target formalisms' different abilities for expressing specifications. However, it is important to distinguish the scenarios where the absence of a bijective translation model is considered to be detrimental – e.g. the well-known issue of loss or degradation of data as a result of its translation (CAx Implementor Forum, 2008) – from those where it is intended –

e.g. the extraction of shape signatures in shape-based retrieval scenarios, as further detailed in the example of Section 5.5.

5.4.3 Identifying the inquiring and inquired processes

Even though it represents a single concept, the identification of the processes in the PDMF4C meta-model is important and necessary. **Process** instances embody branches of knowledge with their own distinct reasoning approach and interpretation rules regarding the product in development. As represented in Figure 5.6, they relate to the definition of a 3D CAD model comparison scenario via three already-defined concepts: the **Expressed_Intent**, the **Software_Tool** and the **Comparison**.

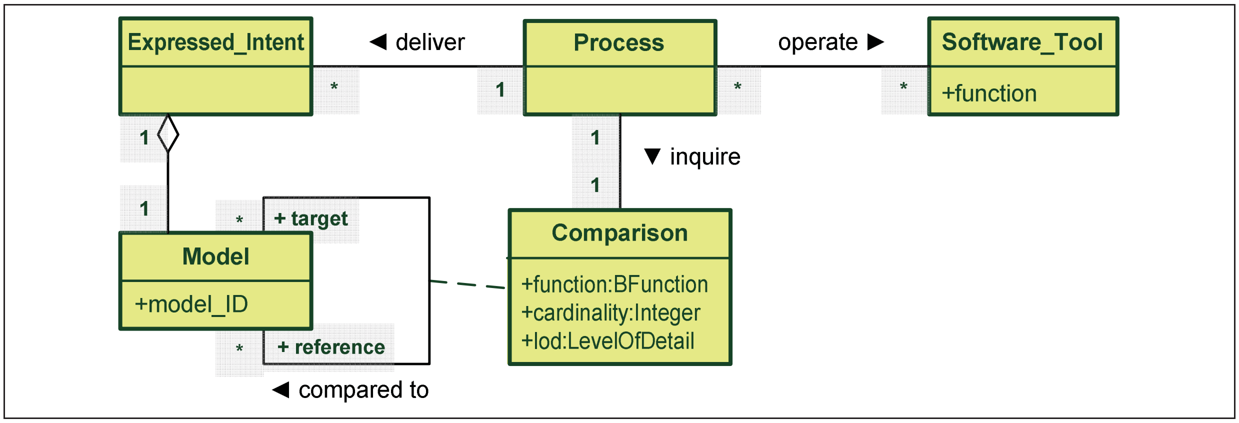


Figure 5.6 Representation of the **Process** concept

When engaged in product development, a process delivers one or many expressed intents about the product. It owns the particular point of view on the product's expected features and functions which develop into a definition and are materialized by a model. Hence, identifying the process that delivered a model subject to a comparison amounts to identifying one of the comparison's inquired processes. As two 3D CAD models are compared, the inquired process or processes' interpretations of the product need to be considered and, thus, weigh in on the calculation and the representation of the differences.

Also, processes within an organization exploit or contribute to the 3D CAD models of new or existing products through the realization of specific tasks by means of software tools. These tools implement or process specific formalisms which have a real influence on the initial expression of intents about the product. Thus, the association of processes and specific formalisms via the software tools supports the recognition of distinct semantic domains.

Finally, identifying the comparison's inquiring process amounts to specifying the initial requirements on how the located similarities and/or differences need to be represented. Proper difference representation is central to the 3D CAD model comparison problem as it directs the interpretation of the calculated results by the inquiring process and provides efficiently for further manipulations.

5.5 Structuring a typical design reuse scenario

This section presents an example of a 3D CAD model comparison scenario structured and represented by means of the proposed PDMF4C meta-model. To better demonstrate the meta-model's representational ability, the scenario of design reuse via shape-based retrieval has been chosen for its reliability, having already been explored in prior works (Cardone, Gupta et Karnik, 2003; Iyer *et al.*, 2005). Design reuse is a specific scenario from the product information reuse application domain.

The shape-based retrieval approach illustrated here has been implemented by Siemens PLM's Geolus Search® engine (Siemens PLM Software inc., 2011a). One of the most important applications of Geolus Search® has been in purchasing, with buyers evaluating the prices for new parts by comparing them with the costs of similarly shaped existing parts (Wolfe, 2006). This example shows the pertinence of the PDMF4C meta-model in the conceptual representation of 3D CAD model comparison scenarios.

5.5.1 Design reuse via the shape-based retrieval of 3D CAD models

As described in Section 5.3.2, the generic product information reuse scenario incorporates two distinct 3D CAD model comparison problems and, thus, two distinct functions: (1) to find candidate models similar to the query model and (2) to locate the similarities and/or differences between the query model and a reduced number of candidate models.

This example focuses specifically on the first problem specifically applied to design reuse. Accordingly, shape-based retrieval constitutes the solution domain, as the problem is characterized by a 1-to-N cardinality and requires a low level of detail. The need for computational efficiency is significant, whereas the requirements regarding difference calculation and representation remain minor.

This scenario is represented via the PDMF4C meta-model in Figure 5.7. A candidate 3D CAD model (target, left) and a query 3D CAD model (reference, right) are each represented as initially unrelated **Model** instances embodying the design **Definition** instances of distinct **Product** instances. A **Comparison** association instance clearly identifies the comparison's basic function and characterizing aspects, as well as the reference and target **Model** instances.

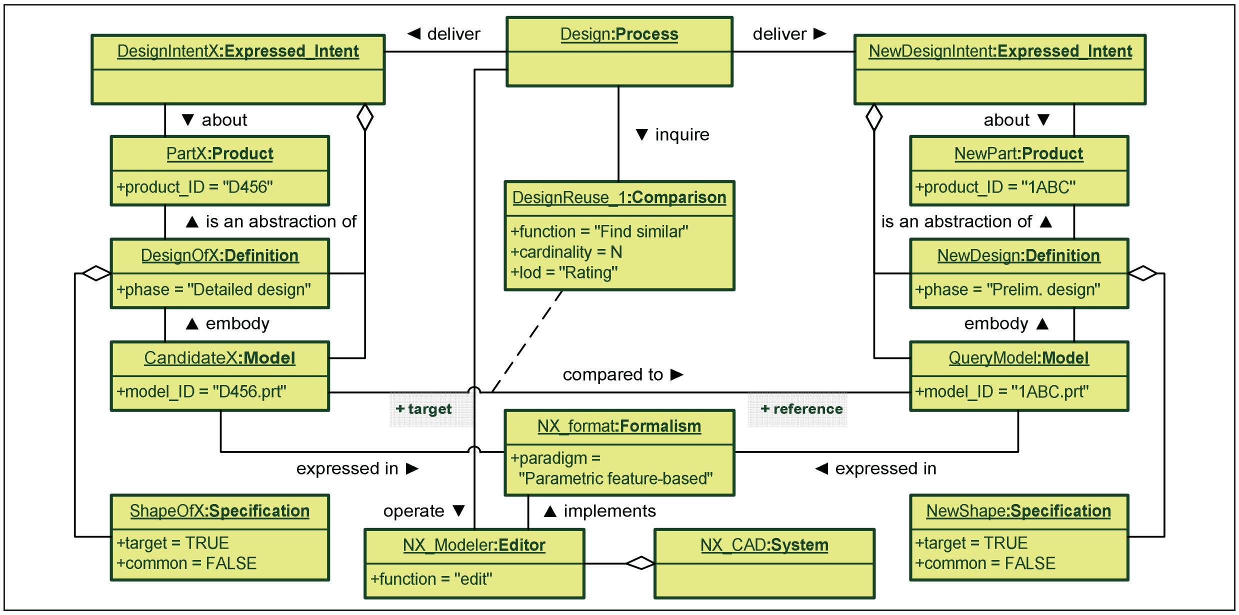


Figure 5.7 Representation of a typical shape-based retrieval scenario

In this design reuse scenario, the design process is the only one involved; therefore, the single Design:**Process** instance is associated with each of the **Expressed_Intent**, **Software_Tool** and **Comparison** instances. Shape is also identified as the feature on which the similarity of the modeled parts will be based; two **Specification** instances identifying the parts' specified shapes as the objects of the comparison are represented and associated as subsets of the **Definition** instances.

5.5.2 Geolus Search® shape-based retrieval solution

Shape-based retrieval methods resort to shape signatures to represent a models' content and thus accelerate the highly recurrent comparison process. As illustrated in Figure 5.8, indexing of candidate 3D CAD models is performed by a shape signature extractor implementing a shape signature algorithm.

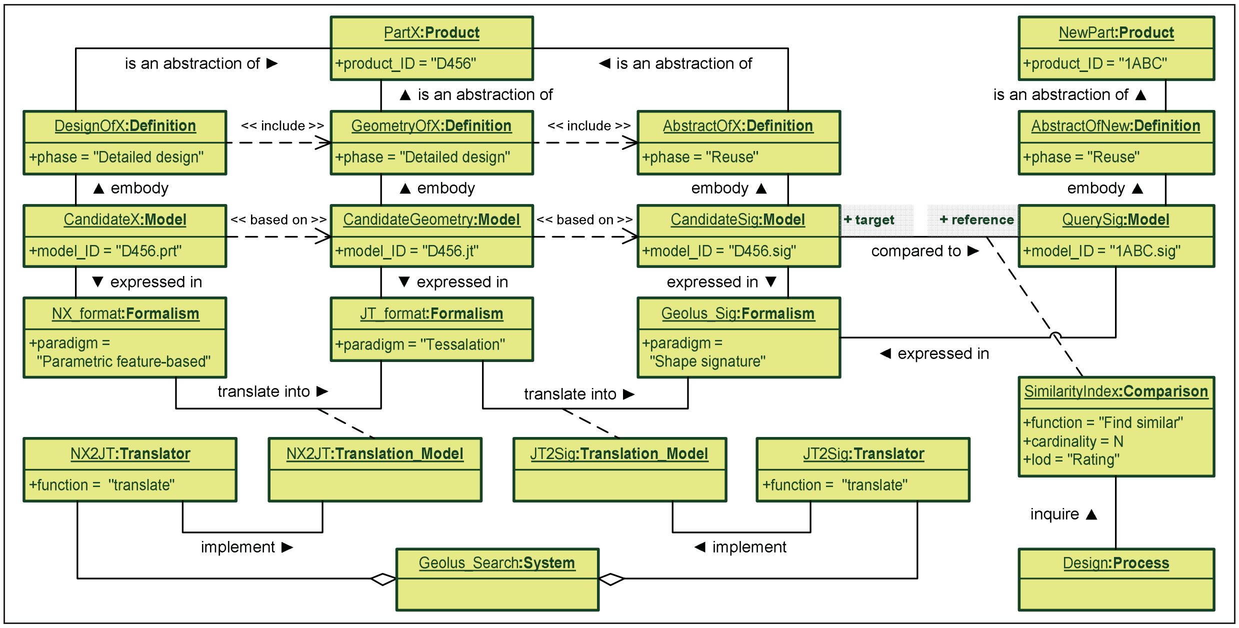


Figure 5.8 Representation of the shape-based retrieval approach of Geolus Search®

The query model's shape signature must also be extracted in order to be compared to those of the candidate models (excluded from Figure 5.8 due to space constraints). The resulting shape signatures are regarded as new **Model** instances that now embody specific yet very limited subsets of specifications from the initial design definitions embodied by the original 3D CAD models. Once abstracted, the contents of all 3D CAD models can be compared to evaluate and rank their similarity.

One particularity of the Geolus Search® solution to shape-based retrieval of 3D CAD models resides in the need for a translation of the original 3D CAD models to a tessellation-based format before the extraction of their respective shape signatures, such as Siemens PLM's proprietary JT format (Wolfe, 2006). Such pre-processing aims at ensuring the 3D CAD interoperability of the 3D CAD repository. Since a tessellation-based formalism has a limited capability to express design intent, shape signatures are only extracted from approximated models embodying reduced geometric definitions. In the current scenario where shape specifications were identified as the objects of the comparison, such particularity will conform to the initial problem representation.

However, if it had been otherwise – e.g. if specifications regarding form features were the objects of the comparison – the described solution would have been unsuitable for this design reuse scenario. Accordingly, it shows how important it is to identify and structure the defining factors of each 3D CAD model comparison scenario in order to efficiently identify the right comparison solution.

5.6 Conclusion

As described in this paper, 3D CAD model comparison contributes positively to PLM, as it promotes the resourceful use and reuse of product data at various stages of a product's lifecycle. 3D CAD models are a means for sharing knowledge for use by the different actors from the product lifecycle; thus, the tools made available to extract and process that knowledge need to assimilate specific settings and objectives. The proposed three-step approach and related PDMF4C meta-model are designed to structure and elaborate 3D CAD model comparison scenarios, providing better representation of the problem to be addressed from a case-based perspective.

The function of 3D CAD model comparison, the compared models' initial compositions and shared relationships, as well as the inquired and inquiring processes, constitute defining factors for comparison scenarios. Depending on the basic function, the results expected from a 3D CAD model comparison will range, in terms of representation, from simple pass/fail diagnoses to complex data structures. Preliminary elements of commonality between the compared models on aspects such as their respective formalisms, embodied product definitions or overall expressed intent are sure to exist and need to be formally identified.

Our current work on 3D CAD model comparison addresses the issue of difference representation as it effects the transposition of engineering change between 3D CAD models. The objective is better interpretation and manipulation by the inquiring process of the differences calculated between two 3D CAD models. Working with the PDMF4C framework has allowed us to thoroughly represent the scenario of engineering change transposition, which incorporates 3D CAD format heterogeneity. From a large-scale perspective, the proposed approach will benefit numerous actors looking to efficiently exploit 3D CAD model comparison in order to reduce product development costs and delays. A manufacturing process planner searching for similar existing parts to reuse manufacturing processes for new parts, or an engineering change committee assessing the overall impact of design changes applied to revised parts, would be just two obvious examples.

CHAPITRE 6

IDENTIFICATION OF SHAPE DIFFERENCES BETWEEN 3D CAD MODELS USING GEOMETRICS CONSTRAINTS.
PART I: DIFFERENCE MODELING

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue « *Computers in Industry* » en juillet 2014.

6.1 Abstract

In mechanical CAD, the model difference identification (MDI) problem will translates into the proper identification and interpretation, from the specific viewpoint of each specialist engaged in product design, of what differentiate either two versions of a part's geometry or two similar part geometries represented in CAD models. The corresponding process comprises five steps: (i) representing the reference and target models in a declarative form according to the difference meta-model (DMM), (ii) mapping corresponding B-Rep shape elements from both models, (iii) transposing the explicit geometric constraint (EGC) schema from the reference model to the target model, (iv) representing shape differences as per the transposed EGC schema, and (v) rendering the difference model in human-readable notation. As the first part of a two-paper contribution to CAD MDI, this paper addresses the first and fourth steps relating to difference representation by presenting a new functional 3D CAD difference modeling approach for the calculation and representation of shape differences by means of EGCs. Accordingly, 3D shape differences are expressed in terms of modified dimensions and violated geometric conditions – i.e., on a level of abstraction more comprehensive for mechanical designers.

6.2 Introduction

The development of modern products is driven by concurrent work conducted by many specialists from distributed teams generating, modifying and/or reusing the product model, making decisions and taking actions accordingly. Critical decisions and actions regarding a product's geometric evolution will depend on the proper identification, representation and assimilation by each task domain of what differentiates a new version of a product's shape from its previous versions. Accordingly, innovative methods and tools are sought out in order to enable the rapid, accurate and comprehensive identification and representation of shape differences between models. The 3D CAD model difference identification (MDI) problem is addressed here in a two-part paper. This paper is the first part and addresses the matter of difference representation, while the second part in an accompanying paper will address the matter of difference calculation.

The MDI problem distinguishes itself from 3D CAD shape similarity assessment (Cardone, Gupta et Karnik, 2003). It requires the detection of differences between 3D shape representations and the reporting of accurate details about those differences, while similarity assessment provides qualitative and abstract evaluations of how similar two shapes may be (Brière-Côté, Rivest et Maranzana, 2011). Recent investigations on current 3D shape difference calculation methods (Brière-Côté, Rivest et Maranzana, 2012a) and commercially-available MDI software technologies (Brière-Côté, Rivest et Maranzana, 2013) have revealed that proper representation of geometric differences in the form of persistent difference models is often eclipsed by visualization mechanisms prematurely displaying low-level geometric calculation results such as volumes, geometric deviations and face colorations. Even though accurate results can be provided, little relevant information about shape differences from a mechanical design viewpoint is actually provided, leaving too much for the designer to interpret in order to grasp the rationale behind a shape change. Even when available, difference models are usually transient, preventing comparison information to be exploited other than for constricted document-based reporting.

Sensitive interpretation and integration of shape differences in processes such as product design directly depend on comprehensive difference modeling. Shape differences must be represented using metrics bearing an adequate type and level of semantics. As regards to mechanical design, standard dimensioning practice (e.g. as found in ASME Y14.5 (American Society of Mechanical Engineers, 2009)) actually reveals the first level of abstraction at which mechanical designers naturally operate. In CAD modeling, this is the level of abstraction at which geometric constraints are added to a design and at which modifications are applied (Bettig et Shah, 2001). Case studies carried out in major aerospace companies (Quintana, Rivest et Pellerin, 2012) revealed the persistent use of marked-up prints to naturally capture engineering changes at early stages of the engineering change management (ECM) process. Correspondingly, shape changes are detailed by means of standard annotations related to engineering drawing elements such as the shape, dimensioning, referencing and tolerancing elements. Thus, the application of CAD geometric constraints for shape difference modeling in an ECM context constitutes a promising avenue.

This paper presents a constraint-based 3D CAD model difference modeling approach enabling the functional measure and characterization of geometric differences between two mechanical parts' explicit models. Our approach calls for the abstraction of explicit geometric constraints (EGCs) and B-Rep shape elements – geometric and topological entities – as the extension of a difference model mapping corresponding elements between the compared CAD models. It is based on a four-level architecture from functional difference modeling in model-driven software engineering (MSDE) aiming at facilitating version merging and archiving that can be beneficial to CAD. Comprehensibility and functionality of the difference model are two major aspects of CAD MDI that are addressed in this paper. Focus is on the initial representation of CAD models enabling difference calculation, addressed in an accompanying paper, and on the final representation of shape differences from the point of view of the compared parts' design intents to optimize their assimilation by mechanical designers.

Following a brief review of recent developments in 3D CAD MDI presented in the next section, we discuss in Section 6.4 the key characteristics of the MDI problem that are addressed in our work. The concept of explicit geometric constraint (EGC) and constraint-based 3D CAD model comparison are introduced in Section 6.5. Key elements and attributes of EGC representation are described in Section 6.6. We propose a difference meta-model (DMM) and a complementing CAD data meta-model in Section 6.7. Finally, in section 6.8, we discuss the practicability of deriving EGC schemas from 3D shape representations in current CAD models. Extending the work presented in this paper, new CAD model difference calculation procedures exploiting the proposed DMM are presented in an accompanying paper.

6.3 Review

Engineering change management (ECM) represents one of many application domains reported in previous work (Brière-Côté, Rivest et Maranzana, 2012a) benefiting from 3D CAD model comparison. Others applications, such as product information reuse (Jackson et Buxton, 2007; Msaaf, Maranzana et Rivest, 2007) and CAD translation validation (CAx Implementor Forum, 2008), among others, are influenced mainly by advancements in the fields of shape similarity assessment (Cardone, Gupta et Karnik, 2003) and shape-based retrieval (Iyer *et al.*, 2005; Zhang et Peng, 2009). As regard to mechanical design and shape change assessment, ECM calls for MDI solutions as they are expected to describe shape differences in relation to a product's actual form and fit specifications, i.e., the specific viewpoint on a product's definition as embodied by 3D CAD models.

6.3.1 Model difference identification

Previous work in MDSE (Kolovos *et al.*, 2009) has identified the three basic components of the MDI process, which we transpose to the comparison of 3D CAD models for the purpose of this paper.

- *Difference representation* processes the information from difference calculation to construct a difference model (Δ) amenable for subsequent analysis and manipulation, such as version merging and archiving.

- *Difference calculation* relates to algorithms establishing relationships, or *mappings*, between the compared models' elements in the difference model, and to algorithms identifying differences between those mapped elements according to specific properties and criteria.

- *Difference visualization* renders the difference model in human-readable notation (e.g., graphics, indented lists, reports, etc.) to enable designers to (quickly) grasp the rationale behind the shape differences.

Model difference representation and calculation are pivotal to any CAD MDI solution and represent the focus of our work. They are addressed separately in this paper (Part I) and in an accompanying paper (Part II), respectively. As for difference visualization, it is considered as the extension of difference representation and not as the conclusion of the MDI process. Good difference visualization promoting better interpretation of model differences will depend on the rendering of a functional and comprehensible difference model.

6.3.2 CAD model difference representation

A recent survey of MDI techniques (Brière-Côté, Rivest et Maranzana, 2012a) has led us to identify three major design-oriented approaches for CAD model differences representation. Figure 6.1 presents two shape differences – a reduced hole diameter and an added round – as represented with respect to each of the three approaches. Table 6.1 shows a sample of the surveyed MDI propositions and implementations classified with respect to both their respective difference calculation methods and representation approaches to exemplify the distinctions between the two concepts.

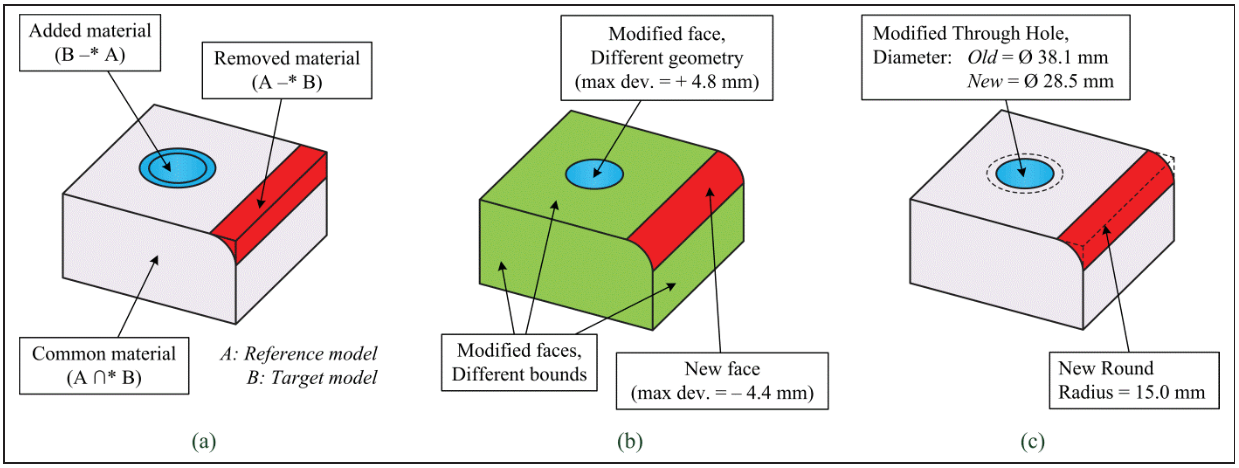


Figure 6.1 Three approaches to the design-oriented representation of a given set of shape differences: a) Delta volumes; b) Boundary-based; c) Parametric differences.

Tableau 6.1 Classification of model difference identification (MDI) solutions surveyed by Brière-Côté, Rivest and Maranzana (2012a) according to their difference calculation and representation approaches

Calculation methods	**Representation approaches**		
	Material delta regions	*Boundary-based*	*Dimensions and parameters*
Spatial occupancy difference	Francois and Cuilliere (2000) Dassault Syst. SolidWorks® (2010) Dassault Syst. CATIA® V5 (2007)	--	--
Point-to-part deviations	PTC Pro/ENGINEER® (2009)	CapVidia CompareVidia® (2010) Lattice Tech. XVL Studio® (2010)	--
Shape superposition	Dassault Syst. CATIA® V5 (2007)	Oracle AutoVue® (2010) Adobe Acrobat® 3DReviewer (2008)	--
Explicit B-Rep matching	--	Pan et al. (2011) Cuilliere et al. (2011) Msaaf et al. (2007)	Autodesk Inventor® Fusion (2010b) SpaceClaim® Engineer (2011)
Procedural representation matching	--	--	Chatelain et al. (2002) PTC Pro/ENGINEER® (2009)

Differences may be represented as material addition, material removal and common material regions between two part geometries via localized 3D delta volumes (Figure 6.1(a)). This first approach is often implemented jointly with spatial occupancy difference calculation algorithms, and used, for example, in CAD/FEA integration applications. Francois and Cuilliere (2000) use an octree representation of a modified part to identify regions of an original FEA mesh model requiring local remeshing. PTC Pro/ENGINEER® (Parametric Technology Corporation, 2009) implements part comparison by analyzing the directions of point-to-part deviations and identifying delta material regions accordingly. Delta volumes will provide metrics such as volume evaluations, local centroids and maximum deviations.

A second approach uses a model's boundaries to represent differences. Related difference models commonly classify faces, edges and vertices from B-Rep models according to some specified difference criteria (Figure 6.1(b)). Other boundary-based difference models, such as the one implemented by Lattice Technology XVL Studio® (Lattice Technology Co., 2010), aggregate deviating facets between tessellated shapes to form distinct difference areas. Boundary-based differences can be calculated by using geometric algorithms such as point-to-part deviation calculation (e.g. CapVidia NV (2010)) or mapping algorithms such as topological entity matching (e.g. Cuillière *et al.* (2011), Pan *et al.* (2011)).

6.3.3 Parametric and dimensional differences

Representing shape differences on higher levels of abstraction in the form of modified dimensions, parameters and form features, which are more familiar to mechanical engineers, constitutes the third difference representation approach. Comparing two CAD models' procedural representations, i.e., their construction tree, will provide differences represented in terms of procedural – new, modified, reordered or suppressed modeling operations – and parametric modifications (Figure 6.1(c)). Chatelain, Maranzana and St-Martin (2002) match similar primitives between two constructive solid geometry (CSG) trees to distinguish unique and parametrically-modified primitives from equivalent ones. A feature comparison function by PTC Pro/ENGINEER® (Parametric Technology Corporation, 2009) matches modeling operations and parameters between model versions by means of static identifiers to detect and measure differences.

Yet, difference models from procedural CAD comparison will not necessarily correlate with an actual part's geometric specifications. Differences between modeling operation sequences do not systematically translate into differences at the resulting shape level. Also, the high variability of procedural solid representation, both structurally and semantically, prevents the comparison of two 3D CAD models unless they originate from the same seed modeling sequence. In a product lifecycle-wide context like ECM, only comparing successive versions of single-view homogeneously formatted models is too restrictive.

Inversely, comparing explicit or evaluated 3D CAD models amounts to calculating the differences between two geometries whose full details are immediately available without the need for any form of pre-calculation (International Organisation for Standardization, 2005). Corresponding difference calculation methods operate exclusively at the geometric and topological levels. CAD interoperability is therefore less of a concern, since geometric and topological entities are implemented throughout a variety of explicit CAD formats. However, there is a limit to what low-level geometric models can provide in terms of meaningful input in a design process. Key design semantics, i.e., the actual part geometry specifications, are often evacuated during explicit geometric comparison.

Remarkably, recent CAD MDI implementations by SpaceClaim® Engineer (SpaceClaim Corporation, 2011) and Autodesk Inventor® Fusion (Autodesk inc., 2010b) manage to yield high-level dimensional or parametric difference models via the comparison of explicit geometries. SpaceClaim® Engineer's (SpaceClaim Corporation, 2011) 3D markup functionality allows CAD users to re-input design data after the comparison via special part dimensions which simultaneously display old and new values from two versions of a part model. This implementation exploits matching algorithms based on static identifiers – also known as persistent naming (Agbodan *et al.*, 2003; Kripac, 1997; Wu *et al.*, 2001) – to match faces between versions.

Autodesk Inventor® Fusion's (Autodesk inc., 2010b) Change Manager module propagates direct-modeling modifications applied to a 3D CAD model in a history-free environment to a previous version of the same model represented in a history-based environment. This comparison algorithm relates low-level geometric differences to the model's original construction tree and computes their representation as modified or new parameterized modeling operations. However, complex modifications often develop into semantically-deprived modeling operations, such as face copies, applied directly to the original shape. A static identity-based matching algorithm is also used to match faces between the explicit and procedural versions of the models.

Since changes made to CAD models are mostly parametric nowadays, expressing changes as purely geometrical like delta volumes or boundary-based differences has become obsolete. In the authors' opinion, the challenge with MDI has evolved into identifying dimensional and parametric changes and transposing them to different domains (i.e. design and engineering). This two-paper contribution aims at overcoming the shortcomings of parametric difference models obtained via the comparison of explicit geometries as observed in SpaceClaim® Engineer's (SpaceClaim Corporation, 2011) and Autodesk Inventor® Fusion's (Autodesk inc., 2010b), among others.

6.4 Model difference identification problem

We emphasize here the shortcomings of the current 3D CAD MDI solutions implementing explicit and/or procedural CAD comparison alike. To do so, we describe two actual MDI scenarios. The first scenario involves the assessment of shape modifications between two models, supporting ECM in a distributed and concurrent design environment. The second scenario involves the comparison of similar part models previously identified via shaped-based retrieval with respect to a reference model, supporting the selection of a suitable candidate for part design reuse.

6.4.1 The ECM scenario

A mechanical designer supplies an original part geometry specified by a *reference* 3D CAD model to an external machining process planning (MPP) specialist whose task is to develop a process plan for the modeled part according to its form and fit specifications among other design and manufacturing information. Taking the machining processes' limitations into account, the MPP specialist ultimately requests a change to a part's original geometry to optimize it from a machinability perspective. To do so, he submits a new 3D CAD model, denoted as the *target* model, presenting a revised geometry to the designer. The designer's task is to assess to what extent the original design intent from the reference model would be affected by the revised geometry from the target model, and whether to accept, reject or amend the shape change request.

6.4.2 The part design reuse scenario

A shape-based retrieval technique, such as one reported by Iyer *et al.* (2005), Zhang and Peng (2009), or Cardone, Gupta and Karnik (2003), or implemented by Siemens PLM Software inc. (2011a) or 3DSemantix inc. (2011), was applied to a CAD database comprising tens of thousands of heterogeneously formatted models. It was used to retrieve a set of existing part models that are found similar to a given reference model embodying desired geometric specifications. At this point, differences between each retrieved similar models and the reference model are only assessed qualitatively. Further analysis of the differences is thus required to select the optimal candidate for design reuse and to instruct its adaptation to desired specifications. To reinforce the selection and the part design reuse initiative, the ensuing comparison of each similar part models with the reference model ought to provide an intuitive and functional representation of model differences relating to the specific application of part design reuse.

6.4.3 Key characteristics

The quality of an efficient 3D CAD MDI solution will come from its ability to proficiently support the mechanical designer in assessing the impact of a shape differences on the design intent. We highlight the following characteristics from these particular scenarios that must be accounted for in developing the adequate MDI solution:

- Compared CAD models are stand-alone, i.e., no persistent formalized associations are available between them, reflecting the reality of distributed and concurrent design environments, and thus preventing resorting to persistent naming to match model elements;

- Compared CAD models are potentially in different CAD formats, raising CAD interoperability concerns, but it can be expected that accurate boundary representations (B-Rep) of both geometries are available, enabling the geometric federation of CAD models (Iraqi Houssaini, Kleiner et Roucoules, 2012) through the abstraction of topological entities; and
- A formal representation of the original design intent is available in the reference 3D CAD model, whether in the form of geometric constraints, dimensions and/or form features, but is absent or ignored from the target or similar models, given that it represents a part's geometry from a different viewpoint or from another application.

As both scenarios presents the same key characteristics, the remainder of this paper focuses on the ECM scenario. Key characteristics of different MDI problems and other 3D CAD model comparison scenarios can be identified and formalized via the framework proposed by Brière-Côté, Rivest and Maranzana (2011).

6.5 Model comparison based on explicit geometric constraints

Exploiting available geometric constraints, among other CAD model elements such as attributes, annotations, GD&T, parameters, design rules and features, constitutes a practical approach to elevate the level of abstraction of the difference model at the designer's level. From both operational and descriptive standpoints, geometric constraints represent the first level and most common type of engineering semantics.

Geometric constraints specify the relationships between shape elements – geometrical and/or topological – and between shape elements and design parameters (i.e., logical and dimensional constraints, respectively), as pictured in Figure 6.2. Explicit geometric constraints (EGCs) encompass constraints that are fully defined and formalized as individual model elements in a 3D CAD model, as opposed to constraints implicit to feature definitions, as exposed by Kim *et al.* (2008). In Figure 6.2, algebraic constraints, or "design rules", denotes relationships between design parameters. They are considered non-geometric and out of the scope of this work.

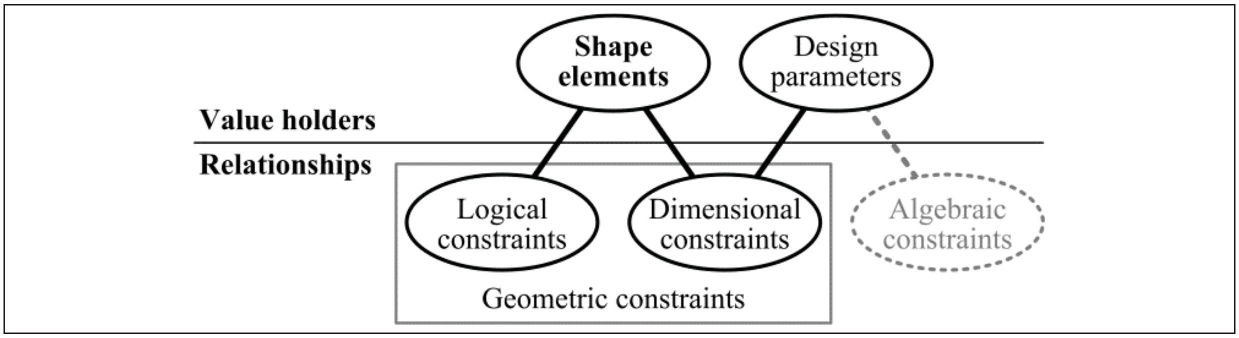


Figure 6.2 Relationships between shape elements, design parameters and geometric constraints (Inspired by Bettig et Shah (2001))

Following the modification of design parameters, constraints should be maintained; otherwise, geometric constraints, which embody the modeled part's form specifications, would be violated. Equally, given two versions of a modeled part, we are interested in identifying which design parameters were modified, and if geometric constraints are maintained in both versions.

We refer to the *reference* model as the model embodying the original version of a part. It should include a set of geometric constraints, making up the reference constraint schema. The proposed difference model aims at determining to what extent the new geometry from the target model respects the reference schema. Design parameters are to be re-evaluated accordingly. It is assumed that the target model is without a viable EGC schema: thus, the reference schema must be transposed to the new shape by means of the shape element mappings established during shape difference calculation.

The concept is exposed in Figure 6.3 with an example. In the reference model (left), a "parallel-distance" geometric constraint (e.g., the `pgc_with_dimension` entity from ISO STEP Part 108 (International Organisation for Standardization, 2005)) is specified between two side planar faces *A* and *B* of a slot, and leads to the definition of design parameter *d* corresponding to the slot's width. When compared to the target model (right), face mappings are first established between faces *A* and *A'*, and between faces *B* and *B'*. Since both related faces are mapped, a similar parallel-distance constraint is transposed on the target shape. Expressing the difference between the models then comes down to validating the parallelism of faces *A'* and *B'* and measuring a new value for target design parameter *d'*. Procedures carrying out EGC transposition as described here are detailed in an accompanying paper.

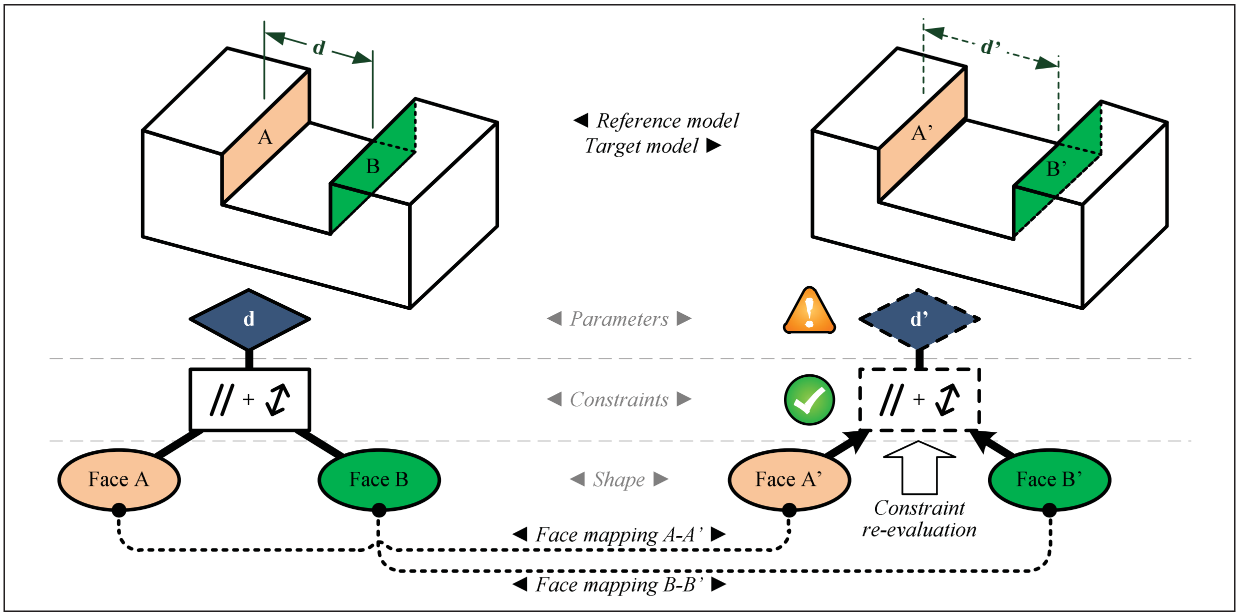


Figure 6.3 Transposition of a geometric constraint and related design parameter on a target model via face mappings

We propose to relate shape differences to the original EGC schema according to the following nomenclature:

- *Maintained constraints* are transposed EGCs whose condition between mapped shape elements are upheld and whose associated parameters' values are constant;
- *Modified constraints* are transposed dimensional EGCs whose design parameters exhibit different values between reference and target models;
- *Violated constraints* are transposed logical EGCs whose original geometric condition is not upheld by corresponding shape elements in the target model; and

- *Unique constraints* are original EGCs that could not be transposed in the target model as they either relate to a shape element unique to the reference model or to geometrically incompatible mapped shape elements in the target model.

Geometric incompatibility occurs when mapped shape elements like faces and edges are described by different geometric equations, preventing an EGC to be transposed in the target model. For example, the cylindrical face of an original blended edge may be mapped to the new planar face of a chamfer applied to the same edge in the target model as a consequence of the equivalence of adjacent faces. Accordingly, a radius constraint on the original cylindrical face cannot be transposed on the new planar face.

6.6 Breakdown of an explicit geometric constraint schema

This section presents the key characteristics of an EGC schema relating the shape elements of a B-Rep model.

6.6.1 Physical vs auxiliary shape elements

In the original EGC schema, some constraints relate both *physical* and *auxiliary* shape elements. Physical shape elements (PSE) are B-Rep model elements making up the physical boundary of the modeled part. They are found in both reference and target models and are the only subjects of shape element matching.

Conversely, auxiliary shape elements (ASEs) comprise, for example, reference geometry created in 3D space and construction geometry used in 2D sketches. They do not make up the physical boundary of the part, but still participate in its definition together with EGC instances within the schema. Accordingly, ASEs are considered part of the original EGC schema and, thus, limited to the reference model.

6.6.2 The arity of EGCs

The number of shape elements related by a single constraint may vary from one to many. However, each EGC is characterized by a basic *arity*; i.e., the minimal number of shape elements that can be related via the geometric relationship it embodies. For example, a single dimensional constraint may specify a single radius value for many circular or cylindrical elements simultaneously. Still, a radial constraint is considered as a *unary* constraint – of arity 1 – as it fundamentally relates to separable shape elements. Constraints such as parallelism, coincidence and distance are considered as *binary* constraints as they fundamentally relate pairs of shape elements. *Ternary* constraints comprise the symmetry constraint, among others.

6.6.3 Elementary vs composite EGCs

From a practical perspective, difference models must be able to reference individually the EGCs and their variations currently used by most 3D CAD systems. Conversely, the functional aspect of difference models, on which our EGC transposition algorithm will depend, calls for a comprehensive, yet constant and consistent set of EGC types to be processed. For example, EGC types defined in the `explicit_geometric_constraint_schema` of ISO STEP Part 108 (International Organisation for Standardization, 2005) form a practical set of geometric constraints, while the theoretical set derived by Bettig and Shah (2001) is considered to be consistent.

A satisfying trade-off between practical and consistent sets is achieved by distinguishing *composite* and *elementary* EGCs. Composite EGCs define multiple geometric relations between shape elements. A composite EGC then decomposes itself into elementary EGCs, which individually represent a single geometric relation among a minimal number of shape elements, corresponding to the EGC's arity. For example, the "parallel-distance" geometric constraint of Figure 6.3 is a composite EGC that can be broken down into a logical 'parallel' and a dimensional 'distance' elementary EGC.

EGC types able to collate many shape element tuples via a single instance, such as the unary radial dimensional constraint, will be referred to as either composite or elementary EGCs. A radial constraint instance relating multiple circular elements simultaneously is referred to as a composite constraint. The multiple radial constraint instances making up such composite constraint are, however, referred to as elementary constraints. Such distinction eventually enables each shape element's radial dimension to be verified individually during a difference calculation.

6.6.4 Directed vs Undirected EGCs

As elaborated in ISO STEP Part 108 (International Organisation for Standardization, 2005), EGCs composing a schema may either be *directed* or *undirected* and, as a whole, characterize the overall *directionality* of the schema, i.e., how the part geometry is to evolve in case of constraint and parameter modifications. Directed constraints are defined with respect to one or more *reference elements*, which are not changed by the action of these constraints if they are invoked during the modification of the model. A directed constraint only allows the opposing *constrained elements* to be modified in such a way that their specified relationships to the reference elements are maintained. Modification of one or more reference elements (from outside the context of the directed constraint concerned) will lead to compatible changes in all the constrained elements it controls.

Inversely, an *undirected* constraint has no reference element, only requiring the constrained condition to hold amongst all pairs of members of a set of constrained elements. All unary constraints are undirected as they fundamentally constrain single shape elements.

6.7 Difference modeling

Functional CAD difference modeling is achieved in this paper by following the example of software model version management from the field of model-driven software engineering (MDSE). While dealing with issues like model patterns (features) and interoperability that are familiar to CAD, software model comparison solutions focus on the functionality of difference models to aptly enable automated version management mechanisms such as model reconstruction and the detection and manipulation of conflicts between parallel versions. Attractive concepts such as the model-based difference representation approach by Cicchetti, Di Ruscio and Pierantonio (2007) and the weaving models by Del Fabro, Bézivin and Valduriez (2006) were therefore transposed to the field of 3D CAD model comparison to be included in our new method.

This section presents the information framework for constraint-based CAD model comparison in the form of a difference meta-model (DMM). The definition of this DMM is part of a general difference modeling approach. It also includes the definition of a CAD data meta-model designed to abstract, rather than convert, most CAD models, regardless to their inherent formats. This CAD data meta-model is independent from the DMM and, thus, can be defined to suit the MDI process requirements.

6.7.1 Explicit CAD data representation

We initiate the representation of differences on the primary association of corresponding shape elements between models. To do so, the distinction between constrained shape elements and constraints must be preserved. An explicit or declarative approach to CAD data representation, allowing model elements to be referenced and manipulated individually, is required. Accordingly, shape elements are to be declared first, and then the constraints between these elements (e.g. as in ISO STEP Part 108 (International Organisation for Standardization, 2005)).

To guarantee an explicit and homogeneous representation of the compared CAD models, we adopt the model-based difference modeling approach described by Cicchetti, Di Ruscio and Pierantonio (2007). By exploiting the four-level model-driven architecture of the Meta-Object Facility (MOF) (Object Management Group, 2011), these researchers in the field of MDSE manage to circumvent any restriction over the data meta-model by abstracting, rather than converting, the compared models to conform a given meta-model. Their approach is *meta-model-independent*, i.e., any meta-model can be applied, MDSE or CAD alike.

Transposed in the domain of CAD, such an approach enables the definition of a difference meta-model (DMM) able to represent differences among CAD models that can be made to conform to a specific explicit CAD data meta-model, regardless of their initial format, as illustrated in Figure 6.4. This four-level model-driven architecture has been used in similar computer-aided engineering (CAE) applications such as the one described by Iraqi Houssaini, Kleiner and Roucoules (2012).

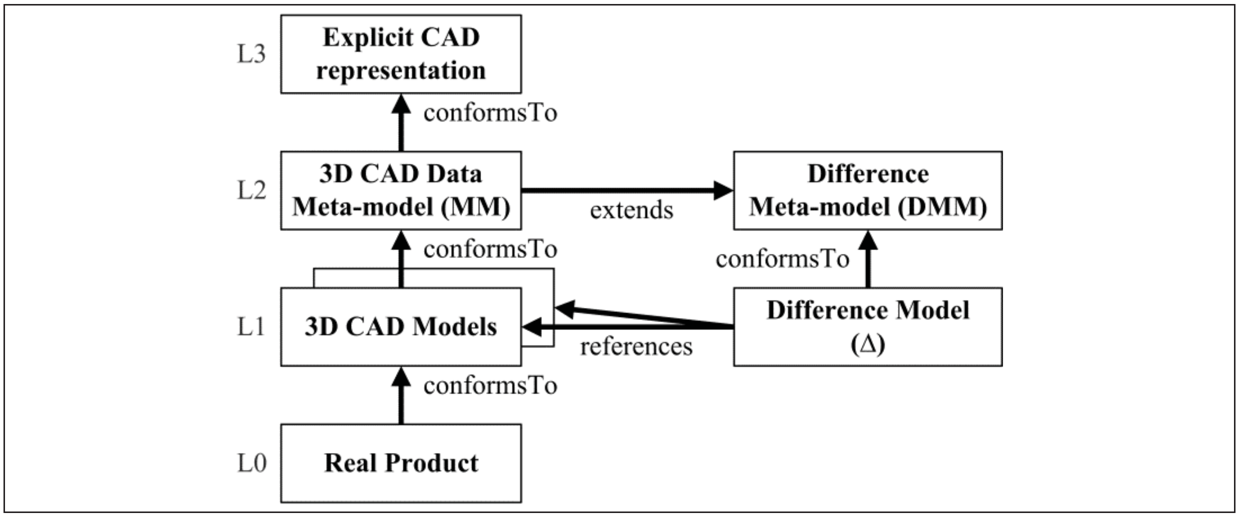


Figure 6.4 Overall structure of the model difference representation approach

The underlying concept of this model-driven architecture is abstraction, which combines generalizing conceptualization and reductive simplification to go from a set of objects, even the models themselves, to concepts that subsume them. For example, CAD models (L1) are embodied abstractions of the similar yet heterogeneous parts and products (L0) to be manufactured according the limited set of specifications that the models convey. Representation of these abstractions depend on the CAD format used, which means that the same set of specifications can be represented heterogeneously. All manufactured parts and products will then be examined with regards to the CAD models to check if they meet the specifications.

A second level of abstraction allows us to circumvent the heterogeneous representations of CAD models (L1) by isolating the conceptual CAD objects such as geometry, topological entities and constraints and, thus, to compare the models according to their constituting elements and not on how these elements are represented. According to which constituting elements the CAD models are to be compared is specified by the CAD data meta-model (L2) which extends the DMM. Difference models are instances of the DMM that relate two compared CAD models. In our application, the CAD data meta-model must in turn respect the overall requirements of explicit CAD representation (individual reference and manipulation of elements, declaration of elements from low- to high-level of semantics, etc.).

6.7.2 CAD data meta-model

Constraint-based comparison justifies the specification of a declarative CAD data meta-model that minimally encompasses the explicit representation of shape boundaries, geometric constraints, design parameters and all relationships between such elements. Through this approach, comparison does not occur between the CAD models themselves, but rather between two instances of the CAD data meta-model within the difference model, abstracting the reference and the target models. Model abstraction allows the manipulation of images of CAD model elements, like geometrical and topological elements and associated EGC, while maintaining their respective implementations.

The proposed CAD data meta-model is illustrated by the UML diagram of Figure 6.5. References to the original CAD models and there constitutive elements that are abstracted in conformity to the proposed meta-model are preserved via the '**ref_CADobj**' attribute. It is designed to record original elements' static identifiers from their respective implementations. Preserving such multi-leveled association between the difference model and the compared models adds to the functionality of the DMM by providing a retroactive read-only access to the original CAD data.

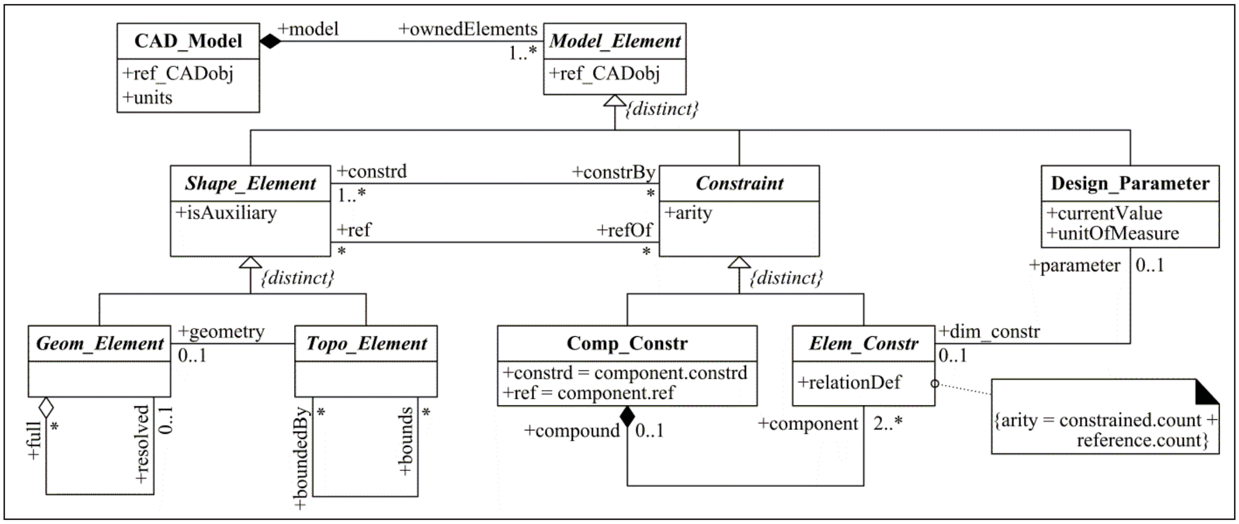


Figure 6.5 UML representation of the proposed CAD data meta-model

6.7.2.1 Shape Elements

Geometrical and topological entities, along with elementary relationships involved in B-Rep, are represented by the ***Shape_Element*** abstract class. Its ***Geom_Element*** subclass extracts information about surface equations (planar, cylindrical, conical, spherical and toroidal), curve equations (circle, ellipse, hyperbola and parabola) and point coordinates.

The ***Topo_Element*** subclass extracts information about basic topological entities such as faces, edges and vertices, but also about bounded curves and surfaces, such as line segments, circular arcs and surface patches to reference bounded and bounding geometric elements separately. It depends on ***Geom_Element*** instances and captures information about how they relate to each other via the boundedBy/bounds association. Once abstracted in a difference model, the validity of the topological structures is not mandatory: non-manifold shapes can be referenced.

The key distinction between *physical* and *auxiliary* shape elements is represented in the CAD data meta-model by means of the '**isAuxiliary**' property. Implicit relationships, like the relationship between a cylinder and its axis, are extracted via a full/resolved aggregation association.

6.7.2.2 Constraint Elements

In the CAD data meta-model, ***Constraint*** instances relate ***Shape_Element*** instances to each other and/or to ***Design_Parameter*** instances, and abstract the geometric conditions that must be maintained. The specific properties of EGCs described in Section 6.6 – arity, directionality and composition – are represented. Referencing composite EGCs from a CAD model in a difference model enables the expression of shape differences with respect to the shape's actual representation, while elementary geometric constraints are easier to manipulate during difference calculation and subsequent analysis.

Subclasses of ***Elem_Constr*** abstract elementary EGCs such as "parallelism", "incidence" and "radius", among others. The extent of these subclasses must allow the abstraction of a consistent, yet practical, set of EGC types, like the ones derived by Bettig and Shah (2001) or declared within the `explicit_geometric_constraint_schema` of ISO STEP Part 108 (International Organisation for Standardization, 2005). Integrational constraints dependent upon topological information, such as curve length and curve midpoint, can be included as EGC types, along with common pre-defined algebraic constraints, such as radius or length equality, since that are common to many CAD formats.

6.7.2.3 Design parameters

The declaration of individual design parameters in the CAD models via the **Design_Parameter** class completes the CAD data meta-model. Relationships to corresponding dimensional EGCs are captured, enabling design parameter transposition as a result of dimensional EGC transposition. **Design_Parameter** instances are characterized by a current value ('**currentValue**') and a unit of measure ('**unitOfMeasure**').

6.7.3 Difference meta-model (DMM)

Mappings between corresponding abstracted sub-elements of the reference and target models constitute the fundamental relationships to be captured and maintained by the DMM. Accordingly, we have adopted the *core weaving meta-model* proposed by Del Fabro, Bézivin and Valduriez (2006) as a suitable baseline for the definition of our DMM. This generic weaving meta-model was designed to support four common requirements for link management. Likewise, we apply these requirements to CAD difference modeling:

1. The basic notion of links or mappings between compared model elements is expressed clearly;
2. Different mapping types are supported, each providing the semantic information on how the elements are related and how they differ, or not, from each other;
3. Mappings with different arities (binary, ternary, etc.) may be defined; for example, an original cylindrical face from the reference model may correspond to two hemi-cylindrical faces in the target model, thus requiring a single mapping with three endpoints; and
4. The DMM has an identification mechanism to uniquely identify model elements, i.e., mappings' endpoints do not contain concrete model elements, but a pointer that enables them to be accessed in the containing models.

The following paragraphs and the UML diagram of Figure 6.6 detail the different elements of the proposed DMM.

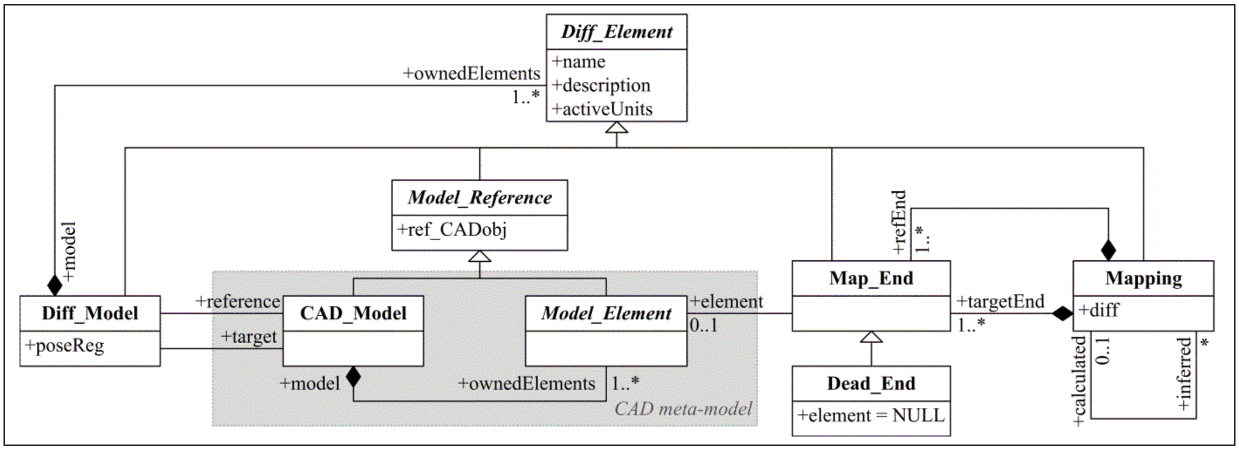


Figure 6.6 UML representation of the difference meta-model (DMM)

The **Diff_Model** class represents the root element that embodies the difference model. It must integrate the two instances of the CAD data meta-model abstrating the reference and the target models and identifies them accordingly in the difference model to capture the directionality of the comparison. Consequently, it also integrates the various mapping between their respective elements. Information about the pose registration, i.e. the difference between the models' positions and orientations in space, is stored at this level.

The **Mapping** class fulfils the first and second requirements. Embodied linking semantics involve the distinction between mapping endpoints belonging to either the reference or the target models to maintain the comparison directionality at a lower level of granularity. Also, calculated/inferred relationships are maintained between mappings inferred from one another for subsequent analysis (e.g. mapped faces leading to their respective geometry's mapping). To capture the varying levels of equivalency/difference between each type of model element (as described in Section 6.5), the **Mapping** class incorporates the '**diff**' attribute.

The **Map_End** handles the third requirement. Every mapping endpoint represents a mapped model element, which makes it possible to create N-ary mappings. The **Dead_End** subclass of **Map_End** enables the expression of model elements unique to either compared model.

The ***Model_Reference*** class satisfies the fourth requirement by maintaining a reference to corresponding CAD objects from the specific implementations of the compared CAD models. Static identifiers can be used as references, but not exclusively. The **CAD_Model** and ***Model_Element*** classes are those from the CAD data meta-model and extend the DMM, preserve the meta-model independence of the DMM. The **CAD_Model** class references and represents an entire CAD model in the DMM. As the root element of the CAD data meta-model, it aggregates ***Model_Element*** instances representing and referencing the CAD models' elements in conformity to the CAD data meta-model.

6.8 Discussion

Constraint-based difference modeling as proposed in this paper relies on the practical assumption that adequate EGC and parameter data can be obtained or derived from 3D CAD representations. Given that parametric feature-based modeling constitutes common practice today, we discuss the present or near-future feasibility of exploiting EGC in the representation of 3D shape differences.

6.8.1 Implicit geometric constraints

Explicit geometric constraints and design parameters constitute fundamental components of 3D CAD procedural modeling; e.g., as applied in 2D sketches. In other utilizations, geometric constraints and related parameters are often implicit to feature definitions, as exposed by Kim *et al.* (2008). Extracting implicit geometric constraints towards the generation of an explicit difference model, like in the proposed approach, may become problematic as it opens the way for variable feature decompositions and interpretations. Moreover, when rendered explicit, constraints from early modeling operations may relate to intermediate shape elements that do not make up the resulting part geometry, as exemplified in Figure 6.7. Differences represented in relation to intermediate shape elements will not correlate with actual part shape specifications.

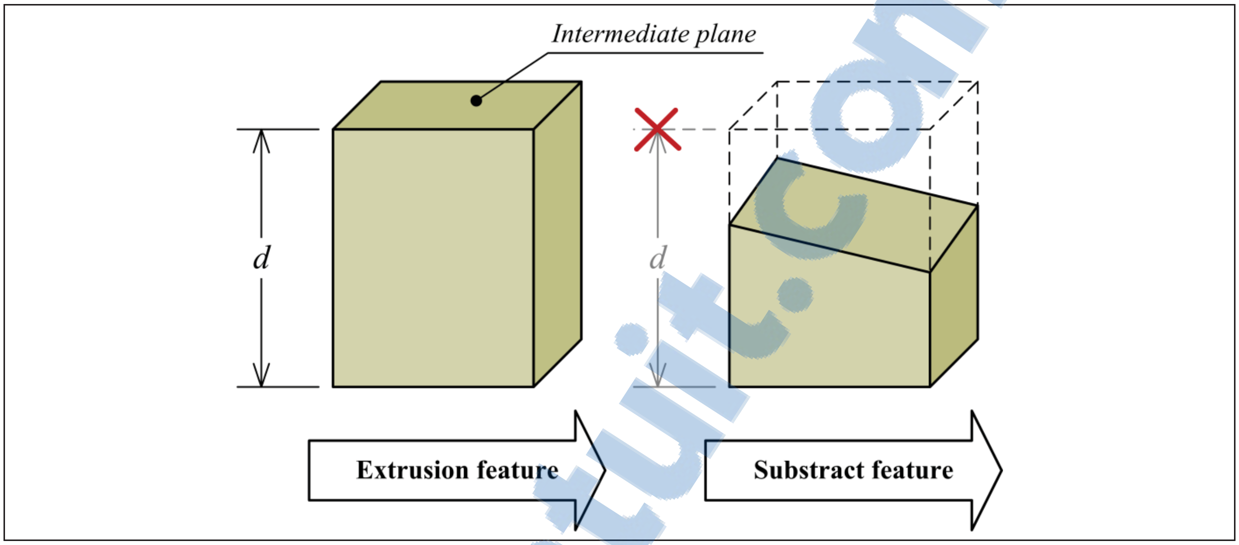


Figure 6.7 Implicit geometric constraint relating to an intermediate plane

Algorithms designed to “flatten” a procedural model and make it explicit, while preserving EGCs, will contribute to produce explicit geometric constraint data on 3D geometry and make it available for difference modeling. Mechanisms for projecting sketch constraints from 2D to 3D geometry, along the lines of the “sketch consumption and dimension migration” process implemented by Siemens PLM Solid Edge® Synchronous Technology (Siemens PLM Software inc., 2009b), could be considered. CAD algorithms parsing constraint data from procedural representations to automatically add 2D dimensions on projected views in engineering drawings are another example.

6.8.2 Direct 3D CAD modeling

The emergence of explicit direct modeling, or “history-free” modeling, into common CAD practice will obviously contribute to make explicit 3D CAD models with EGCs more available. Current direct CAD modeling systems focus on countering the rigidity of parametric feature-based modeling in early design phases and in collaborative environments, stressing on leaving as much freedom as possible to designers. From that perspective, EGCs, already handled by those systems, and, eventually, explicit form features, will play a major role in part modeling in the near future, notably to regulate the evolution of part shape specifications as early as the conceptual design phase.

6.8.3 Product manufacturing information

This proposition focuses on the specific CAD data objects that are EGC. A wider scope would ultimately bring up 3D model-based definition (MDB) and related product manufacturing information (PMI) as a potential source of explicit geometric relationship data. Expressed in formalisms such as ASME Y14.5 (American Society of Mechanical Engineers, 2009) and ASME Y14.41 (American Society of Mechanical Engineers, 2012), PMI is inherently explicit and relates to the shape on the same level of abstraction as geometric constraints. The proposed comparison method could therefore be easily applied to the representation of shape difference with respect to originally specified PMI, as long as such information is represented with full semantics; i.e., as typified objects relating to the explicit shape representation, as opposed to polyline groups simply located in the modeling space.

6.9 Conclusion

This paper tackles the problem of providing intuitive and functional information about differences between two 3D CAD models compared in a mechanical design context. We have shown that the available design-oriented difference representation approaches do not relate shape differences to the design intent, or that they depend on difference calculation techniques not flexible enough to be widely applied in a distributed design environment. Thus, we have illustrated how explicit geometric constraints (EGCs) could be exploited to elevate the level of abstraction of shape difference representation to a level where mechanical designers naturally operate.

Inspired by MDSE, the proposed functional constraint-based CAD difference modeling approach emphasizes on the explicit representation and abstraction of geometric elements, topological elements, geometric constraints and design parameters, and provides a new framework for low-level CAD model elements matching between compared CAD models for refined difference characterization. It enables the individual reference to and manipulation of heterogeneously formatted CAD model elements, such as explicit shape elements and EGCs, during and after the comparison. Accordingly, 3D shape differences can now be expressed in terms of modified dimensions and violated geometric conditions instead of by means of delta regions or classifications of boundary elements.

Further efforts are focused on elaborating difference calculation procedures such as the aforementioned EGC transposition to complement the constrained-based DMM presented in this first paper. The calculation component of our MDI solution and the validation of the overall approach – DMM and algorithms combined – via a concrete example are the subjects of a subsequent paper. EGC transposition will aim at reproducing and adapting the original EGC schema from the reference CAD model with respect to the evolved shape from the target model, generating a difference model enabling precise and comprehensive identification of CAD model differences in shape change assessment scenarios.

CHAPITRE 7

IDENTIFICATION OF SHAPE DIFFERENCES BETWEEN 3D CAD MODELS USING GEOMETRICS CONSTRAINTS.
PART II: EXPLICIT GEOMETRIC CONSTRAINT SCHEMA TRANSPOSITION

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue « Computers in Industry » en juillet 2014.

7.1 Abstract

This paper presents an approach for model difference identification (MDI) between two 3D CAD models representing similar mechanical components. The calculation and representation of shape differences exploits the semantics of geometric constraints in CAD models to elevate difference modeling to a level of abstraction more comprehensive for mechanical designers. The process comprises five steps: (i) representing the reference and target models in a declarative form according to the difference meta-model (DMM), (ii) mapping corresponding B-Rep shape elements from both models, (iii) transposing the explicit geometric constraint (EGC) schema from the reference model to the target model, (iv) representing shape differences as per the transposed EGC schema, and (v) rendering the difference model in human-readable notation. This paper addresses the second and third steps, while the first and fourth steps are addressed in an accompanying paper. Locating and representing shape differences between two CAD models in terms of modified dimensions and violated geometric conditions has applications in engineering change management, e.g., assessing to what extent an original design intent expressed in terms of geometric constraints in a reference model is affected by the revised geometry from a target model and taking the right actions accordingly. The full constraint-based comparison approach is exemplified through the comparison of two similar CAD sketches.

7.2 Introduction

Mechanical design projects commonly engage multiple specialists in distributed environments whose tasks are to concurrently contribute to the evolving product definition, make decisions and take new actions accordingly. As regards to part geometry, these decisions often depend on the proper identification and interpretation, from the specific viewpoint of each specialist, of what differentiates the old and new versions of that geometry as represented in 3D CAD models. Representation of the differences must meet levels of abstraction more comprehensive for mechanical designers than basic regions of material addition/deletion or low-level face coloration to grasp the rationale behind a shape change.

This challenge is referred to as a model difference identification problem (MDI). It is addressed in this two-part paper and can be concisely described by the following statement:

- *Given a target 3D CAD model presenting the revised geometry of a part design, we intend to locate and represent shape differences to assess to what extent the original design intent from the reference 3D CAD model expressed in terms of geometric constraints is affected by the revised geometry from the target model.*

The solution we are proposing to this problem can be divided into the following sub-tasks:

1. Adapt the heterogeneous reference and target CAD models for comparison, operated by abstracting them into a declarative representation conforming to a predefined CAD data meta-model.
2. Establish mappings between corresponding geometric and topological elements of the compared models now represented in a difference model, first enabling low-level comparison.
3. Transpose original shape specifications embodied by the explicit geometric constraint (EGC) schema from the reference model to the target model, enabling high-level comparison.

4. Represent low- and high-level differences between reference and target models within a difference model conforming to a predefined difference meta-model (DMM).
5. Render the difference model in human-readable notation to enable designers to grasp the rationale behind the shape differences.

The first and fourth sub-tasks relating to model difference representation are addressed in an accompanying paper which presents a new functional CAD model difference meta-model (DMM) designed for the assessment of shape differences applied to mechanical parts. Derived difference models capture and convey representations of shape differences between two 3D CAD models from the perspective of their geometric constraints, i.e., on a level of abstraction at which mechanical designers naturally operate. Functional CAD difference modeling is addressed by deriving relevant modeling requirements from functional difference modeling in model-driven software engineering (MSDE) which already supports version merging and archiving.

The second and third sub-tasks relate to model difference calculation and constitute the scope of this second paper. First, the reference and target CAD models must be made to relate via geometric and topological elements – the B-Rep data structure – as it allows heterogeneous CAD models to be federated and compared. Accordingly, we present key requirements that a B-Rep matching algorithm must meet to be applied in constraint-based CAD model comparison. Then, geometric constraints from the reference model – i.e., common CAD model elements bearing elementary engineering semantics – are reproduced using topological mappings and adapted to fit the target shape, highlighting differences from a comprehensible perspective in the process. As for the fifth sub-task, it relates to model difference visualization. It is considered reliant on model difference representation and is therefore left out of scope.

Next section reviews related work on B-Rep element matching algorithms and on difference calculation techniques focused on identifying CAD model differences with respect to a part's dimensions and/or parametric features. Section 7.4 summarizes our constraint-based model comparison method and lays out the difference calculation framework defined by the DMM. Requirements for a shape element matching algorithm, carrying out the second sub-task described above, are stated in Section 7.5. Then, we describe the key procedures for geometric constraint-based difference calculation via EGC schema transposition – i.e., third sub-task – by detailing in Sections 7.6 and 7.7, respectively, the procedure for auxiliary shape element (ASE) transposition and the procedure for EGC transposition. Finally, application of the proposed method is illustrated and validated in Section 7.8 through a simple example involving the comparison of two 2D CAD sketches.

7.3 Related work

An extensive review of 3D CAD model comparison applications, calculation methods and software tools was presented in a previous paper by Brière-Côté, Rivest and Maranzana (2012a). The following section presents related work specific to this paper's scope, i.e., topological entity matching algorithms and B-Rep-based difference calculation towards the design-oriented representation of shape differences via dimensions and design parameters.

7.3.1 Topological entity matching

Data structure matching and comparison distinguishes itself from geometric comparison since it fundamentally processes representation data structures, such as B-Rep, instead of computed geometric results, such as point-to-part deviations (CapVidia NV, 2010). Any CAD representation scheme can be the subject to this type of matching algorithm. For instance, Chatelain, Maranzana and St-Martin (2002) match primitive solids from similar CSG models according to their type, transformation matrix, dimensional parameters and position index in their respective CSG trees. Likewise, PTC's Pro/ENGINEER® (Parametric Technology Corporation, 2009) compares procedural models by matching parametric modeling features via proprietary static identifiers.

We identified four matching approaches for B-Rep data structures following the example of model matching approaches used in model-driven software engineering (MDSE) (Kolovos *et* *al.*, 2009). *Static-identity matching* operates with persistent and unique identifiers assigned to B-Rep elements upon creation by the originating CAD system, commonly referred to as *persistent naming* in literature (Agbodan *et al.*, 2003; Kripac, 1997; Wu *et al.*, 2001). Its application is restricted to model versions originating from the same seed model and created with the same originating CAD system, as implemented by SpaceClaim® Engineer (SpaceClaim Corporation, 2011), for example.

Signature-based matching work with dynamic signatures assembled or computed from the values of each B-Rep element's attributes and properties, most commonly geometric. The *3DComparator* tool described by Msaaf, Maranzana and Rivest (2007) extracts geometric signatures from faces and matches them to distinguish equivalent faces from different faces between two B-Rep models.

Similarity-based matching treats B-rep data structures as typed attributed graphs and matches elements based on the aggregated similarity of their attributes and geometric properties. Algorithms require relative weights to be allocated to each analysed property to regulate them in similarity evaluation and must often undergo an empirical trial and error fine-tuning process. Cuillière *et al.* (2011) presents a similarity-based algorithm that computes the metric tensor, inertia tensor and barycenter of each face's and each edge's respective set of control points from two NURBS-based B-Rep models and gradually matches them by verifying the equivalency of their computed properties in a predefined order. CapVidia CompareVidia® (CapVidia NV, 2010) implements an algorithm producing N-to-M face mappings by extracting sample points on faces and matching faces based on relative point-to-face correlation.

Syntax-specific matching incorporates the semantics of B-Rep data structures, reducing the search space and providing more accurate results. For example, the algorithm proposed by Pan *et al.* (2011) compares two edges if the faces they are adjacent to are already known to match. Similarly, PTC's CoCreate® Modeling (CoCreate Software GmbH, 2008; CoCreate Software GmbH et Gutierrez, 3 juin 2010) implements a B-Rep difference calculation algorithm in which vertex, edge and face mappings are computed progressively using geometric attributes and lower-order mappings – i.e. edges are matched using vertex mappings and faces are matched using edge mappings.

7.3.2 B-Rep-based parametric difference calculation

Recent CAD MDI implementations by SpaceClaim® Engineer (SpaceClaim Corporation, 2011) and Autodesk Inventor® Fusion (Autodesk inc., 2010b) produce high-level difference models. SpaceClaim® Engineer's (SpaceClaim Corporation, 2011) *3D markup* functionality allows CAD users to create part dimensions simultaneously displaying old and new values from two compared versions of a part model. These transient dimensions must be purposely restored at each comparison to produce a difference model relating to the original part specifications.

Autodesk Inventor® Fusion's (Autodesk inc., 2010b) *Change Manager* module propagates direct-modeling modifications applied to a 3D CAD model in a history-free environment to a previous version of the same model represented in a history-based environment. The implied comparison algorithm manages to relate low-level geometric differences to the model's original construction history, and to compute their representation as modified or new parameterized modeling operations. The process of restoring a modeling procedure from explicit shape differences, however, remains complicated. Complex modifications often develop into low-level modeling operations – i.e., bearing no design semantics – such as face copies applied directly to the original resulting shape.

Both of the above-mentioned implementations present the shortcoming of exploiting static identity-based matching algorithms to match B-Rep faces between compared versions of a model. Thus, the scope of these particular implementations is limited to the comparison of different versions of the same originating CAD model file.

Research on multiple-view feature modeling (Bronsvoort, Bidarra et Nyirenda, 2006) goes beyond the calculation and representation of shape differences by automating (after a change) the update of many feature models embodying specific views of the product model. Hoffmann and Joan-Arinyo (2000; 1998) developed an architecture for a product master model that federates CAD systems with downstream application processes for different feature views that are part of the design process. When a shape change occurs, part geometry across federated procedural feature models is kept consistent to a master *net shape* through the maintenance of *geometric certificates* – persistent shape-level mappings between B-Rep entities – and the distribution of *change protocols* – mapping-based representations of differences.

Updating a feature model, which equates to representing shape differences with respect to the editing client's viewpoint, is performed according to the distributed *change protocols*. Among others, the "adjustment by constraint reconciliation" updating procedure measures the dimensional variations on the new net shape according to the edited feature model's specific constraint schema. However, these procedures are subject to many restrictions as they involve automatically editing feature-based procedural models with explicit representations of geometric differences. Such an operation presents a degree of complexity similar to that of feature recognition.

7.4 Approach and related concepts

We recall the three key characteristics from the ECM and part design reuse (Brière-Côté, Rivest et Maranzana, 2012b) scenarios described in the accompanying paper that embodies the MDI problem addressed in this paper and, thus prevents the state-of-the-art solutions described above from being applied :

- No formalized associations persist between compared CAD models, preventing resorting to static identity-based matching or persistent naming to match model elements;
- Compared CAD models are heterogeneously formatted, but the geometric federation of CAD models (Iraqi Houssaini, Kleiner et Roucoules, 2012) via B-Rep is possible; and

- Geometric constraints, dimensions and/or form features are available and effective only in the reference CAD model.

Our constraint-based model comparison approach emulates the concept of constraint schema reconciliation between original and new models, introduced by Hoffmann and Joan-Arinyo (2000). The revised concept is defined as *EGC schema transposition* and elevates the level of abstraction of shape difference representation to that of geometric constraints. As illustrated in Figure 7.1, difference calculation is achieved by extending B-Rep matching and comparison algorithms with the transposition of geometric constraints relating shape elements from the reference model to corresponding shape elements from the target model via the mappings.

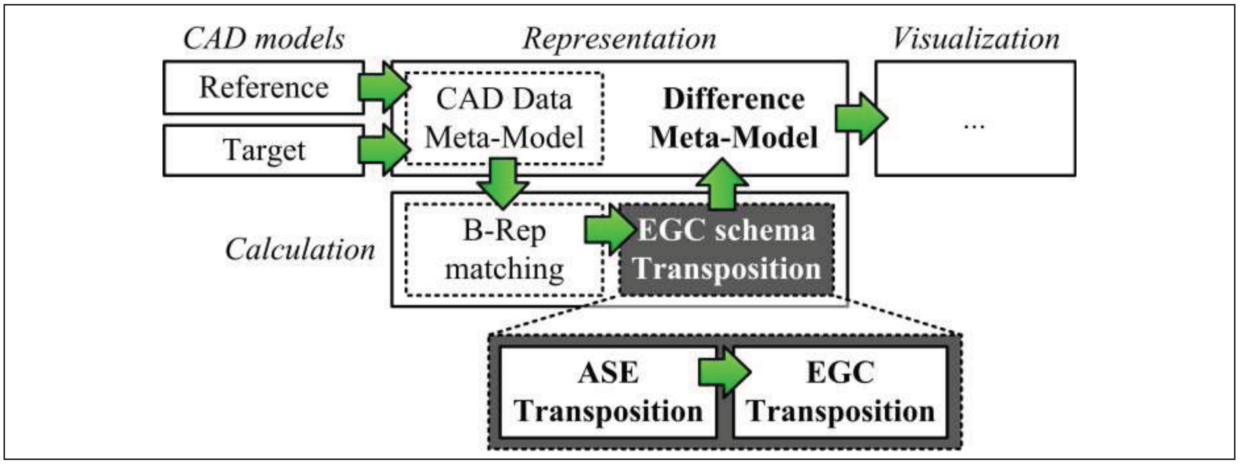


Figure 7.1 EGC schema transposition within the overall framework of the constraint-based CAD model comparison approach

Instead of exploiting persistent mappings in the form of geometric certificates, which are not available in the MDI problem we address, a different matching algorithm is used to produce mappings, on request, between elements of stand-alone B-Rep models. These shape-level mappings are developed between explicit abstractions of the compared models, rather than between the models themselves. It enables the composition of functional stand-alone difference models, independent of the compared CAD models' formats.

The concept of CAD model difference calculation via EGC schema transposition is exposed in Figure 7.2 with an example. In the reference model (left), a "parallel-distance" geometric constraint (e.g. the `pgc_with_dimension` entity from ISO STEP Part 108 (International Organisation for Standardization, 2005)) is specified between two side planar faces *A* and *B* of a slot, and leads to the definition of design parameter *d* corresponding to the slot's width. When compared to the target model (right), face mappings are first established between faces *A* and *A'*, and between faces *B* and *B'*. Since both related faces are mapped, a similar parallel-distance constraint is transposed on the target shape. Expressing the difference between the models then comes down to validating the parallelism of faces *A'* and *B'* and measuring a new value for target design parameter *d'*.

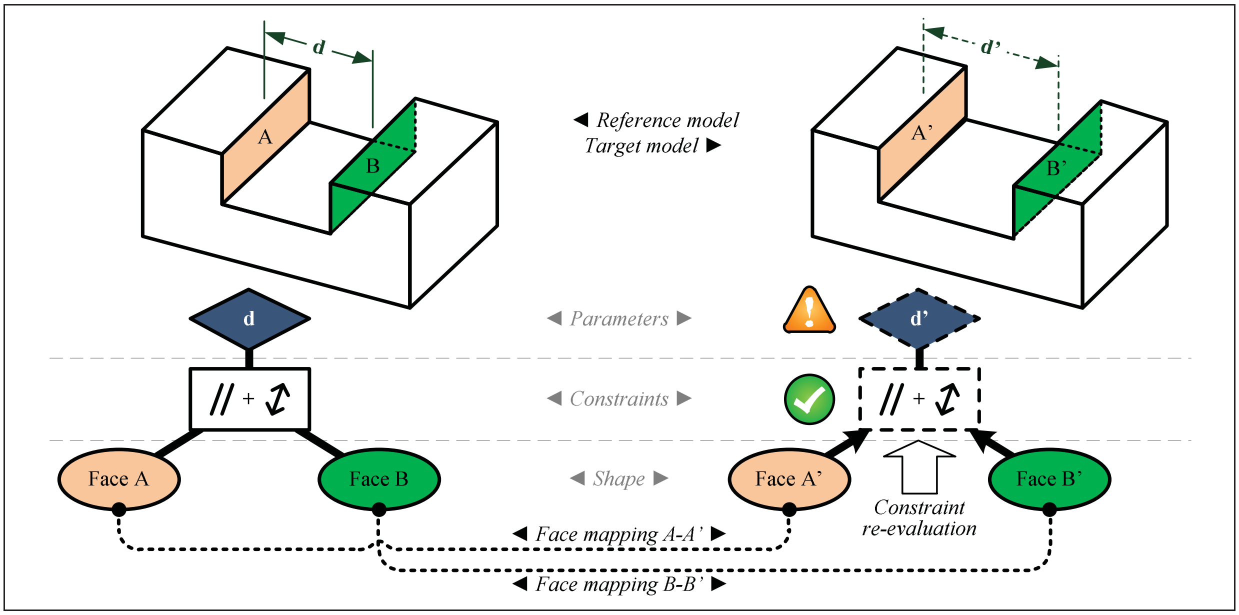


Figure 7.2 Transposition of a geometric constraint and related design parameter on a target model via face mappings

The following paragraphs recap the concept of EGC exposed in the accompanying paper as it is central to both difference calculation and difference representation in the proposed comparison method. The explicit representation of EGCs in a difference model is by the CAD data meta-model which is reported via the UML diagram of Figure 7.3. Difference models generated by the proposed comparison method conform to the DMM, also defined in the accompanying paper, which is depicted via the UML diagram of Figure 7.4.

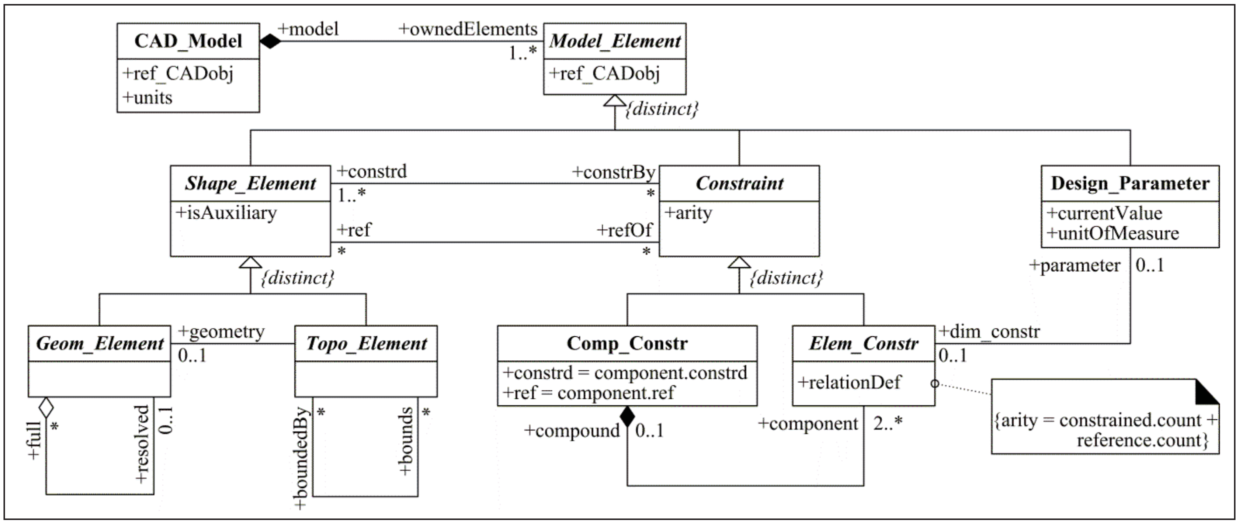


Figure 7.3 UML representation of the proposed CAD data meta-model

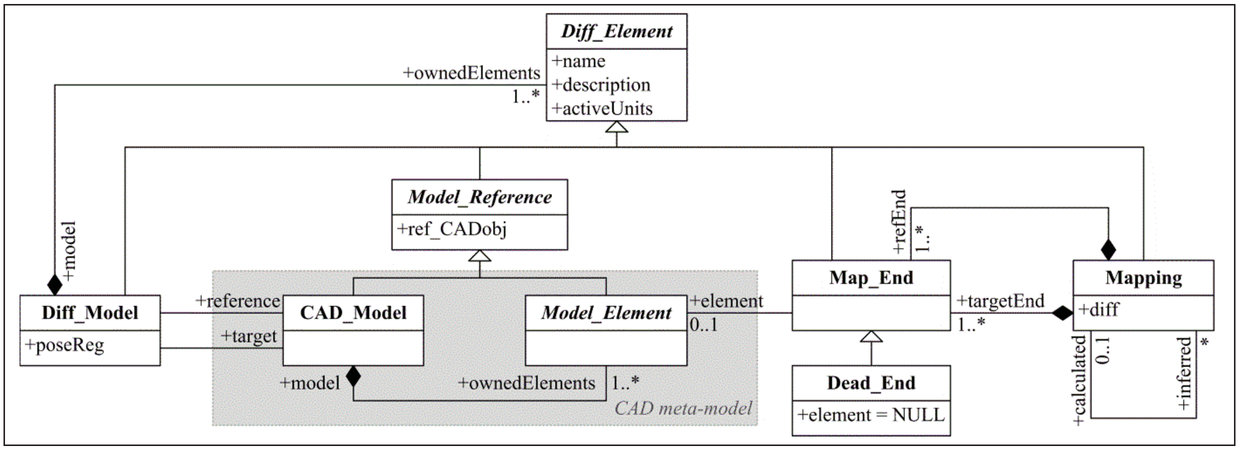


Figure 7.4 UML representation of the difference meta-model (DMM)

7.4.1 Explicit geometric constraints

Geometric constraints specify the relationships between shape elements and between shape elements and design parameters (i.e. logical and dimensional constraints, respectively). Explicit geometric constraints (EGCs) encompass constraints that are fully defined and formalized as individual model elements in a CAD model. Following the modification of design parameters associated with some given dimensional constraints, all other geometric constraints should be maintained. If not, initial shape specifications embodied by the EGC schema are violated.

An EGC schema is composed of EGCs relating to part geometry and of *auxiliary shape elements* (ASEs), such as reference or construction geometry, that do not make up the physical boundary of the part – as opposed to *physical shape elements* (PSEs) – but still contribute in its definition. For the scope of this paper, an EGC schema is assumed to be *sufficient* and *concise*, i.e., the constrained shape is without any degree of freedom and no degree of freedom is concurrently removed by two or more distinct constraints, respectively.

7.4.2 Properties and attributes of EGCs

Each EGC is characterized by a basic *arity*; i.e., the minimal number of shape elements that can be related via the geometric relationship it embodies. Unary constraints relate to a single shape element (e.g. radius constraint), binary constraints relate two shape elements (e.g. distance constraint) and ternary relate three shape elements (e.g. symmetry constraint).

An *elementary* EGC represents a single geometric relation among a minimal number of shape elements, corresponding to its arity. Conversely, a *composite* EGC is made up of many elementary EGCs. It may define multiple geometric relations between shape elements; e.g., the "parallel-distance" geometric constraint of Figure 7.2 can be broken down into a logical 'parallel' and a dimensional 'distance' elementary EGCs. Also, it may collate many shape element tuples via a single instance; e.g. a radial constraint relating multiple circular elements simultaneously to a single parameter representing a common radius value. In the late example, a composite radial constraint will be broken down into multiple elementary constraints to enable each shape element's radial dimension to be verified individually during a difference calculation. Consequently, radial variation within the initial group of circular elements will denote a violation of the initial composite radial constraint.

Directed constraints are defined with respect to one or more *reference elements*, which are not changed by the action of these constraints if they are invoked during the modification of the model. A directed constraint only allows the opposing *constrained elements* to be modified in such a way that their specified relationships to the reference elements are maintained. Inversely, an *undirected* constraint has no reference element, only requiring the constrained condition to hold amongst all pairs of members of a set of constrained elements.

7.5 B-Rep shape element matching

Although we believe that improvements can be still be made on both aspects of calculation and representation precision (Brière-Côté, Rivest et Maranzana, 2013), we approach the problem of developing a new CAD model difference calculation method by considering the B-Rep matching problem as being resolved. Existing shape difference calculation methods are capable of providing adequate topological mappings between explicit geometric models and, thus, implementing the second sub-task of the proposed comparison method. It is not within the scope of this work to further B-Rep difference calculation methods. Instead, we put forward three requirements that must be met by the matching algorithm for it to be applied in our constraint-based model comparison method:

- It must operate on detached, standalone models; therefore, we exclude any static identity-based matching algorithms (Brière-Côté, Rivest et Maranzana, 2012a);
- It must operate on precise (or “exact”) boundary representations (B-Rep) of 3D shapes common to many modern CAD formats, as opposed to tessellated shapes, and produce accurate face, edge and vertex mappings; and
- It must include pose registration (Yang, Lin et Zhang, 2007), executed either manually or automatically, and provide the corresponding results.

As an example, the B-Rep difference calculation algorithm described by CoCreate Software (CoCreate Software GmbH et Gutierrez, 3 juin 2010) and implemented in PTC CoCreate® Modeling PE (CoCreate Software GmbH, 2008) would constitute a suitable option. Its syntax-specific matching algorithm recursively produces vertex, edge and face mappings. Mappings at the geometric level are inferred from topological mappings.

B-Rep shape elements mappings are captured in the difference model which conforms to the DMM of Figure 7.4 and which will ultimately include ASE, EGC and design parameter mappings deriving from them.

7.6 Auxiliary shape element transposition

Transposing an original EGC schema to a target model inevitably involves transposing both EGCs and ASEs from the reference model. Original ASEs must be reproduced and re-evaluated in relation to the new shape beforehand, to allow all original EGCs relating physical and auxiliary shape elements alike to be transposed in a single operation.

7.6.1 Defining vs dependant EGCs

Within an EGC schema, ASEs are regarded as shape elements used exclusively for the definition of other shape elements. It is thus assumed that only directed EGCs relate ASEs to other shape elements, physical and auxiliary alike. Accordingly, distinction can be made between directed EGCs *defining* a given ASE – i.e., constraints involving the ASE as a constrained element – and directed EGCs *dependent* upon the ASE in the definition of their constrained elements. Conversely, there is no restriction regarding the directionality of EGCs relating only to PSEs.

7.6.2 Solving for an intermediate variable

By assuming that all EGCs relating to ASEs are directed constraints, we enable their expression in the form of explicit functions, with the possible distinction of independent and dependent variables for each relationship. Correspondingly, we propose to address the transposition of original ASEs in the target model similarly to the resolution of composed functions to find the values of intermediate variables.

To illustrate our approach, we consider the simple case of an ASE, denoted as a, of given geometric type A from the reference model that is related to original shape element x of given geometric type X via EGC f, and to original shape element y of geometric type Y via active EGC g. By expressing EGC f and g as functions f: $X \rightarrow A$ and g: $A \rightarrow Y$, respectively, their composition leads to the following simple expression:

$$g \circ f = \{(x, y) \in X \times Y | \exists a \in A: (x, a) \in f \wedge (a, y) \in g\} \tag{7.1}$$

To be transposed in the target model, a new ASE $a' \in A$ will be calculated according to new shape elements $x' \in X$ or $y' \in Y$ to which transposed EGC f and g will now relate. Consequently, depending on the known equivalent or differing status of each new shape element in the target model, the evaluation of the new ASE requires that one of the functions f or $(g)^{-1}$ be evaluated with new input variables. As a result, one of the two transposed EGCs is purposely maintained, or *enforced*, in the target model.

7.6.3 Enforceable relationships

The selection of specific EGCs to be enforced for the evaluation of a transposed ASE must result in the evaluation of a single shape element; i.e., each transposed ASE must be *well-constrained* with respect to the target shape by the set of enforced EGCs. Considering the differences that prevail between the original and new shapes, enforcing a specific set of EGCs with respect to the target shape may either be insufficient to evaluate a single transposed ASE, corresponding to an *under-constrained* ASE, or include conflicting EGCs preventing the transposition, corresponding to an *over-constrained* ASE.

Despite the fact that EGCs relating ASEs in the original constraint schema are to be exclusively directed, undirected relationships involving original ASEs may remain, such as an auxiliary point representing the center of a circle. These relationships are not embodied by EGCs, but are implicit to the specification of some geometric and topological elements. In all, we distinguish three groups of enforceable relationships through which each original ASE relates to original shape elements and, thus, through which it may be transposed in the target model:

1. *Implicit relationships* comprise all the relationships between an ASE and other shape elements at the geometric and topological levels. Similar relationships between PSEs are normally formalized in B-Rep data structures. For example, implicit relationships relate conic geometries (circles, cylinders, spheres, etc.) to their respective resolved geometric elements (centers, axis, etc.) when these are declared as separate elements in CAD models. They also relate bounded geometries (line segments, arcs, etc.) to their corresponding bounding elements (endpoints, etc.). Since they are implicit, these relationships are considered undirected.
2. *Defining constraints* relate the ASE to other shape elements as a constrained element. Function f in Eq. (7.1) represents a defining constraint.
3. *Dependent constraints* relate the ASE to other shape elements as a reference element. Function g in Eq. (7.1) represents a dependent constraint.

Two or more ASEs may relate to each other via the relationships described above, which amounts to solving iteratively for multiple intermediate variables in the problem expressed by Eq. (7.1). At a given iteration, each group of relationships identified for a specific ASE may relate to either target-end shape elements or other ASEs not yet transposed in the target model.

Target-end shape elements (TSEs) refer to shape elements from the target model, physical or auxiliary, at the target end of an established mapping. Accordingly, when transposed to the target model and mapped to its original counterpart at the end of a given iteration, an ASE becomes a TSE and increases the number of enforceable relationships for the evaluation of the ASEs still to be transposed in the subsequent iterations. If a mapping has no TSE (i.e., the original element is unique to the reference model), or if a geometric incompatibility arises, the corresponding relationships are removed from the three relationship groups.

7.6.4 Rules for selecting enforced relationships

During a single iteration, enforcing specific relationships with respect to the target shape to transpose an ASE in the target model must abide by the following prioritized selection rules:

- *Rule 1*: Since they are intrinsic to geometric and topological representations, implicit relationships relating an ASE with a TSE are considered inviolable, and thus are systematically enforced. Conflict in the evaluation of the transposed ASE via multiple implicit relationships leads to the dismissal of the transposed ASE.

- *Rule 2*: Logical constraints are enforced ahead of dimensional constraints to further the representation of differences in terms of modified design parameters, which we consider more intuitive for designers. Dimensional constraints may be enforced to obtain a *well-constrained* ASE in cases where logical constraints alone are either insufficient or conflicting.

- *Rule 3*: A transposed ASE must first be evaluated with respect to the target shape via enforced constraints as either a reference or a constrained element. Correspondingly, enforced constraints must ideally come from either the defining or the dependent constraint groups. Simultaneous enforcement of defining and dependent constraints is allowed in cases where logical constraints from the opposite group are enforced over dimensional constraints to follow the second rule, or where an under- or over-constrained ASE remains to be resolved.

In a sufficient and concise EGC schema, the number of defining constraints for each ASE will be adequate but minimal. The number of dependent constraints is assumed to be greater than the number of defining constraints in a majority of cases. As regards to the third rule, evaluating an ASE as a reference element presents a higher probability of achievement and is thus to be attempted first when both constraint groups are available at a given iteration.

Similar to evaluating the value of a in Eq. (7.1), the algorithms to evaluate transposed ASEs with respect to the target shape and enforced EGCs are not specified in this work. Given the original position, orientation and size of a transposed ASE, we assume that current CAD or constraint-solving technologies can be applied to find satisfying solutions once a sufficient set of enforced relationships is specified. Still, due to shape differences between original and target shapes, conflicts may persist when selecting such sets automatically. In those cases, we ultimately turn to user intervention for resolution. Persistent under-constrained ASEs due to insufficient enforced relationships or persistent over-constrained ASEs due to conflicting relationships lead to transposition failure.

7.6.5 Procedure for ASE transposition

As a preliminary procedure to EGC transposition, ASE transposition, depicted via the flow chart of Figure 7.5, aims to reproduce ASEs from the original EGC schema and transpose them with respect to the target shape in the target model. Determining the sequence in which each ASE is transposed and specifying a well-constrained set of enforced relationships constitute the two critical sub-phases of this procedure.

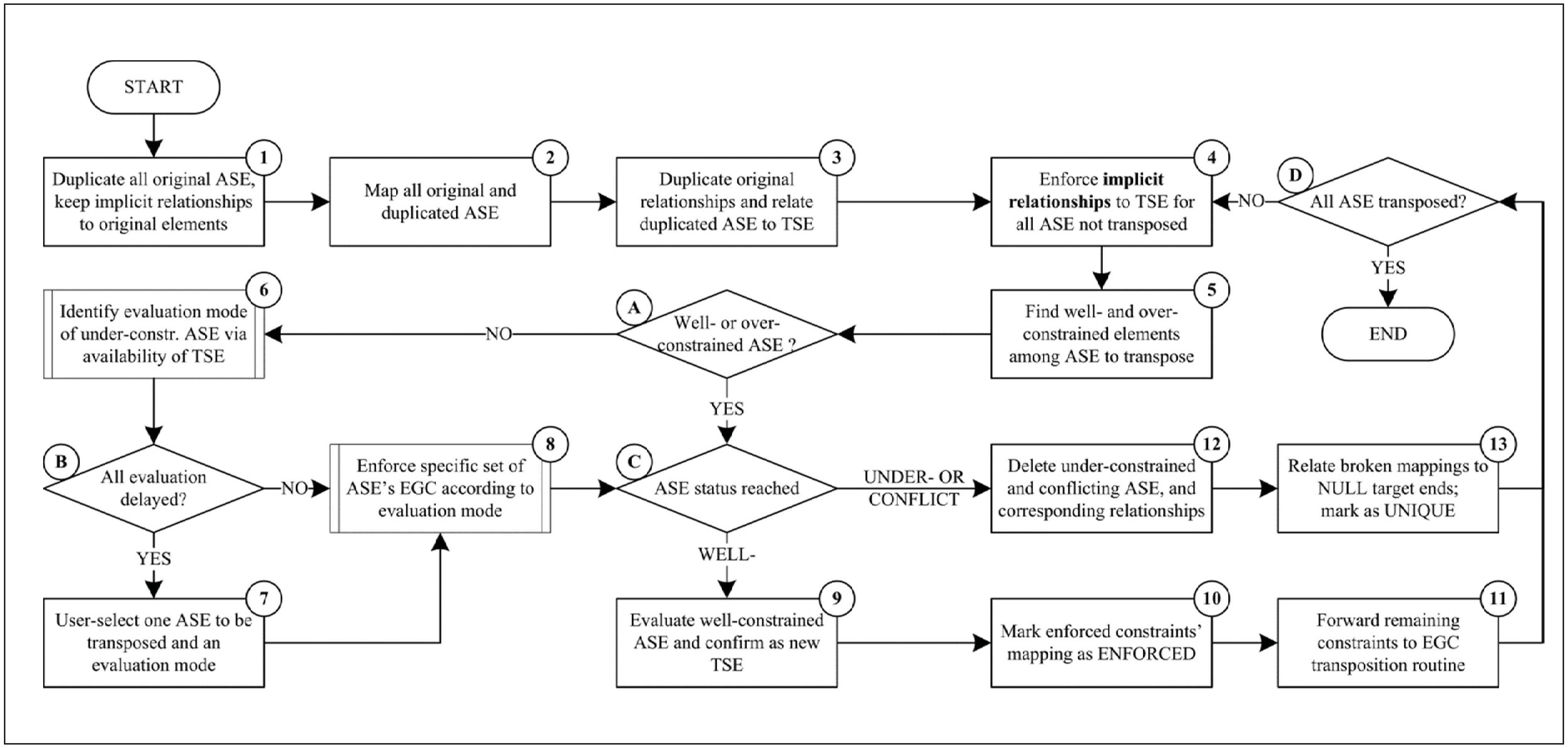


Figure 7.5 Flow chart representing the procedure for ASE transposition

7.6.5.1 Duplicating ASE and corresponding relationships

The first phase of the procedure consists of duplicating all the ASE instances and the related EGC instances from the reference model's image in the difference model (step 7.5-1; i.e., labelled step #1 from flow chart of Figure 7.5) and transposing them to the opposing target model's image (step 7.5-3). Temporary mappings between original and duplicated elements are created prior to transposition (step 7.5-2), pending their validation or deletion at the outcome of the procedure. The transposition of enforceable relationships follows a procedure similar to the one described in following Section 7.7 to locate TSEs and verify geometric compatibility. Geometric conditions and parameter values are not verified at this early stage; duplicated ASEs are not yet transposed with respect to the target shape.

7.6.5.2 Enforcing implicit relationships

The iterative portion of the ASE transposition procedure starts, as per selection Rule 1, with the enforcement of implicit relationships for all the ASEs to be transposed (step 7.5-4). If, as a result of enforcing implicit relationships, individual ASEs become either well- or over-constrained (step 7.5-5), these particular ASEs are forwarded to the last phase of the procedure where they are sorted accordingly. Depending on each ASE's status (decision 7.5-A), well-constrained ASEs are evaluated and confirmed as new target-end elements (steps 7.5-9 thru -11), and over-constrained ASEs are deleted with their corresponding relationships due to conflicts (steps 7.5-12 and -13). Enforcing implicit relationships is repeated as long as well-constrained ASEs are being transposed or over-constrained ASEs are deleted, or until all the ASEs have been resolved (decision 7.5-D).

7.6.5.3 Finding ASEs for transposition via EGC enforcement

If no or insufficient implicit relationships are available to transpose ASEs with respect to the new shape at the beginning of an iteration (decision 7.5-A), EGCs (both defining and dependent) must therefore be included in the set of enforced relationships. As per selection Rule 3, remaining duplicated ASEs are examined to identify those that can be evaluated either as constrained elements or as reference elements (subroutine 7.5-6). Such identification relies on the availability of target-end shape elements (TSEs) at the other end(s) of each ASE's defining and dependent EGCs.

Cases where no ASEs are forwarded to the evaluation phase during iteration due to recurrently unavailable TSEs may arise and prevent the ASE transposition procedure from converging (decision 7.5-B). Possible cyclic references in the original EGC schema, which are not checked for, constitute a known cause of ASE evaluations being delayed. Issues were also encountered in specific occurrences of ternary ECG. In those cases, user intervention is still required to select one ASE and a corresponding evaluation mode to force the procedure to resume (step 7.5-7).

7.6.5.4 Evaluation mode

Duplicated ASEs for which an evaluation mode was identified in the previous phase are processed in this fourth phase (subroutine 7.5-8). Depending on the evaluation mode, particular defining and/or dependent EGCs are identified to make up, along with enforced implicit relationships, a sufficient set of enforced relationships ensuring each ASE's well-constrained status and unambiguous evaluation.

7.6.5.5 Sorting assessed ASEs

The fifth and last phase of the procedure takes in well-constrained ASEs. Those are evaluated according to their corresponding set of enforced relationships identified in the second or fourth phases. Once an ASE is confirmed as a new TSE, the EGCs from its set are transposed, mapped and characterized as enforced constraints (steps 7.5-9 thru -11).

Conversely, ASE instances for which transposition failed are deleted from the target model's image in the difference model, along with all the EGC instances they relate to. ASEs found to be unique to the reference model are mapped to ***Dead_End*** instances at their target end (steps 7.5-12 and -13). The ASE transposition procedure ends when all the ASEs from the original EGC schema are either transposed or found to be unique at the end of the fifth phase (decision 7.5-D). Otherwise, the procedure goes back to the second phase where new TSEs and corresponding relationships are used to evaluate the remaining duplicated ASEs (step 7.5-4).

7.7 Procedure for ECG transposition

The flow chart of Figure 7.6 details the overall procedure followed to transpose the remaining EGC instances from the original EGC schema in the target model. At the beginning of this procedure, original ASEs have been transposed with respect to the target shape, along with enforced EGCs. Neither a specific processing sequence nor user intervention is required at this stage. Essentially, EGC transposition involves up to four simple successive inquiries on individual EGC instances, depending on whether they are logical or dimensional.

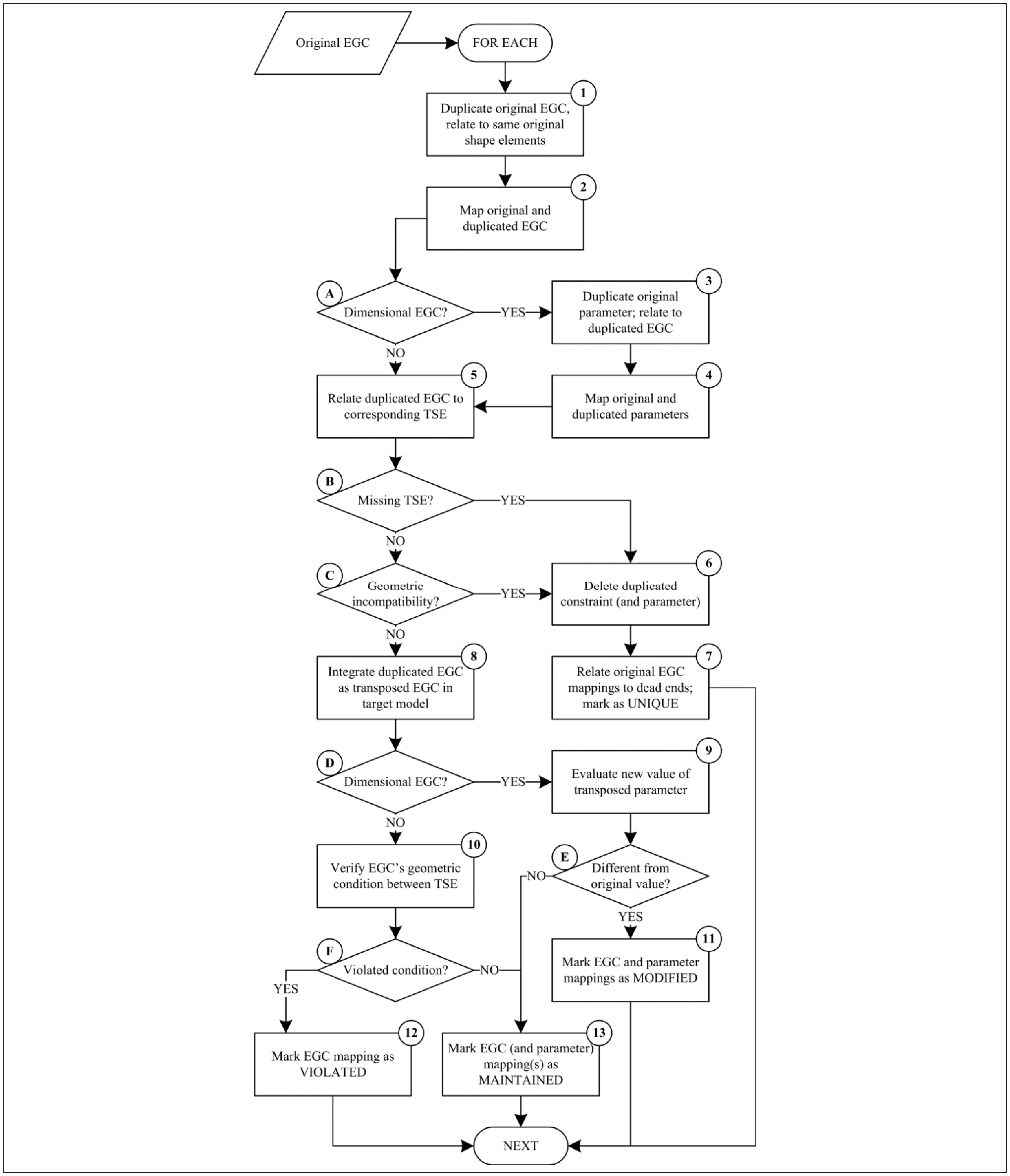


Figure 7.6 Flow chart representing the EGC transposition procedure

Equivalency remains a subjective notion in CAD modeling. Given that the proposed method is designed to compare CAD models regardless of their original format, the matter of differing numeric precisions in the evaluation of geometric elements must be accounted for. As in state-of-the-art CAD and geometric comparison methods, the specification of linear and angular tolerance values by the user to specify the ranges within which similar measured values are considered equivalent is considered throughout the procedures described in this paper.

7.7.1 Searching for TSE

Original shape elements related to an original EGC are examined to determine if they are mapped to TSEs from the target model (steps 7.6-1 thru -5). If original shape elements are found to be unique to the reference model (decision 7.6-B), the transposition is cancelled and the original EGC remains unique to the reference model (steps 7.6-6 and -7).

7.7.2 Checking geometric compatibility

Identified TSEs are examined to determine if they are still geometrically compatible with the transposed EGCs (decision 7.6-C). If one TSE is of a different geometric type, preventing it from being constrained by the transposed EGC, the original EGC remains unique to the reference model.

7.7.3 Locating difference through transposed logical EGC

The original and the transposed EGCs are mapped between the reference and target models (step 7.6-8). If the transposed EGC is logical (decision 7.6-D), corresponding TSEs are examined to determine if the geometric condition imposed by the transposed EGC is maintained in the target model (step 7.6-10). If the geometric condition is maintained (decision 7.6-F), the transposed EGC is characterized as a maintained constraint (step 7.6-13); otherwise, the transposed logical EGC is characterized as a violated constraint (step 7.6-12).

7.7.4 Locating difference through transposed dimensional EGC

If the transposed EGC is dimensional, the transposed design parameter is re-evaluated by measuring the TSE accordingly (step 7.6-9). If the re-evaluated value is equivalent to the original value within a given tolerance threshold (decision 7.6-E), the transposed EGC and its corresponding parameter are characterized as maintained (step 7.6-13); otherwise, they are characterized as modified (step 7.6-11).

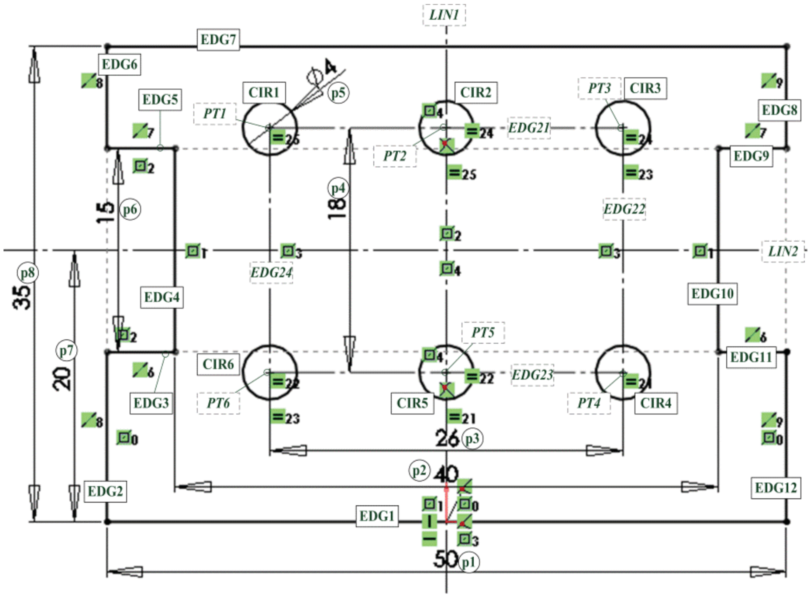
7.8 Example: 2D CAD sketches

In this section, the constraint-based comparison method is applied to the comparison of two 2D CAD sketches to demonstrate the ASE and EGC transposition procedures, as well as to illustrate the usefulness the constraint-based representation of shape differences. Two-dimensional (2D) CAD sketches were chosen over 3D CAD shape representations for this example for their simplicity and relevance. Two-dimensional CAD sketches constitute the most common example of constrained explicit geometry representations available in modern 3D CAD environments. Given that both 2D sketches and explicit constrained 3D shape representations can be abstracted equally with respect to the CAD data meta-model proposed in the accompanying paper, we consider the use of 2D sketches to illustrate our constraint-based method as relevant as if it was applied to 3D shapes.

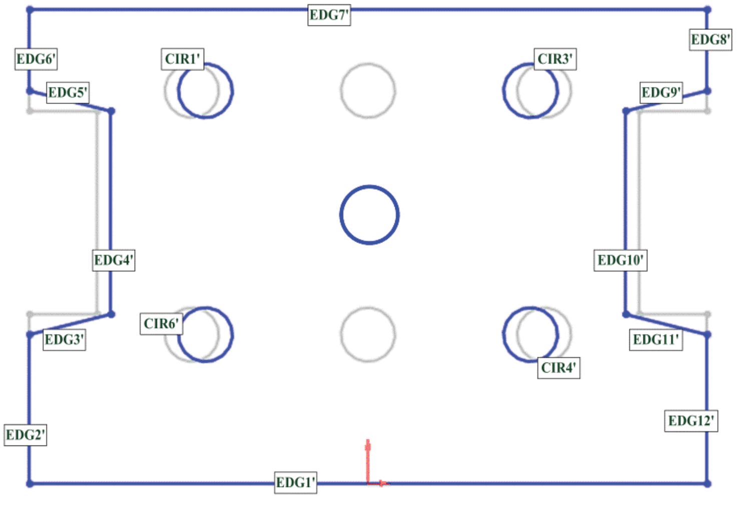
7.8.1 Difference model and shape element mappings

Figure 7.7 presents the reference (a) and target (b) sketches for this example. Two main modifications were applied to the reference sketch to generate the target sketch:

- The depths of the two lateral openings were increased, with an outward draft applied to their respective side edges; and
- The number of holes in the central pattern was brought down from six to five, with the overall width of the pattern reduced.



(a) Reference sketch



(b) Target sketch

Constraint syntax:

#. **TYPE**([constrained];[reference];[parameter])

TYPE in Italics: Composite EGC
[*shape element*] in Italics: ASE

Constraint declarations:

0. **SYMMETRIC**([EDG2,EDG12];[*LIN1*])
1. **SYMMETRIC**([EDG4,EDG10];[*LIN1*])
2. **SYMMETRIC**([EDG3,EDG5];[*LIN2*])
3. **SYMMETRIC**([*EDG22*,*EDG24*];[*LIN1*])
4. **SYMMETRIC**([*EDG21*,*EDG23*];[*LIN2*])
5. ***PARALLEL_DISTANCE***([EDG7];[EDG1];[p8])
    5.1. **SAME_DIRECTION**([EDG7];[EDG1])
    5.2 **DISTANCE**([EDG7];[EDG1];[p8])
6. **COINCIDENT**([EDG11];[EDG3])
7. **COINCIDENT**([EDG9];[EDG5])
8. **COINCIDENT**([EDG6];[EDG2])
9. **COINCIDENT**([EDG8];[EDG12])
10. ***PARALLEL_DISTANCE***([EDG5];[EDG3];[p6])
    10.1. **SAME_DIRECTION**([EDG5];[EDG3])
    10.2 **DISTANCE**([EDG5];[EDG3];[p6])
11. **INCIDENT**([*LIN1*];[ORIGIN])
12. **INCIDENT**([EDG1];[ORIGIN])
13. ***PARALLEL_DISTANCE***([*EDG21*];[*EDG23*];[p4])
    13.1. **SAME_DIRECTION**([*EDG21*];[*EDG23*])
    13.2 **DISTANCE**([*EDG21*];[*EDG23*];[p4])
14. ***PARALLEL_DISTANCE***([*EDG22*];[*EDG24*];[p3])
    14.1. **SAME_DIRECTION**([*EDG22*];[*EDG24*])
    14.2 **DISTANCE**([*EDG22*];[*EDG24*];[p3])
15. ***INTERSECT***([*PT2*];[*LIN1*,*EDG21*])
    15.1 **INCIDENT**([*PT2*];[*LIN1*])
    15.2 **INCIDENT**([*PT2*];[*EDG21*])
16. ***INTERSECT***([*PT5*];[*LIN1*,*EDG23*])
    16.1 **INCIDENT**([*PT5*];[*LIN1*])
    16.2 **INCIDENT**([*PT5*];[*EDG23*])
17. ***PARALLEL_DISTANCE***([EDG12];[EDG2];[p1])
    17.1. **SAME_DIRECTION**([EDG12];[EDG2])
    17.2 **DISTANCE**([EDG12];[EDG2];[p1])
18. ***PARALLEL_DISTANCE***([*LIN2*];[EDG1];[p7])
    18.1. **SAME_DIRECTION**([*LIN2*];[EDG1])
    18.2 **DISTANCE**([*LIN2*];[EDG1];[p7])
19. ***PARALLEL_DISTANCE***([EDG10];[EDG4];[p2])
    19.1. **SAME_DIRECTION**([EDG10];[EDG4])
    19.2 **DISTANCE**([EDG10];[EDG4];[p2])
20. **RADIUS**([CIR1];[];[p5])
21. **RADIUS_EQUAL***([CIR4];[CIR5])
22. **RADIUS_EQUAL***([CIR5];[CIR6])
23. **RADIUS_EQUAL***([CIR6];[CIR3])
24. **RADIUS_EQUAL***([CIR3];[CIR2])
25. **RADIUS_EQUAL***([CIR2];[CIR1])
26. **FIXED_IN_X_DIR**([EDG1])
27. **FIXED_IN_Y_DIR**([*LIN1*])

(c) Listing of original EGC declarations

Figure 7.7 Reference and target 2D CAD sketches compared with respect to the original EGC schema

Shape_Element instances from the reference sketch representing the original PSEs are labelled and their corresponding matches in the target sketch, as the outcome of a shape element matching algorithm, are labelled analogously (e.g. original edge instance labelled **EDG3** from the reference sketch is mapped to new edge instance labelled **EDG3'** in the target sketch). ***Shape_Element*** instances representing original ASEs are labelled in italics and remain unmapped at this point. **Design_Parameter** instances are also labelled in the reference sketch (e.g. **p3**).

Symbols distinguishing EGC types on the reference sketch (Figure 7.7(a)), such as symmetric (refs. 0 thru 4), originate from Dassault Systèmes SolidWorks® 2010 (Dassault Systèmes, 2010), the 3D CAD software used to construct and constrain the sketches. A detailed listing of original ***Constraint*** instances, with the corresponding syntax, is presented in Figure 7.7(c). Constrained elements, reference elements and design parameters, when applicable, are identified for each instance in the difference model. This listing illustrates the breakdown of nine **Composite_Constr** instances (listed in italics), originating from the software's specific geometric constraint typology, into 18 ***Elementary_Constr*** instances as part of the difference model's initialization. For example, ***INTERSECT*** instances, referencing original EGCs constraining the position of a point at the intersection of two lines, are split into two joint **INCIDENCE** instances. All concrete types of ***Elementary_Constr*** and corresponding geometric conditions are derived from Bettig and Shah (2001), except for the **RADIUS_EQUAL*** type which was added because of its relevance in practical CAD modeling.

7.8.2 ASE Transposition

Out of the 37 ***Elementary_Constr*** instances making up the original EGC schema in the difference model, 17 instances require ASEs to be transposed with respect to the target sketch before being themselves transposed. The procedure begins with the classification of enforceable relationships into three groups – implicit, defining and dependant – for each of the 12 original ASEs (6 points and 6 lines), as schematized in Figure 7.8.

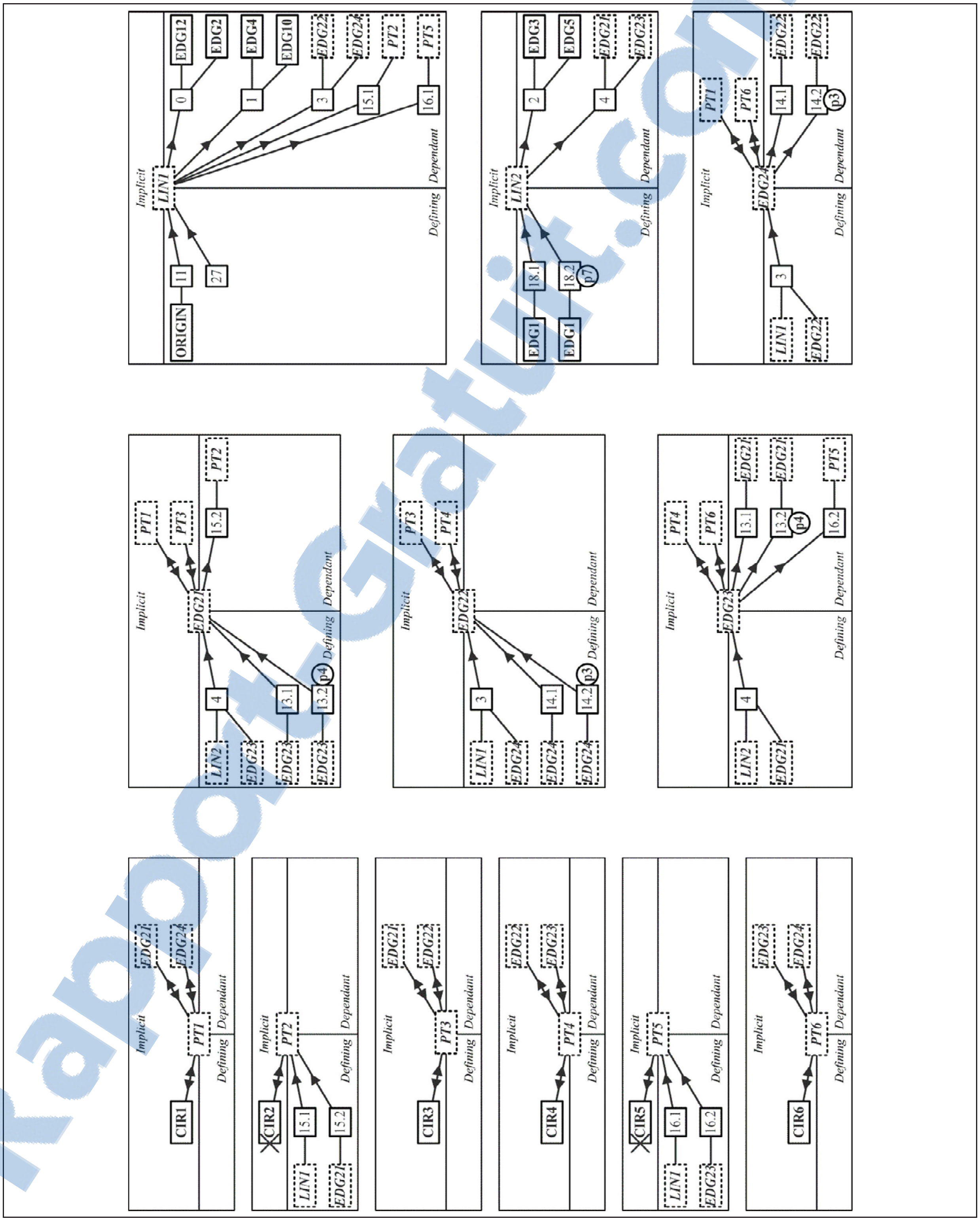


Figure 7.8 Classification of enforceable relationships for all 12 original ASEs

At this early stage, ***Shape_Element*** instances related to duplicated ASEs are either physical shape elements instances from the target sketch or other duplicated ASE instances (italics in dashed-lined boxes). The absence of ***Shape_Element*** instances **CIR2’** and **CIR5’** in the target sketch, given their unique counterparts in the reference sketch, voids two implicit relationships for ASE instances **PT2’** and **PT5’**, respectively; these relationships are withdrawn from the transposition process.

The four iterations required to transpose all 12 ASE instances are detailed in Table 7.1. Priority was given to the enforcement of implicit relationships in the first two iterations. Firstly, sufficient geometric information is found to evaluate four out of the six auxiliary point instances, i.e., those representing the centers of the four mapped circular edges in the pattern. Then, as the transposed auxiliary point instances become TSEs at the end of the first iteration, auxiliary edge instances ***EDG21’***, ***EDG22’***, ***EDG23’*** and ***EDG24*’** are transposed according to their now transposed end points as part of the second iteration.

Tableau 7.1 Report of the four iterations required for ASE transposition from reference to target CAD sketches

Iter.	**Transposed ASE**	**Implicit relationships (I) to**	**Evaluation mode**	**Defining constraints (II)**	**Dependent constraints (III)**	**Transposition status**
1	*PT1’*	CIR1’	--	--	--	New TSE
	PT3’	CIR3’	--	--	--	New TSE
	PT4’	CIR4’	--	--	--	New TSE
	PT6’	CIR5’	--	--	--	New TSE
2	*EDG21’*	*PT1’, PT3’*	--	--	--	New TSE
	EDG22’	*PT3’, PT4’*	--	--	--	New TSE
	EDG23’	*PT4’, PT6’*	--	--	--	New TSE
	EDG24’	*PT1’, PT6’*	--	--	--	New TSE
3	*LIN1’*	--	As-constrained	11, 27	--	New TSE
	LIN2’	--	Mixed	--	2, 4	New TSE
	PT2’	--	As-reference	--	--	Delay, mixed
	PT5’	--	As-reference	--	--	Delay, mixed
4	*PT2’*	--	Mixed (imposed)	15.1, 15.2	--	New TSE
	PT5’	--	Mixed (imposed)	16.1, 16.2	--	New TSE

Since all the available implicit relationships were used in the first part of the transposition, the remaining duplicated ASE instances had to be transposed using defining or dependant EGC. At the third iteration, some of the dependant constraints relating to ***LIN1’*** auxiliary line instance were still related to unevaluated ASE instances (**PT2’** and **PT5’**), thus triggering the ‘as-constrained’ evaluation mode. The corresponding logical defining EGCs (refs. 11 and 27) allowed the evaluation of a well-constrained ***LIN1’*** instance. Conversely, all of the ***LIN2’*** auxiliary line instance’s EGCs were relationships to TSEs, leading to its evaluation via the mixed mode. The first attempt at evaluating via the mixed mode, using an enforceable set comprised of all logical dependant constraints, succeeded immediately, as the two **SYMMETRIC** constraint instances (refs. 2 and 4) formed a redundant, yet sufficient and conflict-free set.

Auxiliary point instances **PT2’** and **PT5’** were tagged for evaluation via the ‘as-reference’ mode during the third iteration, given that no dependent EGC were found to relate to other ASEs. However, since no dependant EGCs were actually available for evaluation, that evaluation was delayed until it could be attempted via the mixed mode, i.e., during the fourth and last iteration. Auxiliary point instances **PT2’** and **PT5’** were transposed by enforcing corresponding logical defining EGCs.

7.8.3 EGC transposition

As a result of ASE transposition, eight ***Constraint*** instances were transposed to the target model and marked as enforced constraints. The remaining 29 ***Constraint*** instances from the original EGC schema were transposed afterwards, with the results presented in Table 7.2, where, for each instance, target-end constrained and reference elements are identified. ***Constraint*** instances of type **RADIUS** (ref.20) and **FIXED_IN_X_DIR** (ref.26) do not comprise reference elements since they only represent unary constraints.

Tableau 7.2 Results of the transposition of EGC from the reference to the target CAD sketches

Ref.	Type	Target-end constrained element(s)	Target-end reference element	Constraint status	New design parameter	Original design parameter
0	SYMMETRIC	EDG2’, EDG12’	*LIN1’*	Maintained		
1	SYMMETRIC	EDG4’, EDG10’	*LIN1’*	Maintained		
2	SYMMETRIC	EDG3’, EDG5’	*LIN2’*	*Enforced*		
3	SYMMETRIC	*EDG22’*, *EDG24’*	*LIN1’*	Maintained		
4	SYMMETRIC	*EDG21’*, *EDG23’*	*LIN2’*	*Enforced*		
5	*PARALLEL_DISTANCE*	EDG7’	EDG1’	Maintained	p8’ = 35 mm	p8 = 35 mm
5.1	SAME_DIRECTION	EDG7’	EDG1’	Maintained		
5.2	DISTANCE	EDG7’	EDG1’	Maintained	p8’ = 35 mm	p8 = 35 mm
6	COINCIDENT	EDG11’	EDG6’	**Violated**		
7	COINCIDENT	EDG9’	EDG5’	**Violated**		
8	COINCIDENT	EDG6’	EDG2’	Maintained		
9	COINCIDENT	EDG8’	EDG12’	Maintained		
10	*PARALLEL_DISTANCE*	EDG5’	EDG3’	**Violated**		
10.1	SAME_DIRECTION	EDG5’	EDG3’	**Violated**		
10.2	DISTANCE	EDG5’	EDG3’	**Modified**	**p6’ = 0 mm**	**p6 = 15 mm**
11	INCIDENT	*LIN1’*	ORIGIN’	*Enforced*		
12	INCIDENT	EDG1’	ORIGIN’	Maintained		
13	*PARALLEL_DISTANCE*	*EDG21’*	*EDG23’*	Maintained	p4’ = 18 mm	p4 = 18 mm
13.1	SAME_DIRECTION	*EDG21’*	*EDG23’*	Maintained		
13.2	DISTANCE	*EDG21’*	*EDG23’*	Maintained	p4’ = 18 mm	p4 = 18 mm
14	*PARALLEL_DISTANCE*	*EDG22’*	*EDG24’*	**Modified**	**p3’ = 24 mm**	**p3 = 26 mm**
14.1	SAME_DIRECTION	*EDG22’*	*EDG24’*	Maintained		
14.2	DISTANCE	*EDG22’*	*EDG24’*	**Modified**	**p3’ = 24 mm**	**p3 = 26 mm**
15	*INTERSECT*	*PT2’*	*LIN1’, EDG21’*	*Enforced*		
15.1	INCIDENT	*PT2’*	*LIN1’*	*Enforced*		
15.2	INCIDENT	*PT2’*	*EDG21’*	*Enforced*		
16	*INTERSECT*	*PT5’*	*LIN1’, EDG23’*	*Enforced*		
16.1	INCIDENT	*PT5’*	*LIN1’*	*Enforced*		
16.2	INCIDENT	*PT5’*	*EDG23’*	*Enforced*		
17	*PARALLEL_DISTANCE*	EDG12’	EDG2’	Maintained	p1’ = 50 mm	p1 = 50 mm
17.1	SAME_DIRECTION	EDG12’	EDG2’	Maintained		
17.2	DISTANCE	EDG12’	EDG2’	Maintained	p1’ = 50 mm	p1 = 50 mm
18	*PARALLEL_DISTANCE*	*LIN2’*	EDG1’	Maintained	p7’ = 20 mm	p7 = 20 mm
18.1	SAME_DIRECTION	*LIN2’*	EDG1’	Maintained		
18.2	DISTANCE	*LIN2’*	EDG1’	Maintained	p7’ = 20 mm	p7 = 20 mm
19	*PARALLEL_DISTANCE*	EDG10’	EDG4’	**Modified**	**p2’ = 38 mm**	**p2 = 40 mm**
19.1	SAME_DIRECTION	EDG10’	EDG4’	Maintained		
19.2	DISTANCE	EDG10’	EDG4’	**Modified**	**p2’ = 38 mm**	**p2 = 40 mm**
20	RADIUS	CIR1’	--	Maintained	p5’ = 4 mm	p5 = 4 mm
21	RADIUS_EQUAL	CIR4’, n/a	--	**Unique**		
22	RADIUS_EQUAL	n/a, CIR6’	--	**Unique**		
23	RADIUS_EQUAL	CIR6’, CIR3’	--	Maintained		
24	RADIUS_EQUAL	CIR3’, n/a	--	**Unique**		
25	RADIUS_EQUAL	n/a, CIR1’	--	**Unique**		
26	FIXED_IN_X_DIR	EDG1’	--	Maintained		
27	FIXED_IN_Y_DIR	*LIN1’*	--	*Enforced*		

7.8.4 Summary

Figure 7.9 presents a graphical summary of the differences located between the two sketches via our constrained-based comparison method. Constraints unique to the reference sketch along with violated and modified constraints are tagged, while transposed ASEs are illustrated and labelled accordingly. Modified dimensional constraints are shown with their correspondingly modifier parameters. The outward drafts applied to the side edges of the lateral openings led to the violation of two coincidence (colinearity) constraints and of two parallelism (distance) constraints originally applied to these side edges. The removal of 2 of the 6 original holes in the central pattern also invalidated most of all radius equality constraints originally applied to the circular edges since they were made to relate serially ($r_1 = r_2$, $r_2 = r_3$, $r_3 = r_4$, etc.).

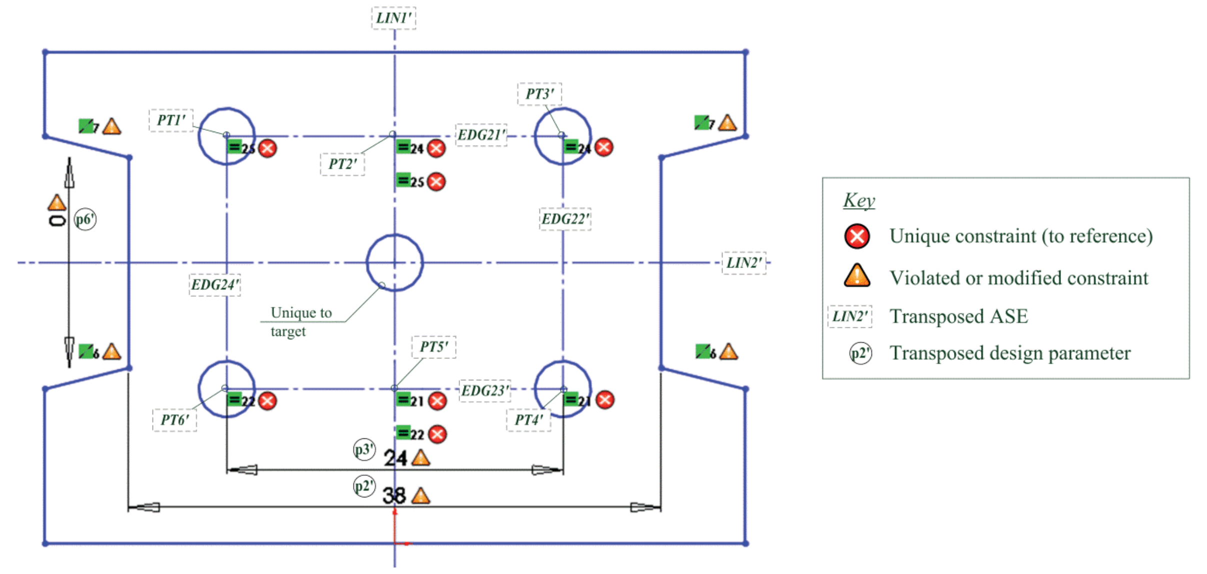


Figure 7.9 Graphical summary of the constraint-based differences identified between the compared 2D CAD sketches

7.8.5 Discussion

One circular edge, representing a new center hole in the pattern, is unique to the target sketch and, thus, was left out of the constraint-based representation of differences. No EGC from the original sketch can be transposed and used to represent this specific difference since the new center hole is not part of the original shape specification. It may be considered as a limitation of the proposed difference calculation and representation method. However, since shape element mappings are part of the difference model along with EGC mappings, addition of new shape elements in target model with respect to the reference model can easily be detected and represented at the shape level. It is not left out of the difference model.

7.9 Conclusion

In this paper, we introduced a method of comparing detached and heterogeneously formatted 3D CAD models of mechanical parts which could respectively represent the original and new version of the part geometry. The calculation and representation of shape differences is based on transposing the explicit geometric constraint (EGC) schema embodying original design intent in the reference model to a similar part geometry. Expressed in terms of modified dimensions and violated geometric conditions, shape differences are therefore described on a level of abstraction more comprehensive to mechanical designers. Differences are represented in difference models which conform to the difference meta-model (DMM) presented in an accompanying paper whose purpose is to capture model elements mappings between the compared models and thus detail differences from the geometric level up to the parametric level. Aside from the ECM scenario addressed in this work, the compared models could also come from 3D search results (Brière-Côté, Rivest et Maranzana, 2012b).

Constraint-based difference calculation begins with the matching of B-Rep shape elements between the compared models to establish the low-level mappings on which EGC schema transposition operates. State-of-the-art B-Rep data structure matching algorithms meeting the requirements stated in this paper, such as the one implemented in PTC CoCreate® Modeling PE (CoCreate Software GmbH, 2008; CoCreate Software GmbH et Gutierrez, 3 juin 2010), can be used. Efforts could however be focused in improving the precision of such algorithms as evaluation trials from previous work (Brière-Côté, Rivest et Maranzana, 2013) have revealed their limitations. Then, face, edge and vertex mappings are exploited to transpose the original EGC schema composed of auxiliary shape elements (ASE), EGCs and design parameters to the new part geometry. Procedures to transpose ASEs and EGCs are detailed and exemplified through the simulated comparison of two similar CAD sketches, a simple, common and mostly available example of constrained explicit geometry representations.

Future work should focus on enhancing the comprehensibility and intuitiveness of difference representation by extending the DMM to include algebraic constraints – commonly referred to as design rules – and form features and by developing the corresponding new difference calculation procedures. Since the DMM is meta-model independent, extending it would amount to revising the CAD data meta-model. First, including algebraic constraints in the DMM would require also including unbound parameters – i.e., user-defined design parameters used in algebraic constraints, but not related to a dimensional constraint. EGC and parameter mappings produced by EGC schema transposition would then be exploited to transpose unbound parameters and algebraic constraints from reference to target models. Then, including form features in the DMM would require developing a coherent declarative or explicit approach to feature representation as current feature representations are mostly implicit in procedural CAD.

CHAPITRE 8

INTEGRATION OF EXPLICIT GEOMETRIC CONSTRAINTS IN THE COMPARISON OF 3D CAD MODELS FOR PART DESIGN REUSE

Antoine Brière-Côté[1], Louis Rivest[1] and Roland Maranzana[1],
[1]Département de Génie de la production automatisée, École de technologie supérieure,
1100 Notre-Dame W, Montréal, Québec, Canada H3C 1K3

Article présenté lors de la « 9th International Conference on Product Lifecycle Management »
ayant eu lieu du 9 au 11 juillet 2012 à Montréal au Canada.

8.1 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.

8.2 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 information reuse have been proposed and surveyed in the last decade (Cardone, Gupta et Karnik, 2003). Commercial 3D shape-based search engines interfacing with current PLM systems are now available (e.g. 3DSemantix inc. (2011), Siemens PLM Software inc. (2011a)).

However, shape-based retrieval only contributes to the overall part design reuse problem as an opening step (Brière-Côté, Rivest et Maranzana, 2012a; Msaaf, Maranzana et Rivest, 2007). As pictured in Figure 8.1, further analysis of the shape differences between each retrieved similar model and the reference model is still required to identify the optimal candidate for design reuse. 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 shape differences relating to the specific application of part design reuse.

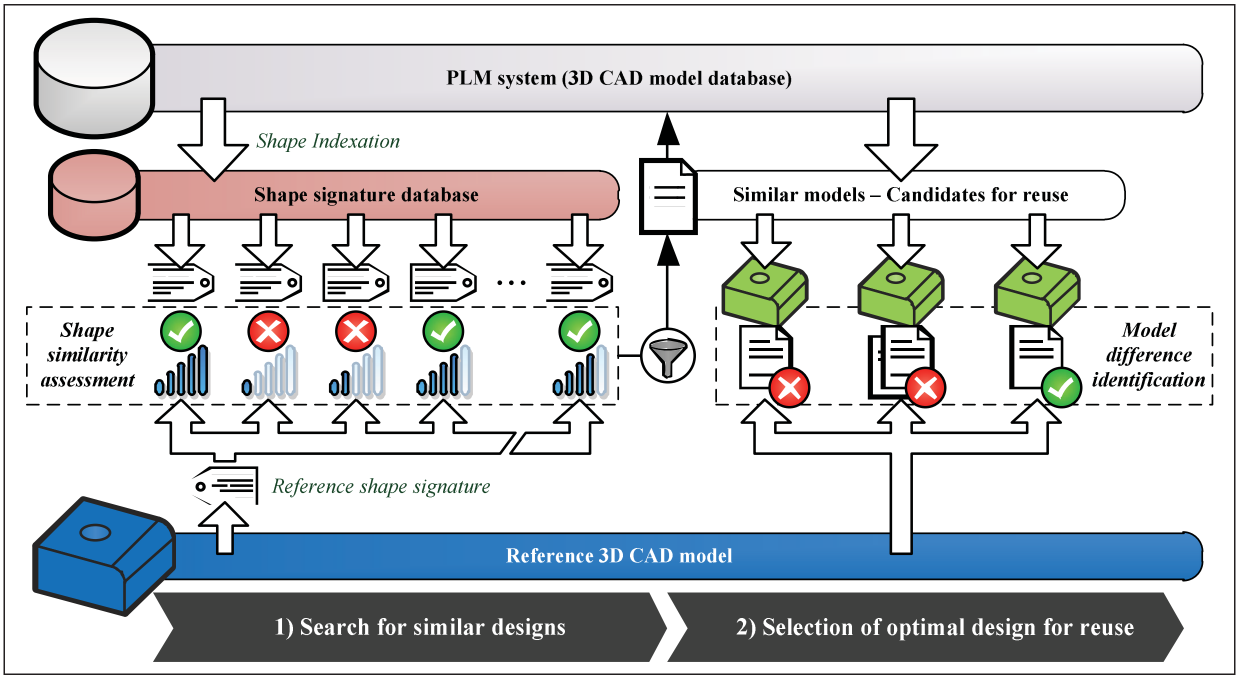


Figure 8.1 Search for similar existing part designs and selection of the optimal part design for reuse

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 American Society of Mechanical Engineers (2009)), 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 8.3 provides background on the 3D CAD model difference identification (MDI) problem and briefly examines previous contributions. Section 8.4 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 8.5. Section 8.6 presents the application of the proposed technique in the comparison of two 2D CAD sketches as a brief illustrative example.

8.3 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 (Brière-Côté, Rivest et Maranzana, 2012a). Similarity assessment generally exploits highly abstracted or reduced geometric model contents (e.g. shape signatures (Cardone, Gupta et Karnik, 2003)) to produce quick diagnoses on two shapes' equivalence ("yes" or "no") or relative similarity (a qualitative, scale-based measure). No details are provided on what actually distinguishes each similar model from the reference. When calculated recurrently, as in shape-based retrieval, 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 ex-pensive, limiting it mostly to pair-wise comparison applications.

8.3.1 Composition of the MDI Problem

Inspired by software model version management (Kolovos *et al.*, 2009), 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 (Yang, Lin et Zhang, 2007).

- 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 (Brière-Côté, Rivest et Maranzana, 2011).

- 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 inter-actions, indented lists and/or reports.

8.3.2 Design-Oriented Difference Representation

A number of 3D CAD model difference calculation techniques were surveyed and detailed by Brière-Côté, Rivest and Maranzana (2012a). 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 Figure 8.2.

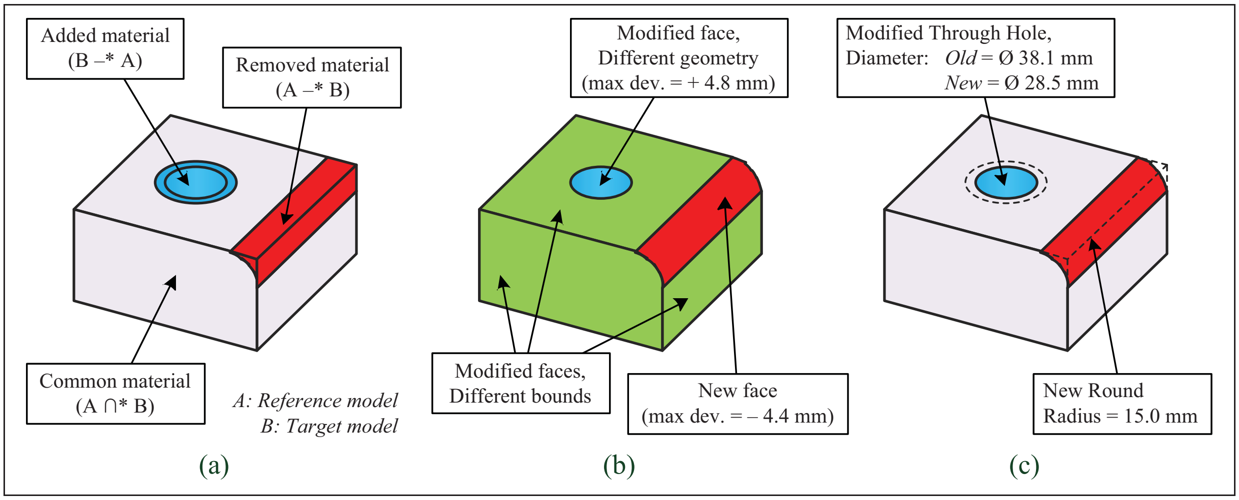


Figure 8.2 Three approaches to the design-oriented representation of a given set of shape differences: a) Delta regions; b) Boundary-based; c) Procedural differences.

A first approach leads to difference models that distinguish localized 3D delta regions of material addition, material removal and common material between part geometries (Figure 8.2(a)). 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. François and Cuillière (2000), Sypkens Smit and Bronsvoort (2009a)). 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 (Figure 8.2(b)), but these may also aggregate deviating facets between tessellated shapes to form distinct difference regions (e.g. Lattice Technology Co. (2010)). Boundary-based differences can be calculated explicitly by using geometric algorithms such as point-to-part deviation calculation (e.g. CapVidia NV (2010)), 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. (3 juin 2010)).

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 (Figure 8.2(c)). De-sign 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.

8.4 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.

8.4.1 Application settings

As part of the part design reuse process described in section 8.2, the MDI problem presents the following characteristics for the compared models:

- They are considered as detached, i.e. no pre-established relation is available be-tween 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 (International Organisation for Standardization, 2005).

8.4.2 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 8.4.3). 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 (International Organisation for Standardization, 2005)). Geometric constraints specify relationships between shape elements and between shape elements and design parameters (i.e. logical and dimensional constraints, respectively), as pictured in Figure 8.3.

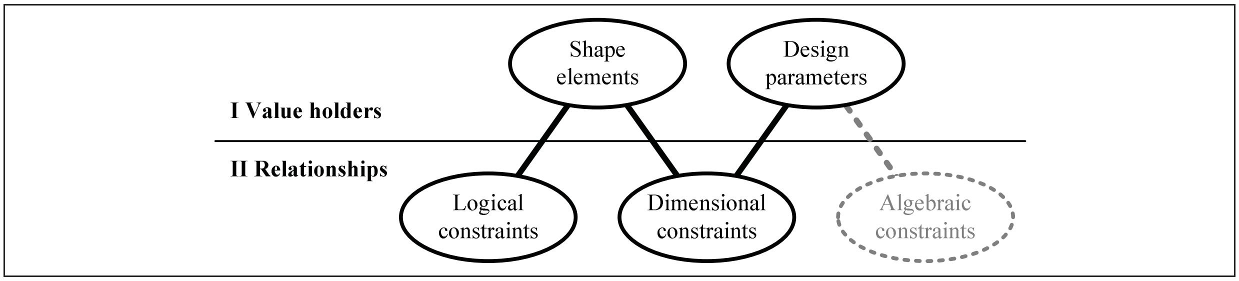


Figure 8.3 The relationships between shape elements, design parameters and constraints[3] (Inspired by Bettig and Shah (2001))

---

[3] Algebraic constraints, or "design rules", specify relationships between design parameters.

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, Di Ruscio and Pierantonio (2008). Schematized in Figure 8.4, the approach is meta-model-independent, which allows us to define an explicit 3D CAD meta-model (MM) for referencing compared data while preserving 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.

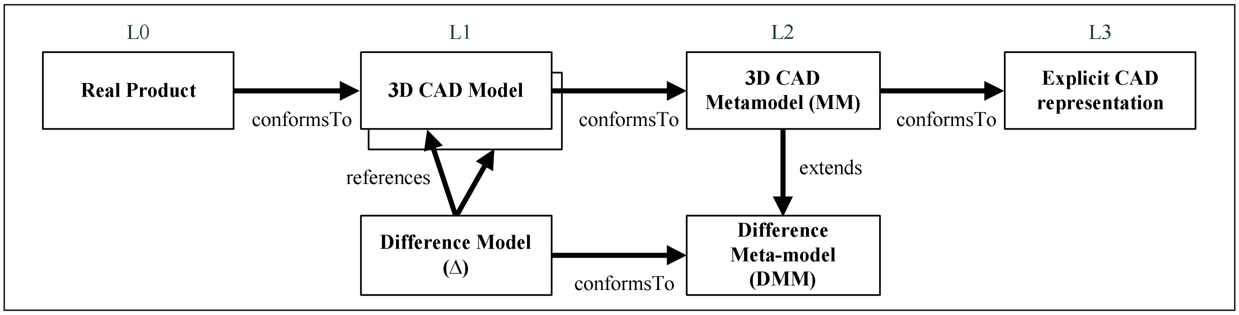


Figure 8.4 Overall structure of the model difference representation approach

8.4.3 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 (Brière-Côté, Rivest et Maranzana, 2011), a suitable algorithm must at least meet these requirements:

- It must operate on detached models, and therefore, static identity-based matching algorithms (Brière-Côté, Rivest et Maranzana, 2012a) 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 Figure 8.5 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).

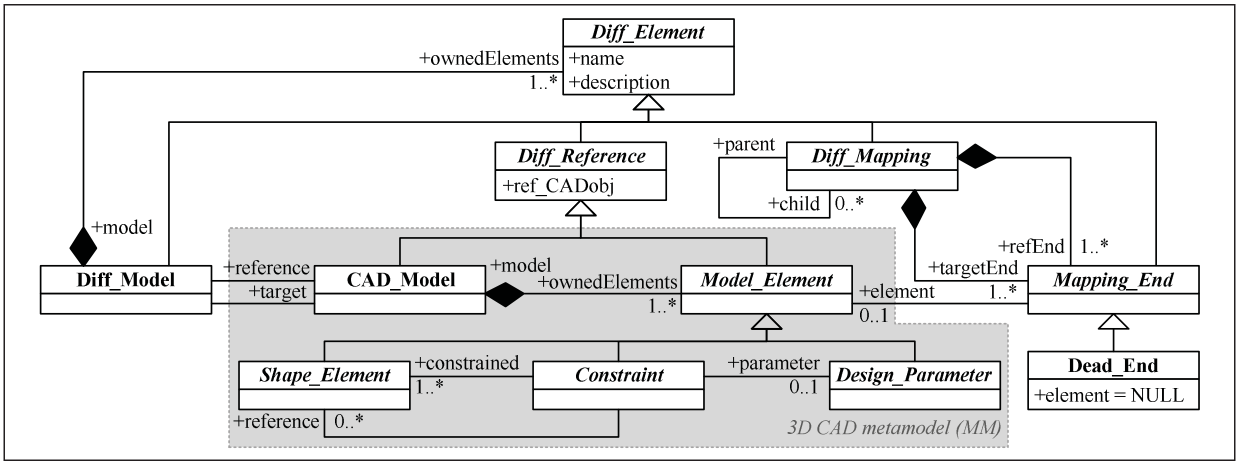


Figure 8.5 Difference meta-model (DMM)

As an example, the B-Rep difference calculation algorithm described by CoCreate Software GmbH and Gutierrez (3 juin 2010) and implemented in PTC CoCreate® Modeling PE (CoCreate Software GmbH, 2008) 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.

8.5 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 Figure 8.6 with an example. In the reference model (left), a "parallel-distance" geometric constraint (e.g. `pgc_with_dimension` entity from ISO STEP Part 108 (International Organisation for Standardization, 2005)) 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*.

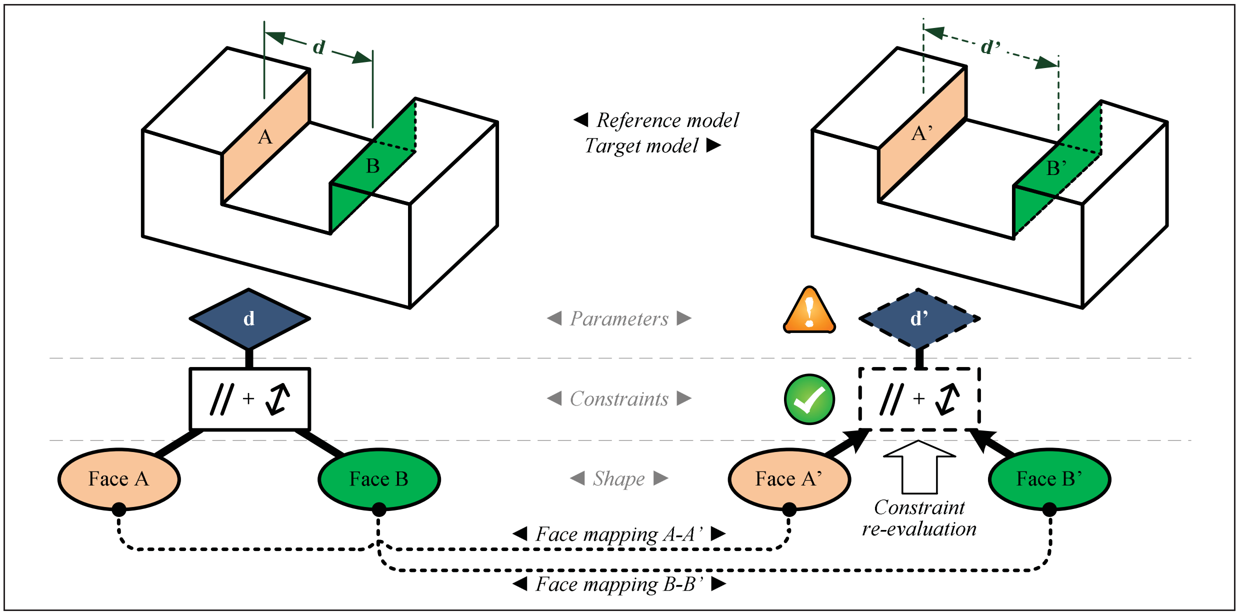


Figure 8.6 Re-evaluation of geometric constraints and related design parameters on target model based on shape element mappings

8.5.1 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 (International Organisation for Standardization, 2005) form a practical set of geometric constraints, while the theoretical set derived by Bettig and Shah (2001) is considered to be consistent.

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 Figure 8.7. 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 Figure 8.6 – 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.

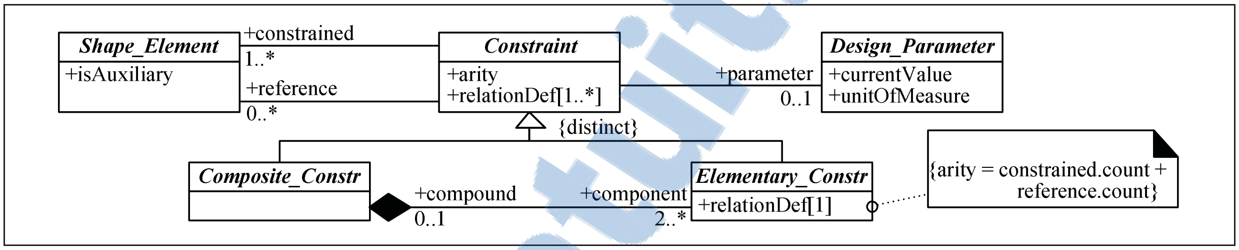


Figure 8.7 Representation of composite and elementary geometric constraints in the DMM

8.5.2 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 8.8 describes how shape element mappings are ultimately processed to elevate shape differences at the level of geometric constraints and design parameters.

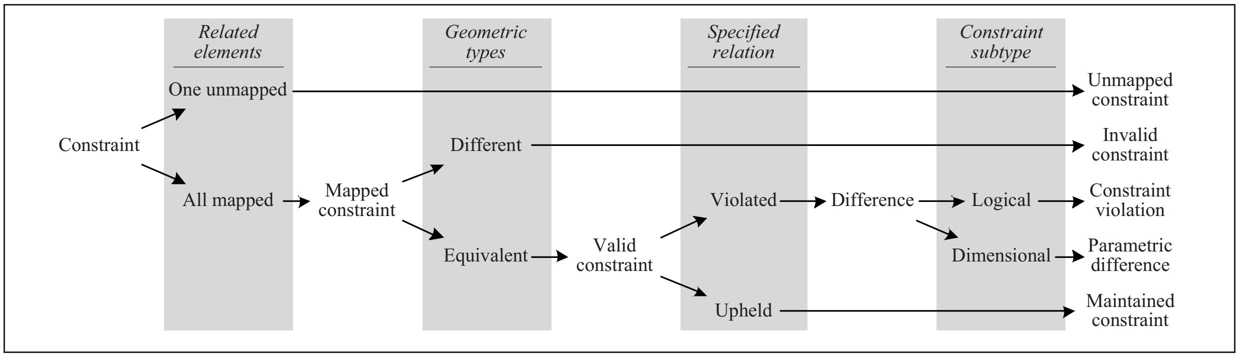


Figure 8.8 Shape element mappings and differences elevated at the level of geometric constraints

8.5.3 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:

$$g \circ f = \{(x, y) \in X \times Y | \exists a \in A: (x, a) \in f \wedge (a, y) \in g\} \tag{8.1}$$

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. (8.1)).

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. (8.1), 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*.

8.6 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 Figure 8.9; nonetheless, the concepts illustrated here fully apply to the comparison of 3D shapes. The left side of Figure 8.9 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).

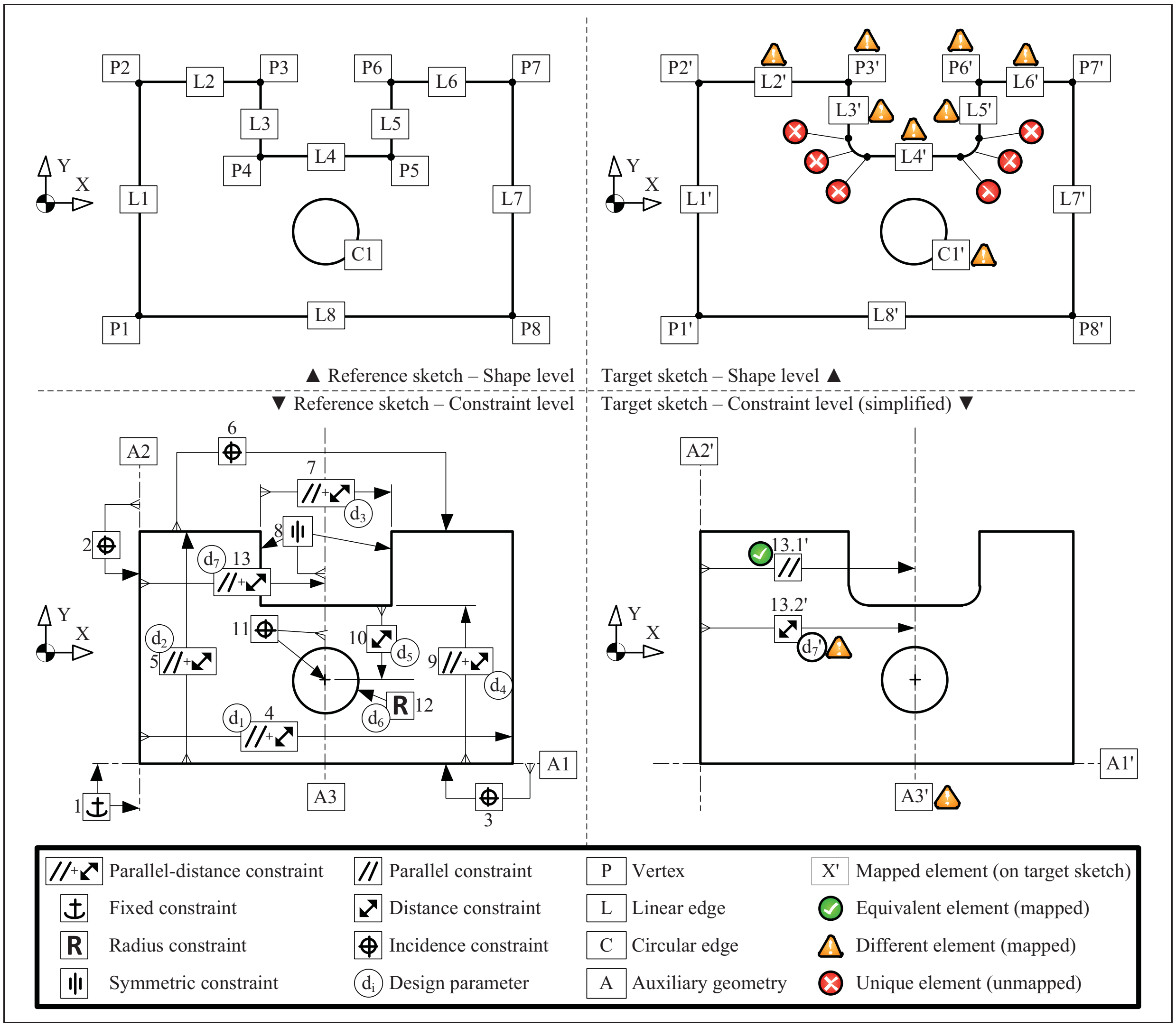


Figure 8.9 Results from the proposed model comparison technique applied to two 2D sketches

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 bot-tom 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 Figure 8.9. 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.

8.7 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.

CHAPITRE 9

SYNTHÈSE ET DISCUSSION

Ce chapitre se veut une rétrospective des différentes propositions présentées au sein des chapitres précédents, i.e., au sein des articles soumis ou publiés. On soulignera d'abord les principales contributions de cette thèse aux problèmes de l'identification des différences géométriques entre modèles CAO et de la transposition du changement entre modèles géométriques experts. Ensuite, seront discutées les limitations de ce travail de recherche en leur adressant des recommandations pour les recherches futures.

9.1 Contributions

9.1.1 Comparaison basée sur les contraintes géométriques

La contribution principale de cette thèse est incarnée par la proposition de cette nouvelle approche de comparaison des modèles CAO et de représentation des différences géométriques basée sur les contraintes géométriques associées à un modèle de référence. De par l'objectif principal des travaux, cette approche se démarque de l'état de l'art récent en favorisant l'interprétation et l'assimilation des résultats de comparaison du point de vue d'un ingénieur et de sa discipline. On parvient ainsi à mieux aborder la problématique de la transposition de la définition géométrique d'un produit entre différents modèles experts qui, dans un contexte réaliste de dispersion – et non d'intégration – du modèle produit, doit faire intervenir le savoir-faire de chaque expert impliqué dans le développement de ce produit.

L'approche proposée privilégie d'abord la comparaison de géométries représentées explicitement, telle une représentation par les frontières (B-Rep), au détriment de la comparaison de géométries représentées implicitement, comme par une représentation procédurale par arbre de construction. Sans les opérations de modélisation qui s'expriment comme des caractéristiques de forme, voir fonctionnelles, et qui « enrichissent » les géométries dans les modèles CAO implicites, la nature des résultats de comparaisons de modèles explicites est grandement limitée en terme d'expressivité de l'intention de conception ou, dans le cas de la transposition du changement, d'expressivité de « l'intention d'évolution ». Grâce à l'approche basée sur les contraintes géométriques proposée, la comparaison des géométries explicites peut désormais produire des résultats sémantiquement significatifs pour des ingénieurs en développement de produit.

Ainsi, l'exploitation conjointe des associations établies entre deux représentations B-Rep similaires et d'un schéma de contraintes géométriques associé à la géométrie au sein d'un modèle CAO permet maintenant d'exprimer les différences d'ordre géométrique entre deux modèles selon un formalisme familier très similaire au formalisme normalisé ASME Y14.5 *Geometric Dimensioning and Tolerancing* (American Society of Mechanical Engineers, 2009) fortement répandu dans la pratique industrielle. L'approche proposée permet la représentation des différences géométriques en termes de :

- Contraintes dimensionnelles (distance, longueur, rayon, diamètre, etc.) dont les valeurs ont évolué,
- Contraintes logiques (coïncidence, parallélisme, perpendicularité, symétrie, etc.) dont les relations entre les éléments géométriques contraints ne sont plus respectées, et
- Contraintes ne pouvant être transposées d'une géométrie à une autre étant donné l'absence d'éléments géométriques contraints équivalents.

L'application de l'approche de comparaison proposée au scénario spécifique de la transposition de l'évolution de la définition géométrique entre modèles experts commande des conditions d'application particulière que plusieurs solutions constituant l'état de l'art ne peuvent satisfaire. Notamment, la caractérisation du delta géométrique entre deux modèles doit pouvoir être réalisée entre des modèles de formats CAO hétérogènes. Grâce au modèle de représentation des différences proposée, i.e., à un méta-modèle des différences, la présente approche exige simplement que les deux géométries soient représentées par leurs frontières (B-Rep), ce qui permet une comparaison fondamentalement géométrique. De plus, un seul des deux modèles comparés, identifié comme le modèle de référence, doit impérativement présenter un schéma de contraintes géométriques explicites incarnant un point de vue spécifique sur la géométrie. Puisque ce sont les éléments géométriques des modèles qui sont comparés, et non les contraintes géométriques elles-mêmes, la présence d'un schéma de contraintes au sein du modèle cible n'est pas exigée.

9.1.2 Contribution à l'état de l'art

Les premiers travaux réalisés dans le cadre de cette thèse contribuent au problème de l'identification des différences entre modèles géométriques en présentant un portrait très large de l'état de l'art des méthodes et des mécanismes de comparaison des modèles en trois dimensions. Ce portrait est qualifié de très large, puisqu'il couvre non seulement les développements publiés au sein d'articles scientifiques, mais également les développements logiciels commerciaux, les brevets d'invention, les normes et les procédures industrielles.

Plusieurs revues de la littérature ont été publiées par le passé sur l'application de la comparaison visant particulièrement la recherche basée sur la géométrie (Cardone, Gupta et Karnik, 2003; Iyer *et al.*, 2005; Li, Liu et Ramani, 2004; Tangelder et Veltkamp, 2004; Yang, Lin et Zhang, 2007; Zhang et Peng, 2009). Toutefois, aucun état de l'art n'avait été jusqu'ici réalisé sur le sujet spécifique de la comparaison binaire de modèles 3D afin d'en repérer et représenter dans les détails les différences géométriques.

Un autre mérite de l'état de l'art présenté dans cette thèse est d'incorporer un inventaire méthodique des outils logiciels d'origines commerciales capables d'identifier des différences entre deux modèles CAO. Afin de mieux représenter les possibilités et les capacités de ces outils dans la comparaison de modèles, deux séries d'essais pratiques encadrées par des protocoles d'expérimentation ont également été réalisées. Une analyse comparative des résultats obtenus lors de ces essais, basée sur cinq critères bien définis et un barème en trois échelons, permet du coup de mieux apprécier le potentiel et les limites des différentes méthodes de comparaison identifiées dans cet état de l'art – lorsque ces méthodes ont pu être associées avec certitude aux différents outils évalués, ce qui n'étaient pas toujours le cas.

9.1.3 Classification des scénarios de comparaison des modèles CAO

Lors de la synthèse de l'état de l'art sur les méthodes et les mécanismes de comparaison des modèles CAO, et plus particulièrement lors de l'inventaire des outils logiciels commerciaux, plusieurs scénarios d'utilisation de la comparaison ont pu être recensés. Dans un souci de bien positionner le contexte et les contraintes des présents travaux, cette thèse présente une classification en six catégories des nombreux scénarios de comparaison des modèles CAO, ainsi que l'identification des domaines de solution et des mandats principaux de la comparaison, initialement illustrés à la figure 3.2 qui est reproduite à la figure 9.1. Cette classification originale et la définition des mandats principaux constituent deux contributions importantes.

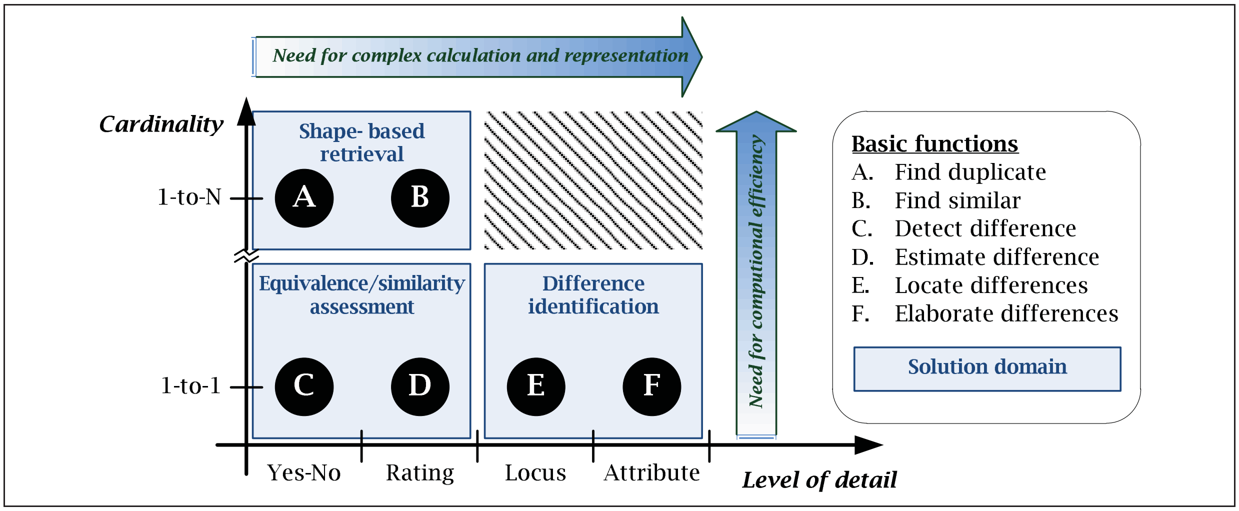


Figure 9.1 Relation entre le niveau de détail et la cardinalité définissant les domaines de solution et les fonctions de base de la comparaison CAO
Tirée de Brière-Côté, Rivest et Maranzana (2012a)

Un premier facteur distinctif émergeant de l'étude des scénarios est la cardinalité de la comparaison, à savoir si le modèle de référence doit être comparé à un ou quelques modèles cibles ou à un grand échantillon de modèles. Ce facteur permet de distinguer fondamentalement la comparaison des modèles CAO appliquée à la recherche basée par la géométrie de la comparaison dans le but de caractériser les différences géométriques. Ainsi, les travaux de cette thèse ont pu être particularisés par rapport aux nombreux travaux antérieurs sur la recherche des modèles 3D basée sur la géométrie.

Ensuite, la nature des informations requises à l'issue de la comparaison, ou le niveau de détails dans la représentation des différences, constitue le second facteur distinctif. Par exemple, la validation par comparaison de la traduction d'un modèle CAO entre deux formats CAO distincts ne requiert qu'un diagnostic de conformité. On ne cherche qu'à savoir si, oui ou non, la traduction a réussi selon certains critères de précision. Une description détaillée des différences entre le modèle original et le modèle migré – ce que certains outils logiciels comme *CompareVidia®* (CapVidia NV, 2010) dédiés à la validation de traduction de formats fournissent malgré tout – est accessoire. À l'opposé, comme dans le scénario de la transposition du changement entre modèles experts, la représentation détaillée et sémantiquement pertinente du point de vue d'un ingénieur est primordiale.

La pluralité des scénarios de comparaison des modèles CAO a finalement motivé le développement d'un cadre de modélisation de ces scénarios, présenté dans l'article de conférence du chapitre 5 (Brière-Côté, Rivest et Maranzana, 2011). Il s'agit là d'une contribution supplémentaire de cette thèse sur l'examen des scénarios de comparaison, qui permet de modéliser des scénarios individuels afin d'en supporter l'analyse.

9.1.4 Parallèle entre modèles CAO et modèles de logiciels

Lors de la phase méthodologique de la modélisation des différences géométriques entre deux modèles CAO, cette thèse a exploré de nouvelles solutions dans le domaine du développement logiciel en établissant un parallèle entre l'évolution de la définition géométrique d'un produit mécanique et l'évolution des codes sources et des modèles de logiciels en développement logiciel. Ce parallèle a ainsi permis de profiter des connaissances et des récents travaux en matière de gestion des versions et de modélisation des différences entre ces versions pour les transposer à la résolution du problème de la représentation des différences géométriques. Cette mise en relation de deux domaines de recherche distincts, mais visant des objectifs similaires constituent en quelque sorte une autre contribution de cette thèse au problème de la transposition du changement entre modèles géométriques.

Il en est surtout ressorti une approche pour la représentation des différences entre modèles géométriques, inspirée de l'approche pour la génération de modèles des différences proposée par Cicchetti, Di Ruscio et Pierantonio (2007) et telle qu'illustrée à la figure 6.4 et qui reproduite à la figure 9.2. Le modèle logiciel, tel le diagramme de classes qui sert en partie à structurer la programmation orientée-objet (Object Management Group, 2011), est un modèle de spécification, comme le modèle CAO, qui définit le produit ou le logiciel en développement avant que celui-ci ne soit réalisé (fabriqué ou programmé). Il a toutefois été considéré que la caractéristique principale du modèle logiciel, qui en fait un candidat idéal pour la représentation des différences comme le démontrent les développements en ce sens en génie logiciel, est qu'il est avant tout descriptif ou explicite, i.e., il représente uniquement l'objet modélisé dans son état final ou attendu. Ce n'est pas le cas du modèle CAO conventionnel représenté sous la forme procédurale, i.e., sous la forme d'une séquence d'opérations à caractère géométrique qui, prise hors du contexte de la construction même du modèle CAO, revêt peu ou aucune signification par rapport aux spécifications géométriques représentées.

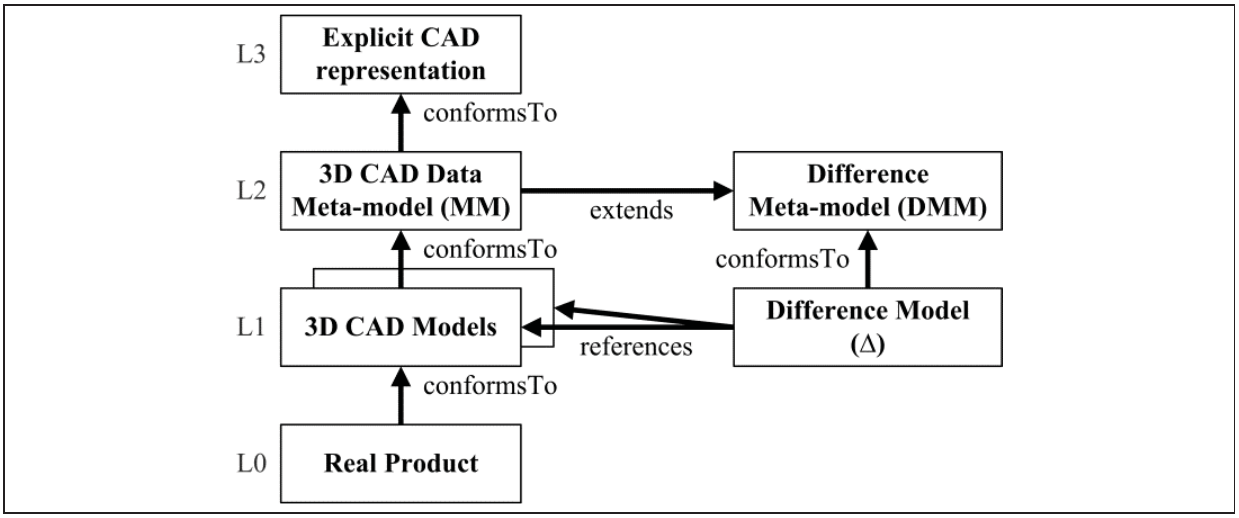


Figure 9.2 Architecture de représentation des différences entre modèles CAO

Ce constat, jumelé aux observations réalisées lors de l'évaluation des outils logiciels, a permis de statuer que, à l'instar de l'identification des différences entre modèles logiciels, le repérage et la représentation des différences entre modèles CAO dépendent à la base d'une représentation explicite des géométries et des données expertes qui la décrivent. Selon l’approche proposée, cette « explicitation » des modèles CAO se réalise ainsi grâce à un méta-modèle CAO – un modèle des modèles CAO – garantissant le caractère explicite de la représentation des données au sein de ces modèles afin qu’elles soient comparées.

9.2 Limitations et recommandations

La prochaine section apporte une contrepartie aux contributions de cette thèse en mettant en évidence certaines limitations de notre proposition. Ces limitations concernent avant tout les postulats émis en début de thèse ou lors de l’élaboration de la solution qui, malgré leur pertinence, peuvent générer un écart entre les concepts proposés par cette thèse et la réalité pratique de la modélisation CAO en développement de produits mécaniques. Afin d’ouvrir la voie à des solutions qui, on le souhaite, combleront ces écarts lors de travaux de recherche subséquents, cette section détaillera également des recommandations ciblées.

9.2.1 Comparaison géométrique par les frontières

Au sein de l’approche de comparaison proposée, cette thèse n’impose pas l’application d’un algorithme de calcul des différences géométriques bien précis : l’emphase est principalement mise sur la représentation de ces différences dans l’objectif de faciliter leur interprétation et leur assimilation par un ingénieur concepteur. Des exigences sont néanmoins formulées quant aux capacités d’un tel algorithme pour en assurer son adéquation avec une représentation des différences basée sur les contraintes géométriques explicites. Ces exigences sont :

- Doit pouvoir comparer des modèles détachés (voir postulat de la section 1.2.3);
- Doit comparer des géométries représentées par leurs frontières (B-Rep); et
- Doit produire des associations entre éléments topologiques (faces, arêtes et sommets) équivalents, ce malgré la présence probable d’une différence géométrique entre ces éléments.

Un exemple d’algorithme de calcul des différences géométriques applicable à l’approche proposée est d’ailleurs cité. En effet, l’algorithme implémenté par le logiciel *PTC*

CoCreate® Modeling PE (CoCreate Software GmbH, 2008) et dont le principe est décrit dans le brevet d'invention US 2010/0135535 (CoCreate Software GmbH et Gutierrez, 3 juin 2010) remplit les exigences de l'approche, en plus de présenter des résultats prometteurs au terme des évaluations logicielles (Brière-Côté, Rivest et Maranzana, 2013).

À ce stade-ci, il apparaît toutefois nécessaire de discuter des possibles limites d'un tel algorithme, et d'autres similaires basés sur la représentation B-Rep, dans la réalisation d'une comparaison de deux modèles CAO pour en repérer les différences du seul point de vue de la forme des pièces mécaniques comparées. On se réfère ainsi au second postulat de cette thèse décrite à la section qui statue comme étant sémantique toute différence entre deux modèles qui est fondamentalement géométrique, i.e., qui n'est pas issue de la représentation numérique des modèles CAO.

9.2.1.1 Représentation topologique de différences géométriques

Les évaluations logicielles résumées dans l'article du chapitre 4 (Brière-Côté, Rivest et Maranzana, 2013) ont démontré qu'il est difficile pour un algorithme de calcul des différences de clairement distinguer les cas de figure particuliers impliquant simultanément une équivalence géométrique et une différence topologique. Deux de ces cas de figure énumérés dans le tableau 4.3 comme les modifications 'A' et 'F' témoignent de la complexité de l'opération.

Dans un premier temps, la modification 'A' a pour résultat de rendre géométriquement coïncidentes deux faces qui, topologiquement, sont parfaitement distinctes. En fait, du point de vue des caractéristiques de forme, ces deux faces n'appartiennent tout simplement pas à la même caractéristique, comme le présente la figure 4.6. Ici, la différence topologique, théoriquement détectable grâce à l'orientation des faces qui sont opposées, doit idéalement outrepasser l'équivalence géométrique et indiquer la différence.

Un algorithme effectuant strictement la comparaison au niveau géométrique – comme le calcul de déviations entre géométries ou *point-to-part deviations*, méthode utilisée par des nombreux algorithmes testés – n'identifiera pas la différence étant données les géométries équivalentes des deux faces. L'algorithme recommandé du logiciel *PTC CoCreate® Modeling PE* (CoCreate Software GmbH, 2008) est parvenu à repérer la différence, mais pas à associer topologiquement les faces coïncidentes avec leurs faces correspondantes respectives. Utilisé pour la représentation des différences basée sur des contraintes géométriques, l'algorithme aurait causé une erreur dans la transposition du schéma de contraintes d'une version à l'autre en refusant de reproduire toute contrainte associée à ces faces.

L'exploitation de la définition explicite de caractéristiques de forme, similairement aux contraintes géométriques explicites, mais selon un niveau d'abstraction plus élevé, aurait probablement permis de détecter que les deux faces n'appartiennent pas à la même caractéristique. La disponibilité de définitions explicites de caractéristiques de forme au sein de modèles CAO constitue toutefois un obstacle tout aussi important – sinon, plus important – que la disponibilité des contraintes géométriques explicites.

Dans un second temps, la modification 'F' représente l'exemple typique d'une différence purement topologique. En effet, cette modification ne modifie en rien la géométrie de l'ouverture cylindrique présentée à la figure 4.7, puisque seule la subdivision en faces de la surface cylindrique est différente (les deux faces au sein du modèle original deviennent trois faces au sein du modèle modifié). Donc, lorsqu'il est question d'identifier seulement les différences au niveau de la forme – comme pour l'approche proposée dans cette thèse – la modification 'F' ne doit générer aucune différence.

Lors du test effectué sur le logiciel *PTC CoCreate® Modeling PE* (CoCreate Software GmbH, 2008), aucune différence n'a effectivement été identifiée. Toutefois, ce résultat n'est pas attribuable à l'algorithme de calcul des différences sous-jacent, mais plutôt au processus de traduction des formats CAO qui a permis d'importer un modèle STEP dans l'environnement de modélisation du logiciel. En effet, une fois importés, les modèles comparés ne présentaient plus aucune subdivision de la surface cylindrique, démontrant une altération imprévue.

Dans le cas de l'approche de comparaison proposée, un tel résultat est souhaitable. Un algorithme de prétraitement des représentations B-Rep à comparer, supprimant les subdivisions topologiques redondantes de faces de même géométrie à la manière du processus d'importation STEP du logiciel, serait sûrement utile pour garantir le repérage des différences strictement géométriques. Une précaution importante dans la conception d'un tel algorithme serait toutefois d'assurer l'intégrité et la cohérence des schémas de contraintes géométriques explicites associées aux faces et aux arêtes prétraitées.

Selon d'autres scénarios de comparaison, ce type de différences purement topologiques peut revêtir une signification et il serait cette fois malavisé de les négliger. C'est effectivement le cas lorsque les modèles CAO 3D sont utilisés pour la génération de maillages destinés aux analyses par éléments finis. Dans un tel scénario, la subdivision d'une face en plusieurs sous-faces ou d'une arête en plusieurs sous-arêtes de géométrie équivalente permet notamment d'isoler les zones où une charge mécanique est appliquée en vue de l'analyse des contraintes et des déformations de la pièce modélisée par éléments finis.

Sinon, la mise au point d'un algorithme précis de calcul des différences géométriques capable d'apparier un groupe d'éléments topologiques à un autre groupe entre les représentations B-Rep – i.e., associations N:M plutôt 1:1 – demeure à être réalisé. Avec un tel algorithme, les deux faces cylindriques au sein du modèle original seraient maintenant associées conjointement aux trois faces cylindriques correspondantes au sein du modèle modifié. L'intégrité de la topologie des modèles comparés serait alors préservée tout en fournissant l'information nécessaire à la transposition des contraintes géométriques.

9.2.1.2 Géométries complexes

La comparaison géométrique de modèles CAO incorporant des éléments géométriques complexes tels des courbes et des surfaces B-Spline ou NURBS comporte ses limitations.

Grâce à une méthode comme le calcul de déviations entre géométries, il est simple de détecter des écarts en plusieurs localisations entre un profil gauche original et un profil modifié. Dans un contexte d'inspection par ordinateur, la mesure de ce type d'écarts entre les deux profils – un profil usiné et un profil nominal – fournit un excellent indicatif de la conformité d'une pièce fabriquée et du procédé de fabrication utilisé. Toutefois, en contexte de conception et plus particulièrement de transposition de changement, une telle mesure entre deux versions d'un même profil nominal n'est d'aucune signification pour la compréhension et l'interprétation de l'évolution dans la définition géométrique.

Dans le cadre de cette thèse, il est donc nécessaire d'élever le niveau sémantique de la différence afin d'arriver à en décrire la source du point de vue des spécifications géométriques. L'approche proposée requiert d'abord la création d'associations entre les éléments géométriques et topologiques constituant les deux modèles CAO. Pour ce qui est des éléments géométriques complexes, i.e., ceux qui ne sont pas typés comme des éléments géométriques analytiques tels des plans, des cylindres, des sphères, etc., la création de ces associations peut être laborieuse.

La représentation B-Rep de tels éléments peut notamment constituer une source importante de variabilité topologique entre modèles CAO de formats CAO différents. De plus, dans le cas du test d'identification de différences entre deux surfaces complexes (modification 'J', tableau 4.3) effectué sur le logiciel *PTC CoCreate® Modeling PE* (CoCreate Software GmbH, 2008), la différence a bel et bien été identifiée, mais l'association entre les faces complexes a échoué. La mise au point d'un algorithme de calcul des différences géométriques précis produisant le plus fidèlement possible des associations entre les éléments topologiques de deux représentations B-Rep demeure donc à réaliser. L'association de plus de deux éléments topologiques correspondants entre les deux modèles comparés permettrait notamment d'associer des courbes et des surfaces complexes de géométrie équivalente, i.e., issus d'un même ensemble de spécifications géométriques, mais dont le découpage topologique diffère. Une telle capacité d'association multiple des éléments topologiques fut notamment observée au sein du logiciel *CompareVidia*® (CapVidia NV, 2010). Toutefois, lors de ces observations, la qualité de ces associations multiples n'était pas satisfaisante : des groupes de faces représentant notamment des congés et des arrondis étaient inutilement associés en groupe alors que l'association individuelle de ces faces était possible.

Puis, pour l'expression des différences entre deux courbes ou deux surfaces complexes en termes de contraintes géométriques, l'application de l'approche proposée dépend de la disponibilité de contraintes géométriques explicites définissant ce type de géométries au sein d'un modèle CAO. Théoriquement, ces contraintes géométriques existent, car elles incarnent normalement les conditions de continuité (coïncidence, tangence, etc.) de ces éléments complexes avec les éléments adjacents ou les positions explicites des nœuds. En fait, elles représentent les principales spécifications de forme complexe incluses à la définition géométrique d'une pièce et, donc, elles représentent la cible originelle des modifications d'ingénierie : rares sont les formes complexes de pièces mécaniques spécifiées en termes de points de contrôle et de pondérations. La question de la disponibilité des contraintes géométriques sous la forme explicite au sein des modèles CAO fait justement l'objet d'une discussion approfondie à la prochaine section.

9.2.2 Contraintes géométriques explicites

L'approche de comparaison géométrique de modèles CAO telle que proposée dans ces pages se base sur l'exploitation d'un schéma de contraintes géométriques explicites provenant d'un des deux modèles comparés et associé directement à la géométrie explicite représentée par les frontières (B-Rep). Ces contraintes géométriques explicites fournissent le point de référence de l'interprétation technique de la forme modélisée – ce qui fait du modèle géométrique le modèle de spécification d'une pièce mécanique – qui permet, après la caractérisation du delta géométrique, de mieux représenter l'évolution de la définition géométrique de la pièce. Par exemple, en examinant dorénavant la contrainte dimensionnelle spécifiant la distance entre deux faces d'un modèle, il devient possible de caractériser le déplacement relatif d'une de ces faces par rapport à l'autre au sein d'une nouvelle version du même modèle non seulement comme une différence de forme, mais également comme une différence dimensionnelle au sein des spécifications.

Une telle proposition repose toutefois sur la prémisse que, dans la pratique courante en CAO, les modèles de pièces mécaniques comprennent généralement ces schémas de contraintes géométriques explicites. La disponibilité réelle de ces schémas, surtout sous la forme explicite, peut raisonnablement être remise en question et constitue l'une des limitations de l'approche de comparaison basée sur les contraintes géométriques.

Le principal obstacle réside dans le fait que, traditionnellement, les modèles CAO sont générés et représentés sous la forme procédurale ou implicite. Par contre, de nouvelles variantes des logiciels de modélisation CAO viennent graduellement changer le paradigme de la modélisation procédurale en offrant des outils de modélisation directe ou explicite, ce qui ne peut toutefois pas garantir la présence de schéma de contraintes géométriques explicites. Puis, on peut envisager l'exploitation des données CAO exprimées dans un formalisme très similaires à celui des contraintes géométriques, soit les cotes dimensionnelles et le tolérancement géométrique, mais certaines limites devront être considérées. Ces trois aspects de la disponibilité limitée des contraintes géométriques explicites au sein des modèles CAO comparés sont abordés dans les prochains paragraphes.

9.2.2.1 Contraintes géométriques implicites

Une première question pertinente demande s'il existe suffisamment de données CAO disponibles sous la forme de contraintes géométriques explicites au sein des modèles CAO basés sur la paramétrisation de caractéristiques de modélisation (*parametric feature-based*), largement utilisés dans la pratique industrielle. En réalité, les contraintes géométriques explicites et les paramètres qui s'y rattachent sont parmi les notions constituantes de la modélisation CAO procédurale. Par exemple, la création d'esquisses en deux dimensions ainsi que l'opération de « contraindre » ces esquisses à l'aide de contraintes géométriques, préalables requis à la définition d'opérations de modélisation 3D telles le balayage ou la révolution, constitue de la modélisation explicite.

Toutefois, une bonne partie des contraintes géométriques au sein des modèles CAO procéduraux demeurent implicites aux définitions des caractéristiques de modélisation 3D, tel qu'exposé par Kim *et al.* (2008). Les contraintes géométriques définies au sein des esquisses 2D demeurent explicites à ce niveau de modélisation, mais doivent être considérées comme implicites au niveau de la géométrie résultante puisqu'internes à la définition des opérations de modélisation. De plus, lorsqu'elles sont bel et bien explicites, les contraintes géométriques peuvent être associées malgré tout à des éléments géométriques intermédiaires et donc n'avoir aucune signification dans la spécification de la géométrie explicite finale. Un élément géométrique intermédiaire est un élément géométrique généré par une opération de modélisation qui est modifié, remplacé ou supprimé de la géométrie résultante par une opération subséquente dans la séquence d'opérations, ne jouant donc un rôle que dans le processus de construction du modèle.

La figure 9.3 représente une catégorisation des éléments des modèles CAO selon leur niveau d'abstraction par rapport à la forme modélisée. On peut également l'interpréter comme une représentation des différents niveaux de sémantique de la forme résultante dans un contexte de développement de produit mécanique. La modélisation et la représentation procédurales implicites opèrent principalement aux niveaux des caractéristiques de forme et technologiques qui, une fois définies et « compilées », génèrent les éléments résultants aux niveaux géométriques et topologiques. On peut qualifier cette approche d'élaboration de la forme depuis des éléments complexes vers des éléments de granularité plus fine.

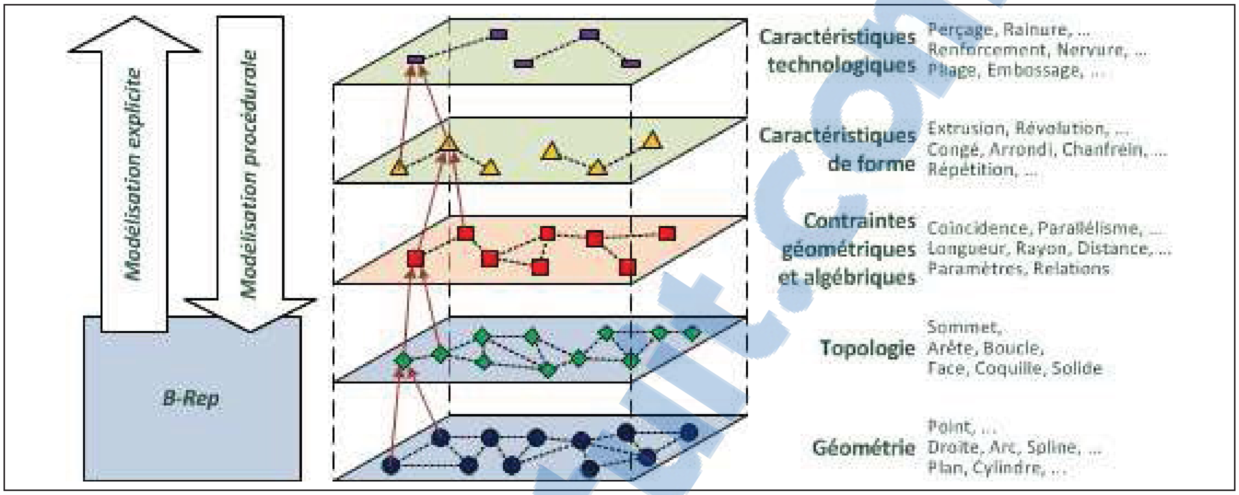


Figure 9.3 Catégorisation des éléments d'un modèle CAO selon les niveaux d'abstraction de la forme
Inspirée de Brunetti et Grimm (2005)

On peut donc partir de la prémisse que les données CAO définissant les contraintes géométriques explicites de la géométrie résultante existent au sein des modèles procéduraux, mais qu'elles demeurent implicites ou « encapsulées » au sein de la représentation procédurale. L'approche de comparaison basée sur les contraintes géométriques proposée dans cette thèse pourrait s'appliquer plus commodément moyennant des algorithmes permettant le calcul séquencé des caractéristiques de modélisation d'une représentation procédurale afin d'en extraire non seulement une géométrie B-Rep résultante, mais également un schéma associé de contraintes géométriques explicites issu des définitions implicites de ces caractéristiques. L'objectif de tels algorithmes seraient en quelque sorte de « consommer » l'arbre de construction d'un modèle CAO et de rendre ce dernier explicite en en conservant le plus de spécifications de forme possible aux niveaux des contraintes géométriques et, également, des caractéristiques de forme et technologiques.

Il existe des algorithmes en CAO proposant des objectifs similaires aux algorithmes recommandés ici. Par exemple, au sein de nombreux logiciels de modélisation CAO 3D tels *CATIA® V5* (Dassault Systèmes, 2007), *SolidWorks®* (Dassault Systèmes, 2010) ou *Solid Edge® ST* (Siemens PLM Software inc., 2009b), des algorithmes permettent d'extraire des contraintes géométriques définies au sein d'esquisses 2D – donc, implicites aux caractéristiques de modélisation – afin de les incorporer avec liaison aux mises en plan de la géométrie 3D. Également, le prototype logiciel *Inventor Fusion Technology* d'Autodesk inc. (2010b) permet la conversion d'un modèle CAO procédurale en un modèle CAO explicite tout en conservant des regroupements de faces constituant certaines caractéristiques de forme. Finalement, le logiciel *Solid Edge® ST4* (Siemens PLM Software inc., 2011b) permet de convertir en cours de modélisation un modèle représenté de manière procédurale en un modèle explicite tout en conservant, en association avec la géométrie 3D, des contraintes géométriques implicites et explicites, ainsi que des caractéristiques de forme.

L'utilisation d'algorithmes de traitement a posteriori des modèles CAO procéduraux présente malgré tout l'inconvénient de traiter et d'extraire sans discrimination des données CAO afin de les exploiter ultérieurement lors de prises de décision alors que certaines de ces données peuvent n'incarner aucune spécification dans la définition géométrique finale d'un composant mécanique. En effet, par exemple, une contrainte géométrique insérée dans une esquisse 2D avec pour unique rôle de compléter l'état « entièrement contraint » de cette esquisse n'aurait aucune signification au niveau des spécifications géométriques une fois associée à la géométrie 3D résultante. Transposée vers une version évoluée de la géométrie à la suite d'une comparaison selon l'approche proposée par cette thèse, l'équivalence ou la différence associée à cette contrainte pourrait induire une interprétation erronée du changement au niveau de la définition géométrique de la pièce.

9.2.2.2 Modélisation CAO explicite en trois dimensions

L'émergence de la modélisation explicite dans la pratique usuelle en CAO contribuera certainement à augmenter le nombre de modèles CAO de pièces mécaniques incorporant directement des contraintes géométriques explicites associées à la géométrie 3D. Tel qu'illustrée à la figure 9.3, l'approche de la modélisation explicite est à l'opposé de la modélisation procédurale. En effet, la modélisation explicite est davantage axée sur l'abstraction de la forme : elle se base d'abord sur la création et l'édition de la géométrie 3D représentée par les frontières (B-Rep) qui est par la suite enrichie par l'addition de contraintes géométriques et la définition de caractéristiques de forme et technologiques.

Les plus récents logiciels de modélisation CAO explicite semblent avant tout mettre l'emphase sur des aspects tels l'opérabilité et l'interopérabilité du point de vue de l'utilisateur. On recherche ainsi à contrer la rigidité de la modélisation par caractéristiques paramétrées durant les premières phases de conception et en contexte de collaboration, i.e., en laissant le plus de liberté possible aux concepteurs dans leurs activités de conception.

Selon ce point de vue, toutefois, l'addition de contraintes géométriques à un modèle CAO, étape pourtant nécessaire à la définition des spécifications géométriques d'une pièce, peut être considérée comme étant contre-productive à l'activité de modélisation. Des contraintes géométriques ajoutées trop tôt durant la phase de conception peuvent être considérées comme des obstacles à la conception. Aussi, compléter la définition détaillée de la géométrie 3D grâce aux contraintes géométriques pourrait être considéré comme une étape redondante si les spécifications dimensionnelles sont normalement définies lors d'une mise en plan subséquente.

Il demeure qu'une plus grande implantation de la modélisation CAO explicite dans la pratique industrielle est à envisager. En développement de produit, l'activité de modélisation est intimement liée à l'activité de conception qui exige un contrôle de la définition géométrique du produit qui croit au cours de la phase d'acquisition et qui culmine à la phase de modification (figure 1.1). Pour y parvenir, il faudra élaborer de nouvelles procédures de modélisation qui allieront rigueur, efficacité et flexibilité. Conséquemment, les contraintes géométriques explicites et, éventuellement, les caractéristiques de formes et les caractéristiques technologiques explicites, seront davantage utilisées pour définir des spécifications géométriques dès les premières phases de modélisation CAO et ainsi réguler les modifications géométriques selon l'évolution de la définition de la pièce.

9.2.2.3 Cotes dimensionnelles et tolérances géométriques

Cette thèse met l'emphase sur les données CAO spécifiques que sont les contraintes géométriques explicites, car elles incarnent une interprétation technique initiale d'une forme permettant par la suite à l'approche de comparaison proposée d'élever le niveau d'abstraction sémantique d’une représentation des différences entre deux modèles. Toutefois, dans la pratique courante de la CAO, selon les procédures de modélisation en place, il peut ne pas y avoir de corrélation directe entre les contraintes géométriques au sein d’un modèle CAO et les spécifications géométriques et dimensionnelles d’une pièce. En d’autres mots, un schéma de contraintes géométriques peut parfois évoquer davantage un processus de construction d’un modèle que la définition géométrique d’un produit.

L’approche émergente en CAO 3D de la définition par modélisation, ou *model-based definition* (MBD), présente le potentiel de résoudre cette limitation. L’approche de MBD implique l’ajout de dimensions et tolérances géométriques directement au modèle CAO 3D, qui devient alors un modèle 3D annoté ou enrichi qui peut dorénavant être exploité, non seulement à la construction de la maquette numérique, mais également pour la fabrication et l’inspection des composants (Quintana, 2011; Quintana, Rivest et Pellerin, 2012). Lorsqu’exprimées selon des formalismes tels celui définie par la norme ASME Y14.5 (American Society of Mechanical Engineers, 2009), les annotations 3D du modèle enrichi, communément appelées *product manufacturing information* (PMI) dans l’industrie, s’associent à la géométrie d’une pièce au même niveau sémantique que les contraintes géométriques et définissent explicitement cette géométrie.

Les PMI représentent donc une autre source potentielle de relations géométriques explicites exploitables par l’approche de comparaison proposée. Une différence repérée entre deux modèles CAO 3D, dont l’un d’eux est enrichi par ces annotations 3D, pourrait alors entre représentée sous la forme d’une tolérance géométrique invalidée (ex. : perpendicularité, parallélisme, symétrie, etc.) ou d’une dimension dont la valeur a changé (ex. : distance, rayon, diamètre, etc.). Il est important de noter que, dans un tel cas de figure, les intervalles de tolérance dimensionnelle et géométrique représentés par ces annotations 3D seraient écartés de la représentation des différences de forme (« *Form* ») entre les modèles CAO, puisque leur fonction réside plutôt dans la spécification des ajustements (« *Fit* »).

L'exploitabilité des PMI pour la comparaison des modèles CAO selon l'approche proposée dépend toutefois de la nature et de la complexité de ces annotations 3D qui peuvent différer selon les formats de données CAO, même si elles se basent toutes sur un même formalisme. Comme l'expliquent Zuray et Delaunay (2009), les données PMI peuvent être représentées au sein du modèle CAO enrichi selon quatre niveaux de sémantique tels qu'illustrés à la figure 9.4 :

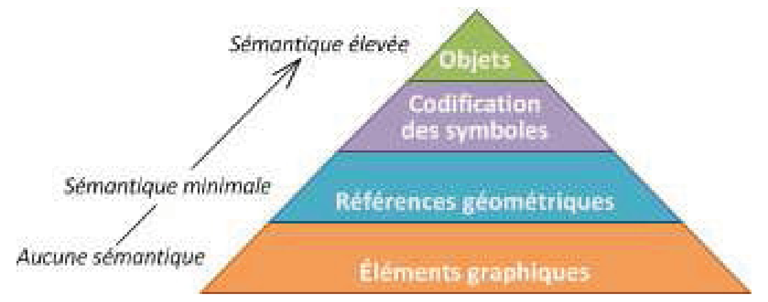


Figure 9.4 Niveaux de représentation sémantique des annotations 3D du modèle enrichi Inspirée de Zuray et Delaunay (2009)

- Représentation par éléments graphiques : les annotations 3D sont des lignes et des arcs représentant graphiquement des cotes et des tolérances géométriques sans aucun lien à la géométrie 3D;
- Avec références géométriques : les annotations 3D sont des groupes d'éléments graphiques dont la position dans l'espace de modélisation – et donc en rapport à la géométrie modélisée – est sauvegardée;
- Par codification des symboles : le formalisme de cotation dimensionnelle et de tolérancement géométrique est sauvegardé sous la forme de codes, ce qui permet en plus d'interpréter, non sans risque d'erreur, la nature et le contenu des annotations; et
- Par objets typés : le formalisme de cotation dimensionnelle et de tolérancement géométrique est inclus intégralement à la représentation numérique du modèle CAO.

Afin de pouvoir exploiter les PMI pour la comparaison des modèles CAO en vue de la transposition du changement, il est alors important d'établir des exigences minimales par rapport à la représentation de ces annotations 3D. Principalement, pour substituer les contraintes géométriques explicites, les PMI devront être clairement typés et associés sans ambiguïté aux éléments de la représentation B-Rep. La représentation des PMI par objets typés au sein des modèles CAO 3D enrichis serait donc nécessaire.

9.2.3 Synthèse des recommandations

À la lumière des limitations discutées précédemment, il est possible d'identifier trois principales avenues d'amélioration de l'approche de comparaison des modèles CAO proposée dans cette thèse. Ces avenues d'amélioration et les recommandations correspondantes sont résumées au sein du tableau 9.1.

Tableau 9.1 Synthèse des recommandations

Avenue d'amélioration	**Recommandations**
Calcul des différences géométriques	Mettre au point un meilleur algorithme de calcul des différences géométriques produisant des associations entre éléments topologiques de représentation B-Rep : • Inclure certaines informations topologiques à la comparaison géométrique, comme l'orientation des faces, afin de détecter les différences topologiques significatives; • Permettre l'association de plus de deux éléments topologiques entre les modèles de référence et cible afin de négliger les subdivisions topologiques non significatives du point de vue de la définition géométrique du produit; • Développer un algorithme de prétraitement des modèles CAO à comparer supprimant les subdivisions topologiques non significatives du point de vue de la définition géométrique du produit.
Disponibilité des contraintes géométriques explicites	Extraire l'ensemble des contraintes géométriques au sein d'un modèle CAO afin de les rendre explicites et associées à la géométrie 3D et, donc, exploitable par l'approche proposée : • Contraintes géométriques implicites à la définition des caractéristiques de modélisation d'un modèles CAO procédural; • Contraintes géométriques explicites au sein d'esquisses 2D; • Contraintes géométriques utilisées dans la définition des conditions de continuité des courbes et des surfaces complexes; • Annotations 3D d'un modèle CAO enrichi (MBD) représentant des spécifications dimensionnelles et de tolérancement géométrique.
	Développer des procédures de modélisation CAO explicite valorisant la définition de contraintes géométriques 3D pour : • Définir explicitement les spécifications géométriques de la pièce modélisée, telles les cotes fonctionnelles; • Réguler les modifications apportées aux modèles CAO afin qu'elles respectent l'évolution réelle de la géométrie du produit.
Expressivité de la représentation des différences géométriques	Exploiter une représentation explicite des caractéristiques de forme et technologiques pour une meilleure caractérisation des différences : • Développer un algorithme rendant explicites les caractéristiques de modélisation au sein d'un modèle CAO procédural; • Développer des procédures de modélisation CAO explicite incluant la définition des caractéristiques de forme et technologiques comme l'agrégation d'éléments de forme et de contraintes géométriques.

CONCLUSION

Un aspect particulier du PLM repose sur l'exigence que tout collaborateur au cycle de vie du produit devrait toujours avoir accès sans difficulté à l'information adéquate dont il a besoin pour réaliser sa tâche. Cette thèse explore les mécanismes de l'évolution de la définition du produit, plus spécifiquement la transposition du changement géométrique entre des modèles experts. Conséquemment, l'approche de comparaison géométrique des modèles CAO 3D proposée dans cette thèse vise à procurer aux ingénieurs de conception et des méthodes, notamment, une représentation significative et fonctionnelle des différences entre un modèle de référence incarnant une nouvelle version de la géométrie d'une pièce et un modèle cible auquel l'évolution doit se propager.

Les différences géométriques calculées au sein de la représentation par les frontières (B-Rep) des modèles CAO par certains algorithmes existants peuvent dorénavant être exprimées selon le niveau d'abstraction de la forme sémantiquement plus riche des contraintes géométriques et des paramètres de conception. Il s'agit là du niveau d'abstraction le plus familier pour tout ingénieur mécanicien, soit celui sur lequel sont naturellement définies et sur lequel évoluent les spécifications de forme d'une pièce.

Représentées sous leur forme explicite, les contraintes géométriques associées à la géométrie du modèle expert original, offrant une interprétation spécifique de la géométrie d'une pièce selon le point de vue de la discipline cible, sont transposées vers la géométrie d'un modèle représentant l'état modifié de la pièce et sont réévaluées en fonction de cette géométrie évoluée. Ce mécanisme permet ainsi d'exprimer le delta géométrique caractérisant l'évolution – i.e., les résultats de la comparaison géométrique – en fonction du maintien ou de l'abandon des contraintes géométriques originales et/ou de la variation de contraintes dimensionnelles. Contrairement à une représentation primitive des différences basée, par exemple, sur l'état des faces et des arêtes entre deux représentations B-Rep, l'approche proposée basée sur les contraintes géométriques vise à mieux supporter l'ingénieur dans l'analyse de l'impact de l'évolution géométrique d'une pièce sur le modèle expert dont il a la responsabilité, valider l'évolution du point de vue de sa discipline et, ultimement, le mettre à jour de manière efficace et cohérente.

L'approche de comparaison géométrique proposée dans cette thèse combine l'exploitation d'algorithmes de calcul des différences géométriques, générant des associations entre les éléments de deux représentations B-Rep, avec une nouvelle approche de modélisation des différences inspirée du domaine de la conception logicielle basée sur la modélisation (MDSE). Cette approche fonctionnelle de modélisation des différences, basée sur la méta-modélisation des modèles CAO, veut permettre la mise en référence individuelle et la manipulation d'éléments de modèles CAO hétérogènes tels les éléments topologiques et les contraintes géométriques explicites, et ce pendant et après la comparaison. On établit ainsi de nouvelles assises pour un système de gestion des versions des modèles CAO où, comme au sein des systèmes de gestions des versions des codes sources, il deviendrait possible de formaliser numériquement l'évolution des modèles du point de vue spécifique des spécifications géométriques, et non des opérations de modélisation, et d'effectuer des opérations telles la combinaison et la fusion de versions.

Nos travaux de recherche ont notamment démontré que la comparaison de paires de modèles CAO 3D afin d'identifier dans les détails les différences géométriques les distinguant n'attire pas beaucoup d'attention au sein des domaines de recherche de la CAO et du PLM. En tout cas, ce sujet n'attire pas autant l'attention des chercheurs que le sujet analogue de la recherche des modèles 3D basée sur la géométrie. Ainsi, en examinant de manière concourante l'ensemble des scénarios, des outils logiciels et des méthodes de calcul impliquant la comparaison des modèles CAO, cette thèse étaie le potentiel de ce domaine de recherche pour répondre aux défis actuels de la gestion du cycle de vie du produit et encourage les nouveaux développements.

Conjointement avec la synthèse de l'état de l'art, les essais logiciels ont permis d'identifier les divers domaines d'application de la comparaison et de décrire les facteurs permettant de les distinguer les uns des autres, facteurs tels la fonction de la comparaison dans un problème donné. En recherchant les caractéristiques des meilleurs algorithmes d'identification des différences existants pouvant contribuer efficacement au problème de la transposition de changement, l'importante variabilité observée parmi ces algorithmes et, conséquemment, parmi les résultats générés pour les outils logiciels implémentant ces algorithmes, a mené au constat qu'aucune solution de comparaison des modèles CAO ne peut être utilisée avantageusement à n'importe scénario sans égard aux particularités de ce dernier. Voilà pourquoi il a été jugé adéquat de proposer un cadre de représentation des scénarios de comparaison des modèles CAO et ainsi mieux supporter le choix ou la conception de solutions dans ce domaine émergeant.

En complément au scénario de la transposition du changement, la comparaison de modèles CAO et la caractérisation de leurs différences proposées dans nos travaux permet notamment de fournir à des concepteurs mécaniques une représentation intuitive du delta distinguant un modèle de référence et des modèles similaires repérés grâce aux engins de recherche basée sur la géométrie. Effectuée par paires, notre approche de caractérisation permet l'expression des différences en relation avec les contraintes géométriques explicites du modèle de référence qui incarnent les spécifications géométriques recherchées. Dans un contexte de réutilisation des modèles en conception, elle permet donc de mieux appuyer l'étude des modèles candidats et la sélection du meilleur modèle.

En plus de contribuer à l'avancement des techniques de calcul et de représentation des différences entre modèles CAO de produits mécaniques, cette thèse constitue en quelque sorte un assentiment à la modélisation CAO explicite. L'étude des techniques de comparaison existantes, selon qu'elles opèrent sur des représentations CAO 3D procédurales ou explicites, a permis notamment d'observer la faible corrélation entre la représentation d'un modèle sous la forme d'un historique de construction et les réelles spécifications d'ingénierie dont il est le médium. En d'autres mots, parce qu'il présente exclusivement le résultat de la modélisation, l'usage du modèle CAO explicite est plus pertinent dans l'activité de spécification que son équivalent procédural, davantage orienté vers l'activité de modélisation. Sans désavouer le paradigme de la modélisation CAO par caractéristiques paramétrées fortement répandu dans la pratique courante, il apparaît opportun du point de vue de cette thèse que des efforts de recherche soient dirigés vers l'enrichissement des représentations CAO explicites pour perfectionner, notamment, la représentation des caractéristiques technologiques, et vers l'élaboration de procédures de modélisation explicite efficaces qui viendraient appuyer, éventuellement, les développements en définition de produit basée sur les modèles 3D (MBD).

ANNEXE I

INFORMATIONS RELATIVES À LA PUBLICATION D'ARTICLES CONSTITUANT CETTE THÈSE

Tableau-A I-1 Sommaire de la soumission du premier article de journal (chapitre 3)

Journal:	Computer-Aided Design and Application (CAD&A)
Titre:	Comparing 3D CAD Models: Uses, Methods, Tools and Perspectives
Auteur(s):	Antoine Brière-Côté, Louis Rivest, Roland Maranzana
Date de soumission:	28 septembre 2011
Date de première révision:	23 janvier 2012
Publication	Volume 9, Numéro 6, pp. 771-794. (2012)

Tableau-A I-2 Sommaire de la soumission du second article de journal (chapitre 4)

Journal:	Computer-Aided Design and Application (CAD&A)
Titre:	3D CAD Model Comparison: An Evaluation of Model Difference Identification Technologies
Auteur(s):	Antoine Brière-Côté, Louis Rivest, Roland Maranzana
Date de soumission:	3 mars 2012
Date de première révision:	1 octobre 2012
Publication	Volume 10, Numéro 2, pp. 173-195. (2013)

Tableau-A I-3 Sommaire de la soumission du premier article de conférence (chapitre 5)

Conférence:	8th International Conference on Product Lifecycle Management PLM11 Eindhoven, Pays-Bas, 11 au 13 juillet 2011
Titre:	A three-step approach for structuring 3D CAD model comparison scenarios
Auteur(s):	Antoine Brière-Côté, Louis Rivest, Roland Maranzana
Date de soumission:	28 février 2011
Date d'acceptation:	22 avril 2011
Numéro de référence	Paper #141

Tableau-A I-4 Sommaire de la soumission du second article de conférence (chapitre 8)

Conférence:	9th International Conference on Product Lifecycle Management PLM12 Montréal, Canada, 9 au 11 juillet 2012
Titre:	Integration of explicit geometric constraints in the comparison of 3D CAD models for part design reuse
Auteur(s):	Antoine Brière-Côté, Louis Rivest, Roland Maranzana
Date de soumission:	10 février 2012
Date d'acceptation:	3 avril 2012
Numéro de référence	Paper #118

ANNEXE II

EXTRAITS DE LA VALIDATION DES ALGORITHMES DE TRANSPOSITION DES ÉLÉMENTS DE FORME AUXILIAIRES

Tableau-A II-1 Légende des contraintes utilisée pour les schémas de validation

Icône	**Désignation selon différentes sources**		
	Bettig et Shah (2001)	**STEP Part 108** (International Organisation for Standardization, 2005)	**SolidWorks® 2010** (Dassault Systèmes, 2010)
⚓	FIXED	fixed_element_geometric_constraint	Fix
—	FIXED_IN_X_DIR	--	Horizontal
|	FIXED_IN_Y_DIR	--	Vertical
∥	SAME_DIRECTION	parallel_geometric_constraint	Parallel
⊥	RIGHT_ANGLE	perpendicular_geometric_constraint	Perpendicular
∠	ANGLE	agc_with_dimension	Angular dimension
⤧	DISTANCE + SAME_DIRECTION	pgc_with_dimension	Parallel dimension
⤢	DISTANCE	pdgc_with_dimension cdgc_with_dimension	Dimension
•—	--	--	Endpoint
—•—	--	--	Midpoint
⊕ ou ⊙	COINCIDENT INCIDENT	incidence_geometric_constraint	Coincident Collinear
✕	--	--	Intersection
◎	COINCIDENT_ LOCATING_POINT	coaxial_geometric_constraint	Concentric
Ø	--	--	Diameter
R	RADIUS	rgc_with_dimension	Radius
⌒	OFFSET_CURVE	parallel_offset_geometric_constraint	Offset entities
⚲	TANGENT	tangent_geometric_constraint	Tangent
=	--	--	Equal
≡	SYMMETRIC	symmetry_geometric_constraint	Symmetric
🔒	*Implicit relationship (Section 7.6.3)*		

Tableau-A II-2 Légende des éléments utilisée pour les schémas de validation

Éléments de modèle		**Résultats de transposition**	
Icône	**Description**	**Icône**	**Description**
LIN2	Élément de forme physique *Physical shape element* (PSE)	✓	Contrainte préservée / *Maintained constraint* Élément de forme associé entre modèle / *Mapped shape element*
CIR4	Élément de forme physique modifié / *Modified PSE*	⚠	Contrainte modifiée ou enfreinte / *Modified or violated constraint*
LIN3	Élément de forme unique *Unique PSE*	✖	Contrainte orpheline / *Unique constraint* Élément de forme unique / *Unique shape element*
DP7	Paramètre *Design parameter*	ⓘ	Contrainte imposée / *Enforced constraint*
LN1	Élément de forme auxiliaire *Auxiliary shape element* (ASE)	?	Élément de forme auxiliaire non-transposé / *Not transposed ASE*
←	Élément de référence *Reference element*		
→	Élément contraint *Constrained element*		
x.x ■	Contrainte élémentaire *Elementary constraint*		

A II.1 Premier cas de figure

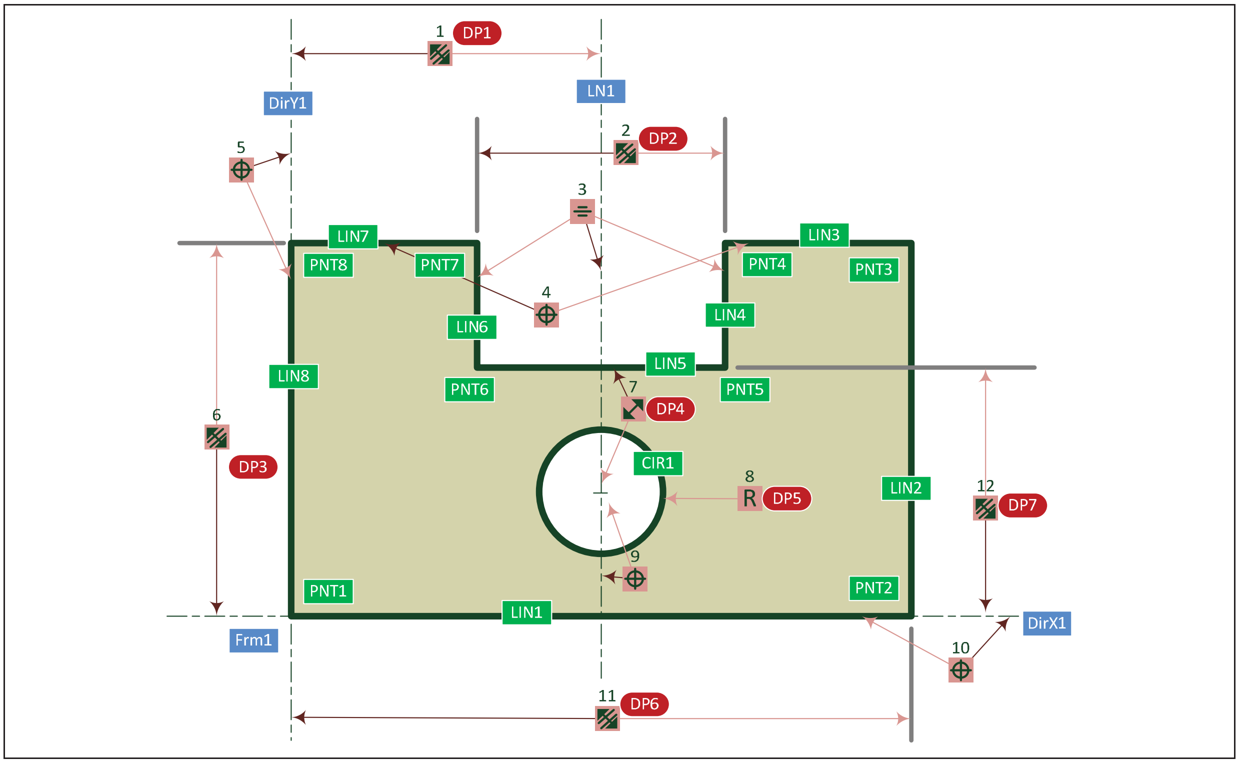


Figure A II-1 Premier cas de figure : Forme de référence originale et schéma de contraintes géométriques explicites

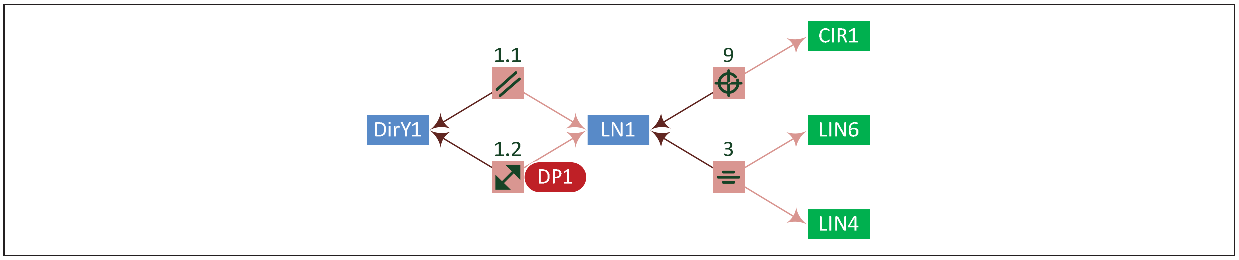


Figure A II-2 Premier cas de figure : Schéma original de contraintes relatives à l'élément de forme auxiliaire à transposer

A II.1.1 Premier cas de figure : Première forme cible modifiée

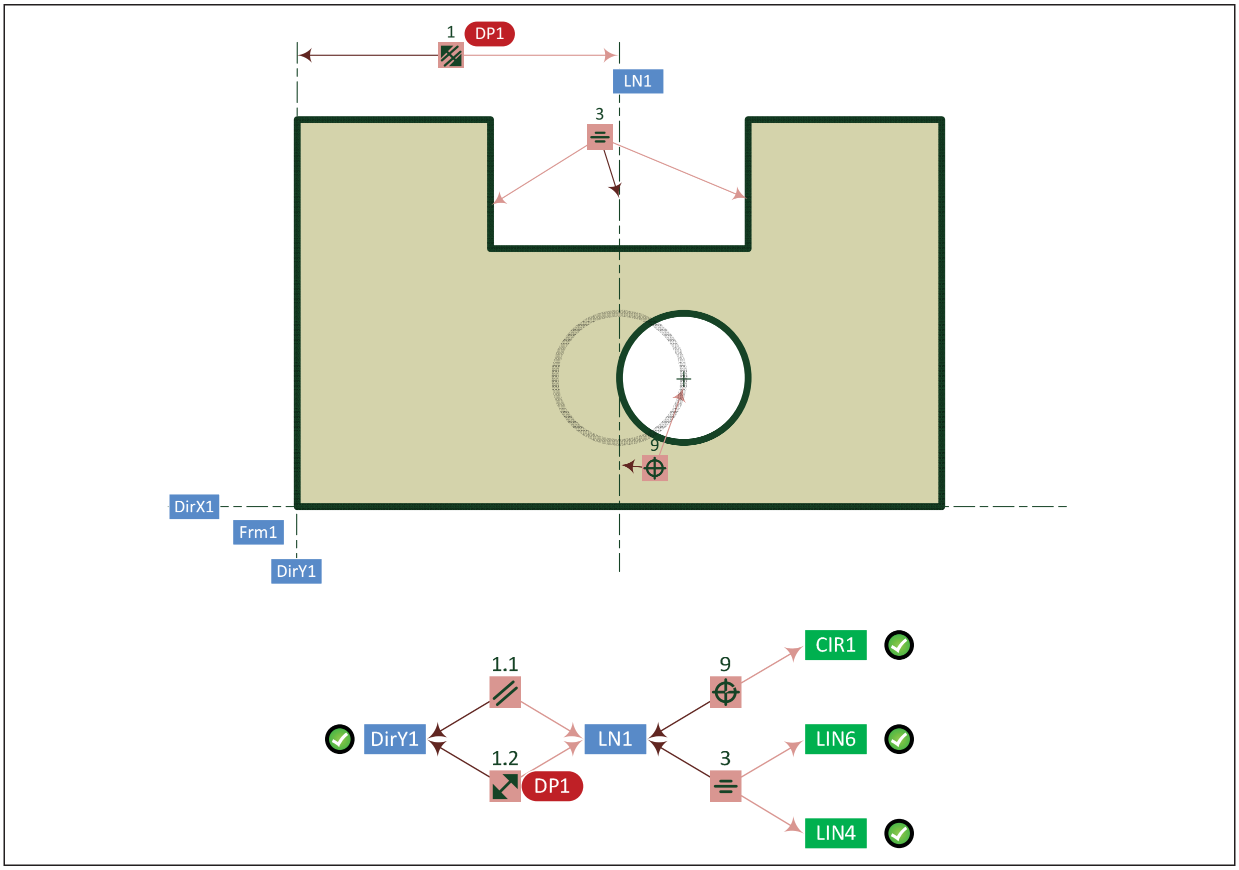


Figure A II-3 Premier cas de figure : Première forme cible modifiée et état du schéma original de contraintes en fonction des éléments de forme modifiés

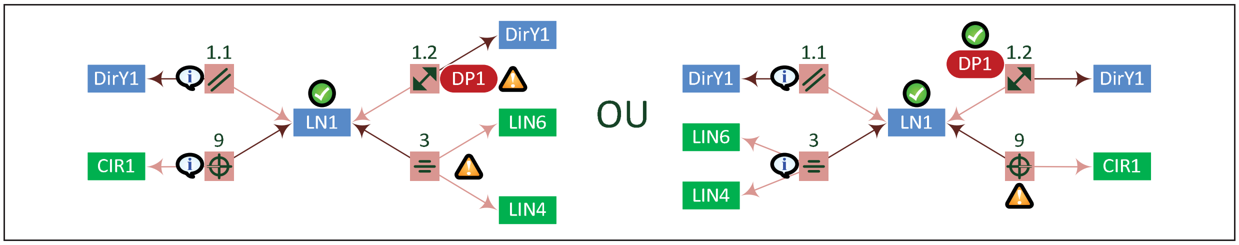


Figure A II-4 Premier cas de figure : Transposition de l'élément de forme auxiliaire vers la première forme cible

A II.1.2 Premier cas de figure : Deuxième forme cible modifiée

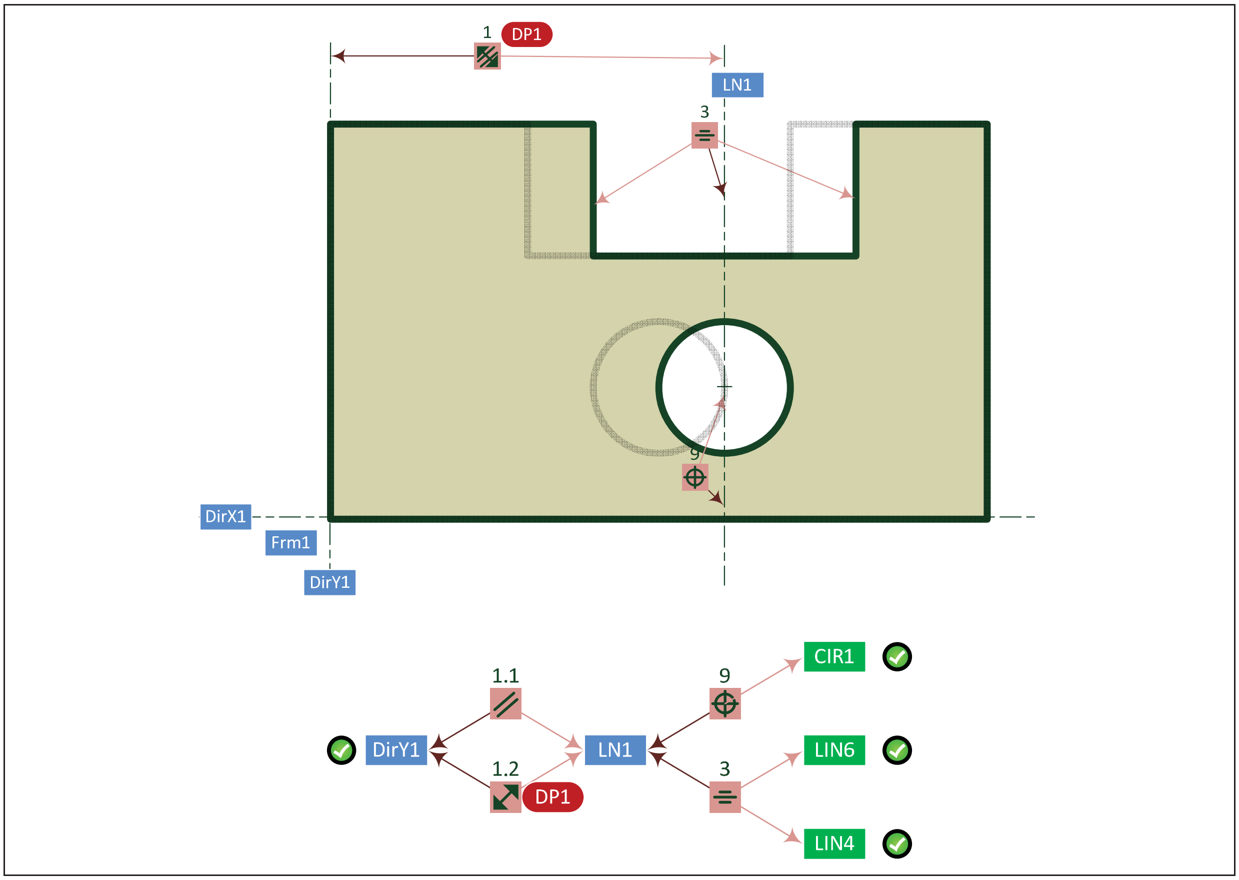


Figure A II-5 Premier cas de figure : Deuxième forme cible modifiée et état du schéma original de contraintes en fonction des éléments de forme modifiés

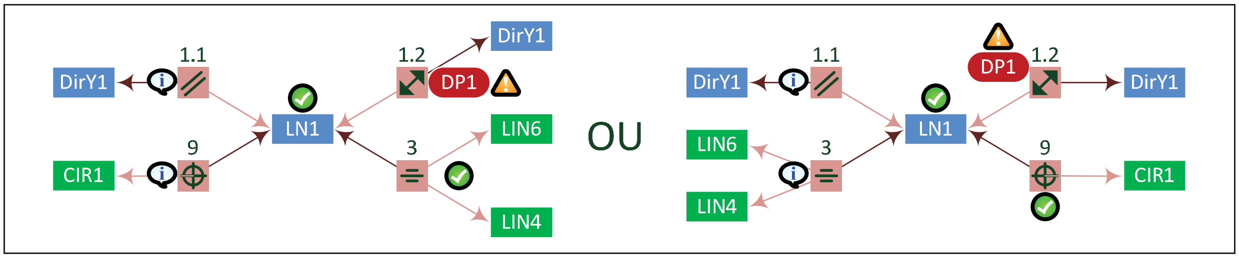


Figure A II-6 Premier cas de figure : Transposition de l'élément de forme auxiliaire vers la deuxième forme cible

A II.1.3 Premier cas de figure : Troisième forme cible modifiée

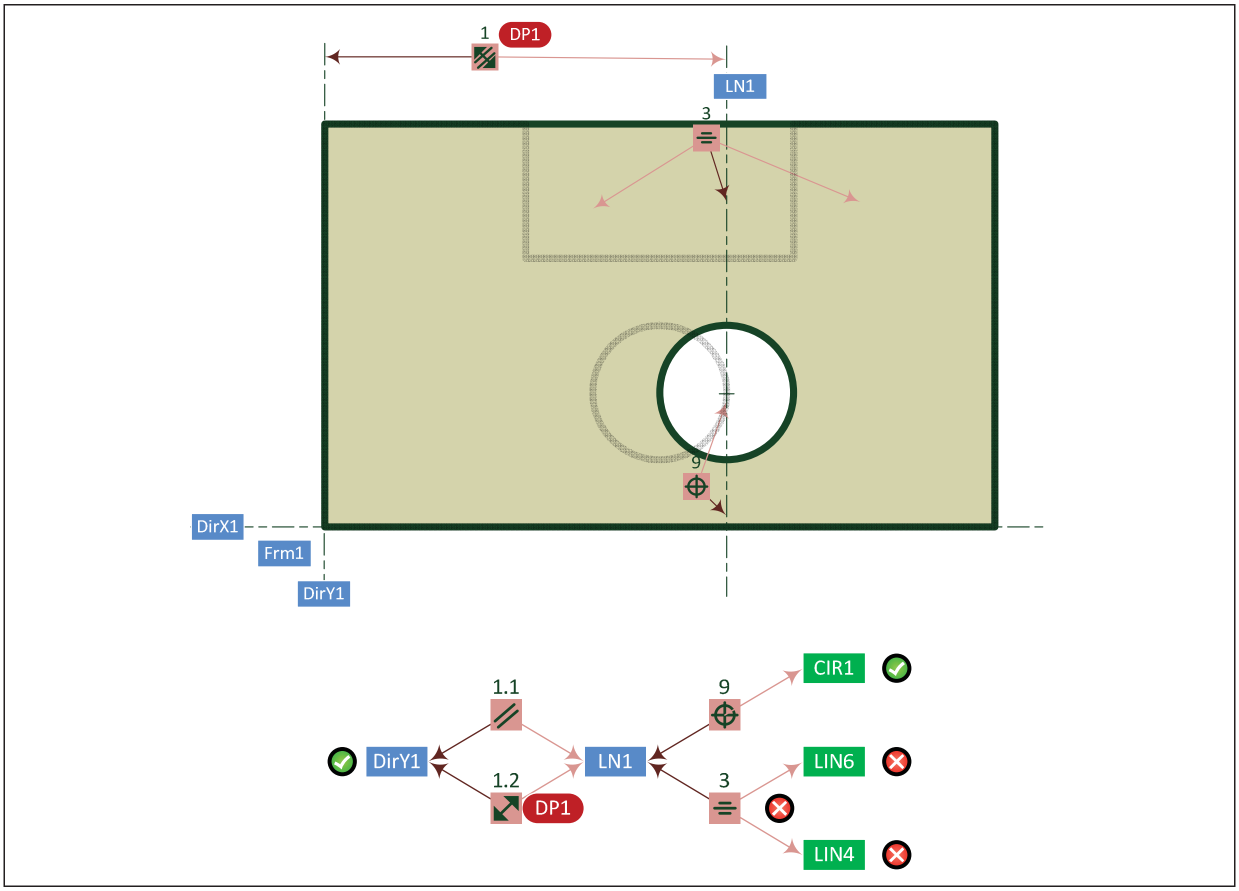


Figure A II-7 Premier cas de figure : Troisième forme cible modifiée et état du schéma original de contraintes en fonction des éléments de forme modifiés

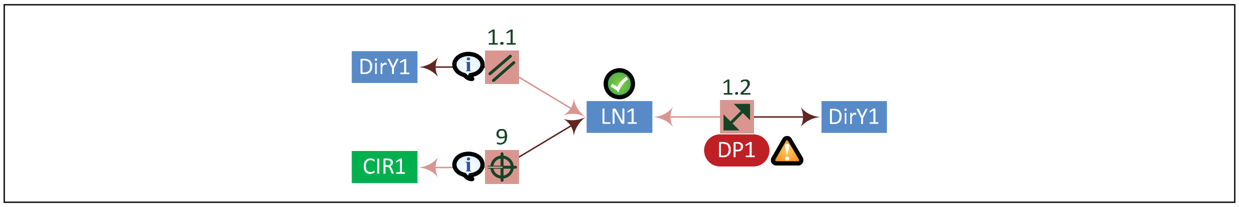


Figure A II-8 Premier cas de figure : Transposition de l'élément de forme auxiliaire vers la troisième forme cible

A II.1.4 Premier cas de figure : Quatrième forme cible modifiée

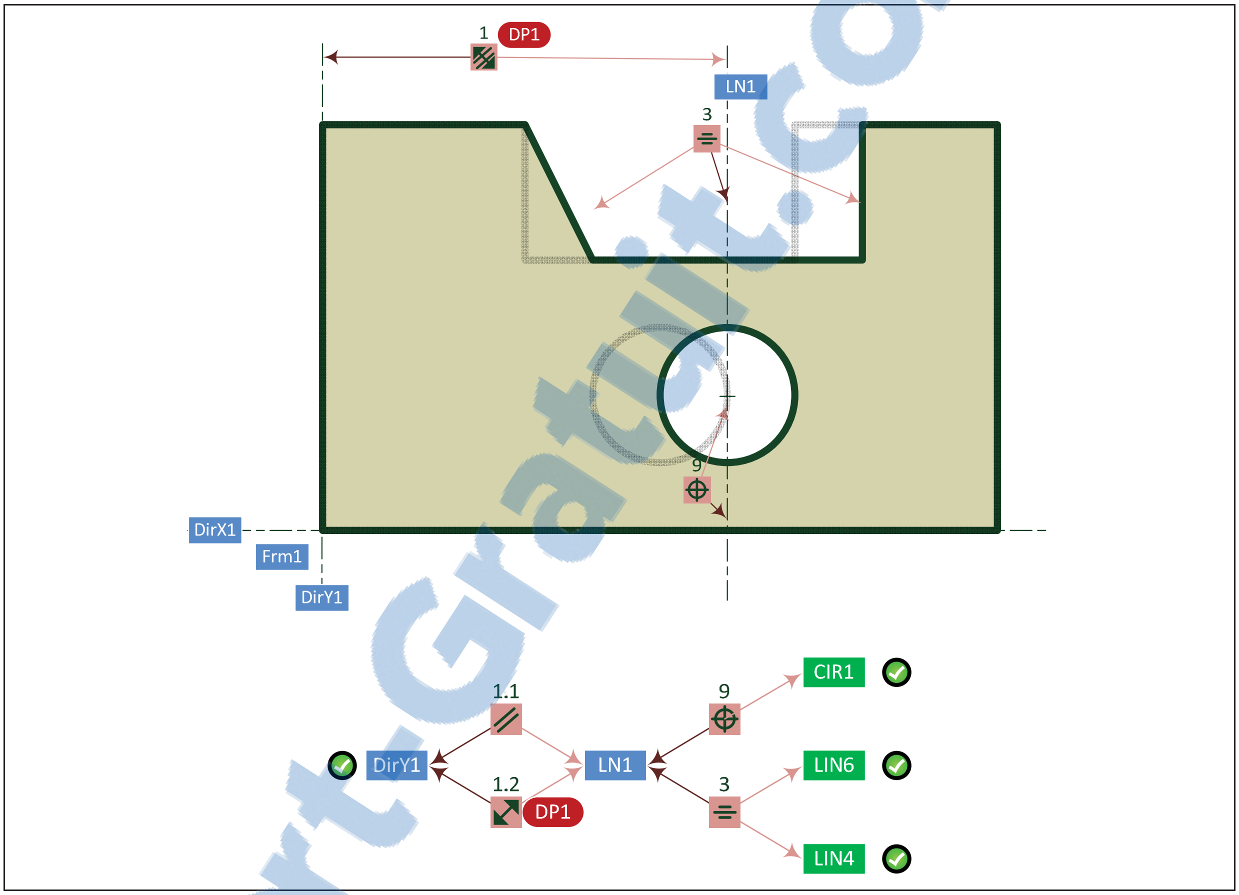


Figure A II-9 Premier cas de figure : Quatrième forme cible modifiée et état du schéma original de contraintes en fonction des éléments de forme modifiés

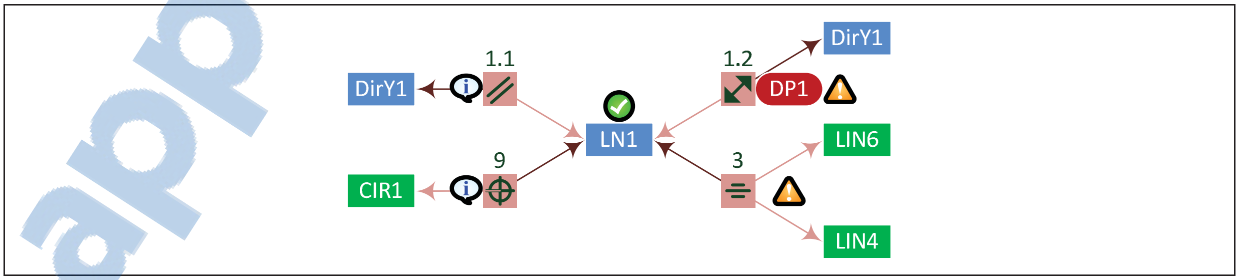


Figure A II-10 Premier cas de figure : Transposition de l'élément de forme auxiliaire vers la quatrième forme cible

A II.2 Deuxième cas de figure

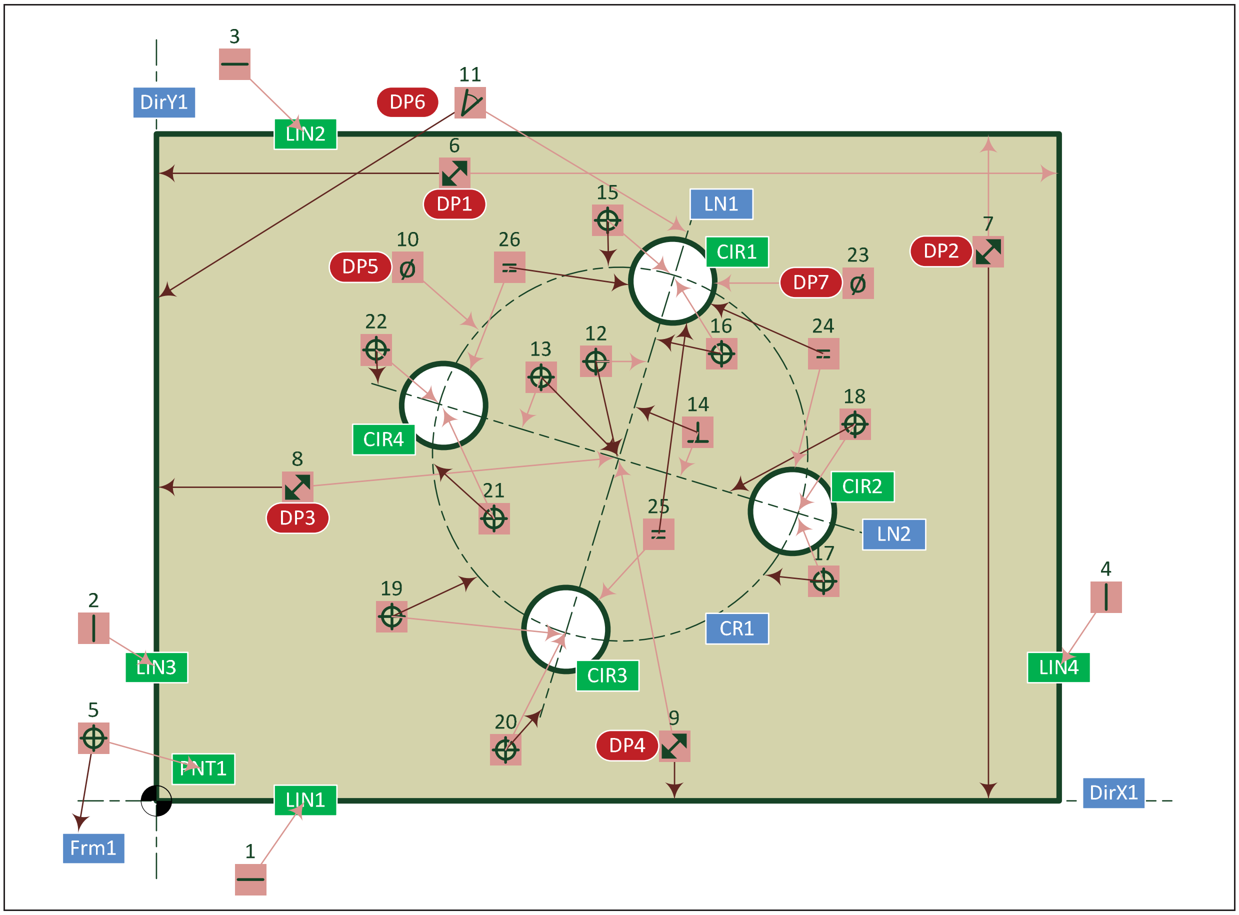


Figure A II-11 Deuxième cas de figure : Forme de référence originale et schéma de contraintes géométriques explicites

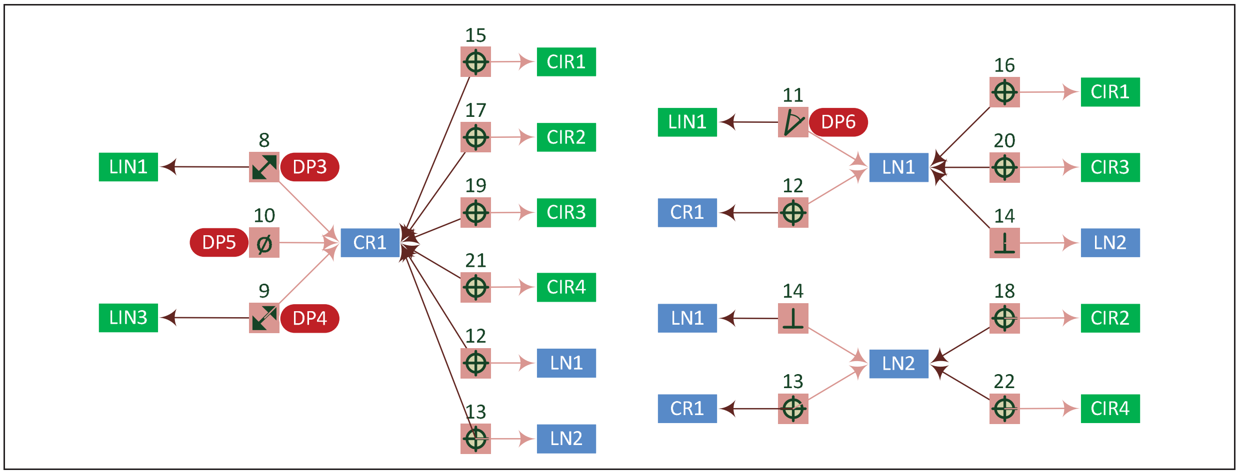


Figure A II-12 Deuxième cas de figure : Schémas originaux de contraintes relatives aux éléments de forme auxiliaires à transposer

A II.2.1 Deuxième cas de figure : Première forme cible modifiée

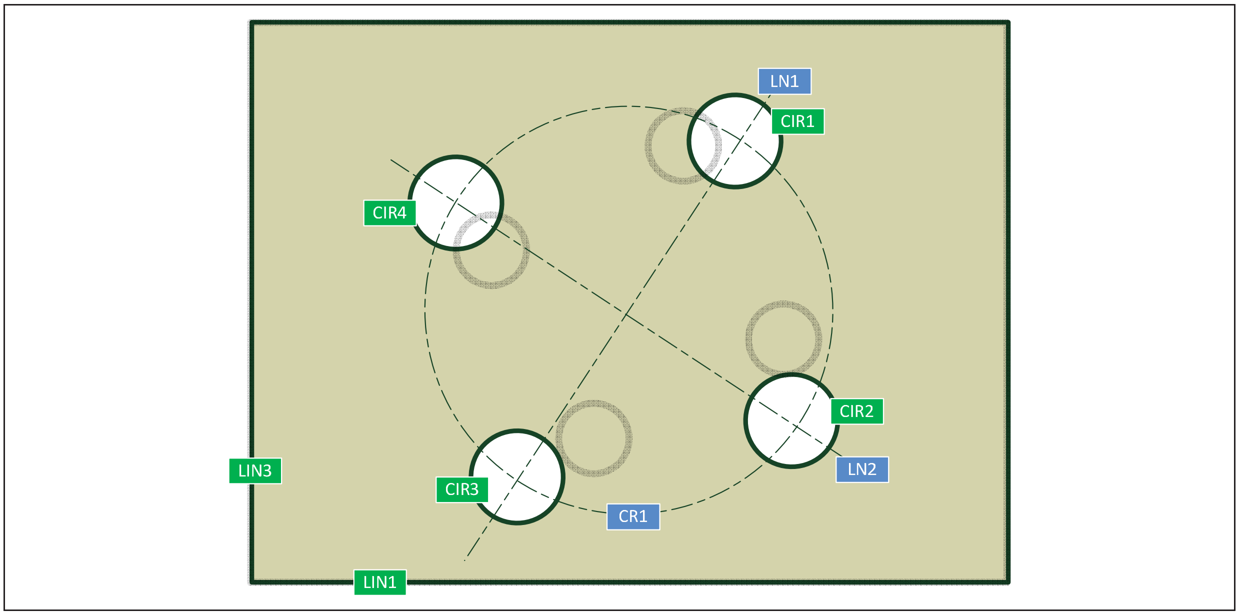


Figure A II-13 Deuxième cas de figure : Première forme cible modifiée

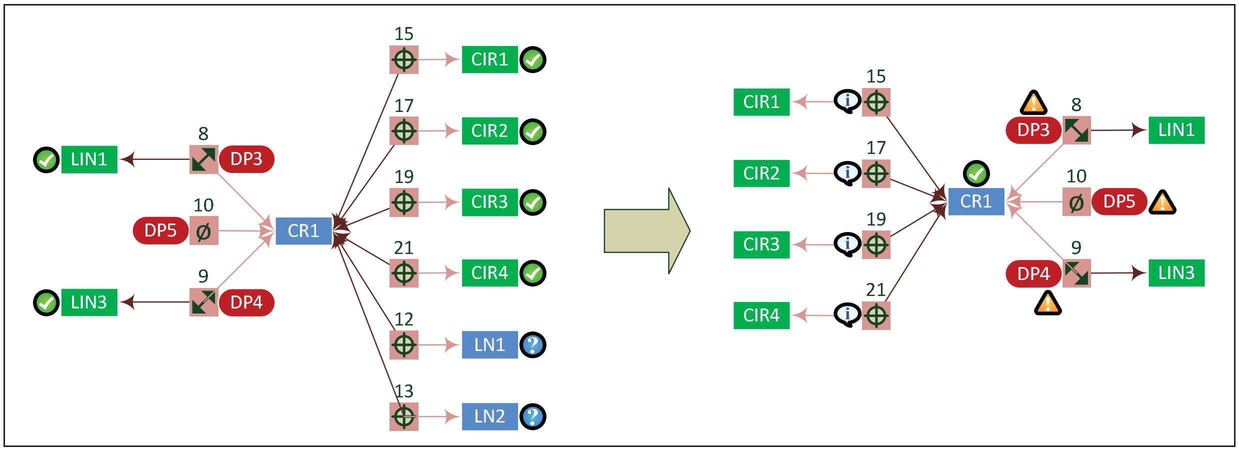


Figure A II-14 Deuxième cas de figure : Transposition de l'élément de forme auxiliaire 'Cr1' vers la première forme cible

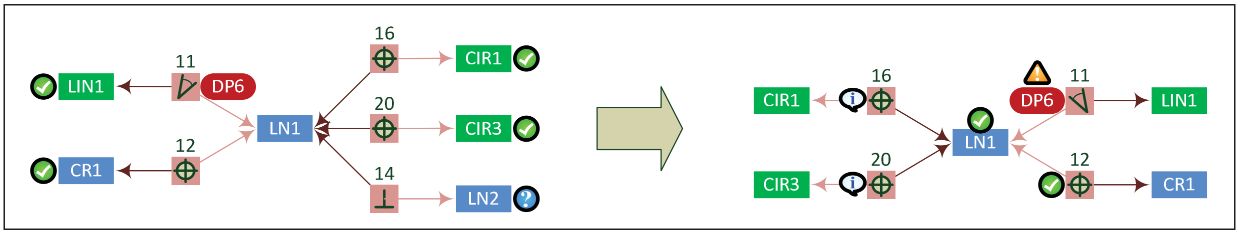


Figure A II-15 Deuxième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln1' vers la première forme cible

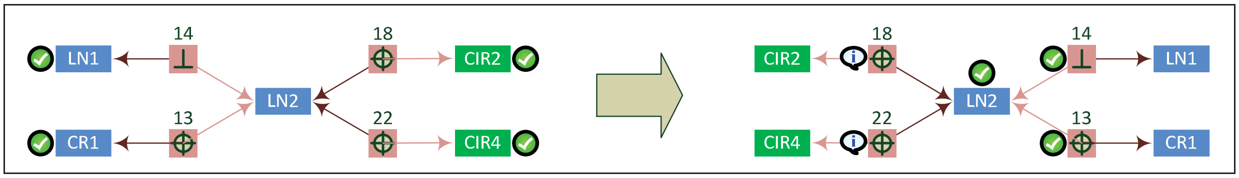


Figure A II-16 Deuxième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln2’ vers la première forme cible

A II.2.2 Deuxième cas de figure : Deuxième forme cible modifiée

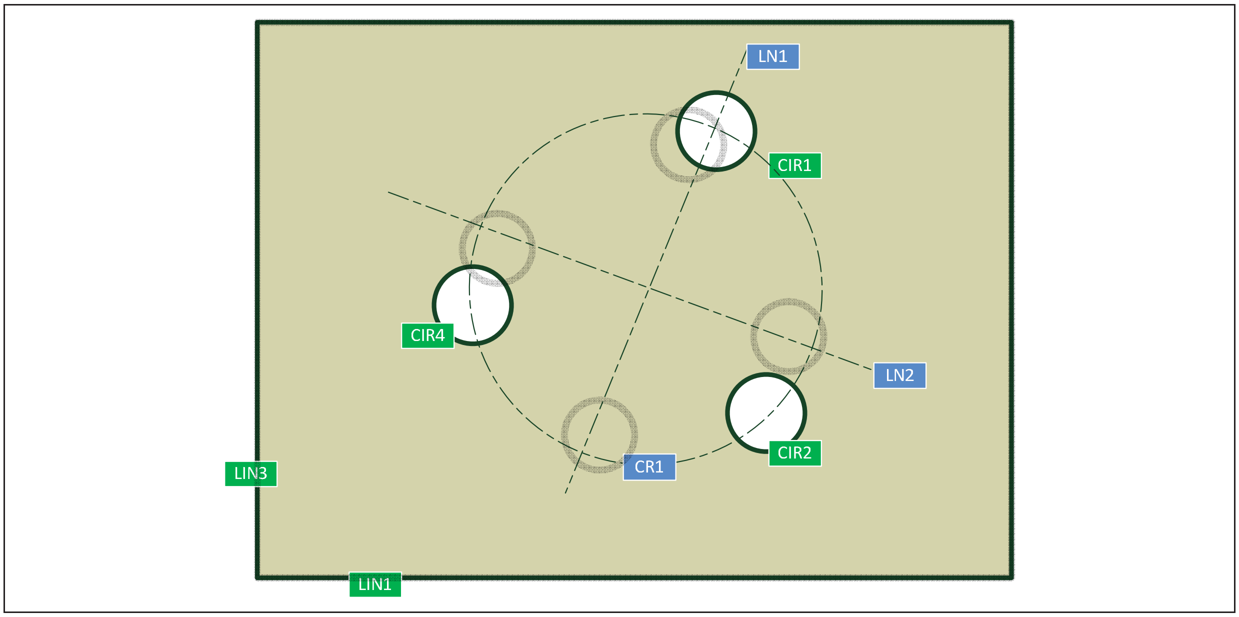


Figure A II-17 Deuxième cas de figure : Deuxième forme cible modifiée

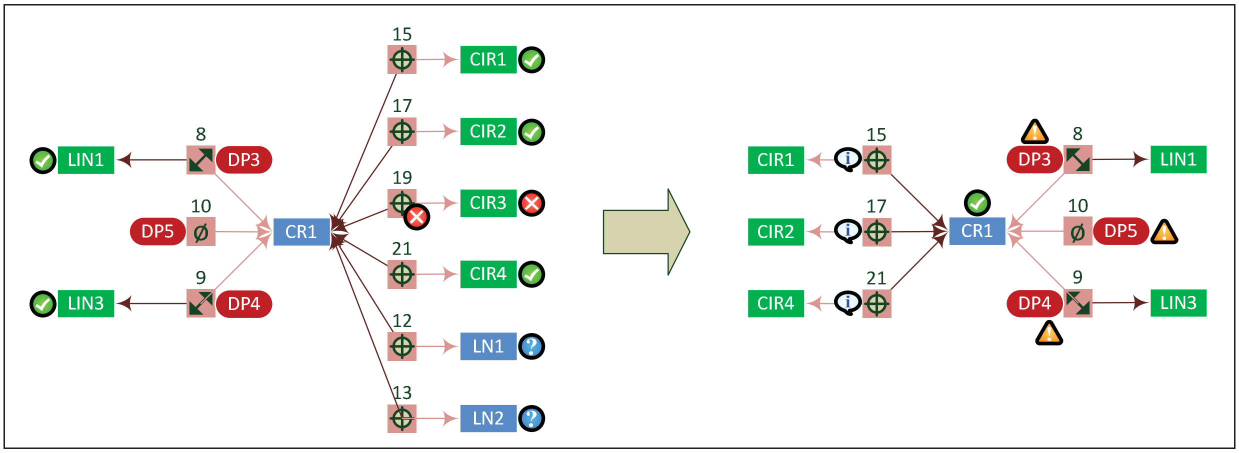


Figure A II-18 Deuxième cas de figure : Transposition de l’élément de forme auxiliaire ‘Cr1’ vers la deuxième forme cible

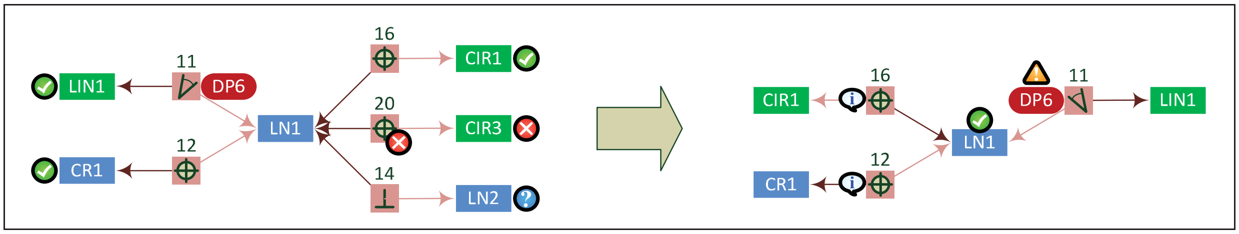


Figure A II-19 Deuxième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln1' vers la deuxième forme cible

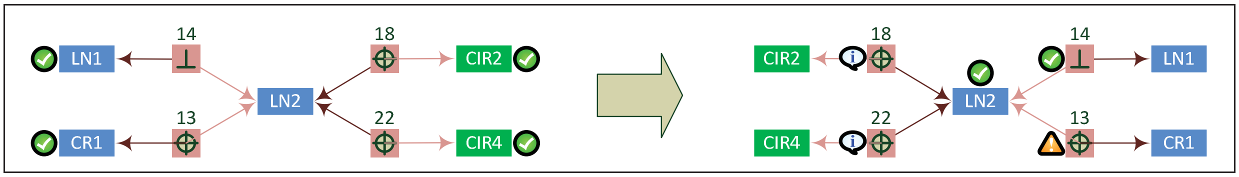


Figure A II-20 Deuxième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln2' vers la deuxième forme cible

A II.3 Troisième cas de figure

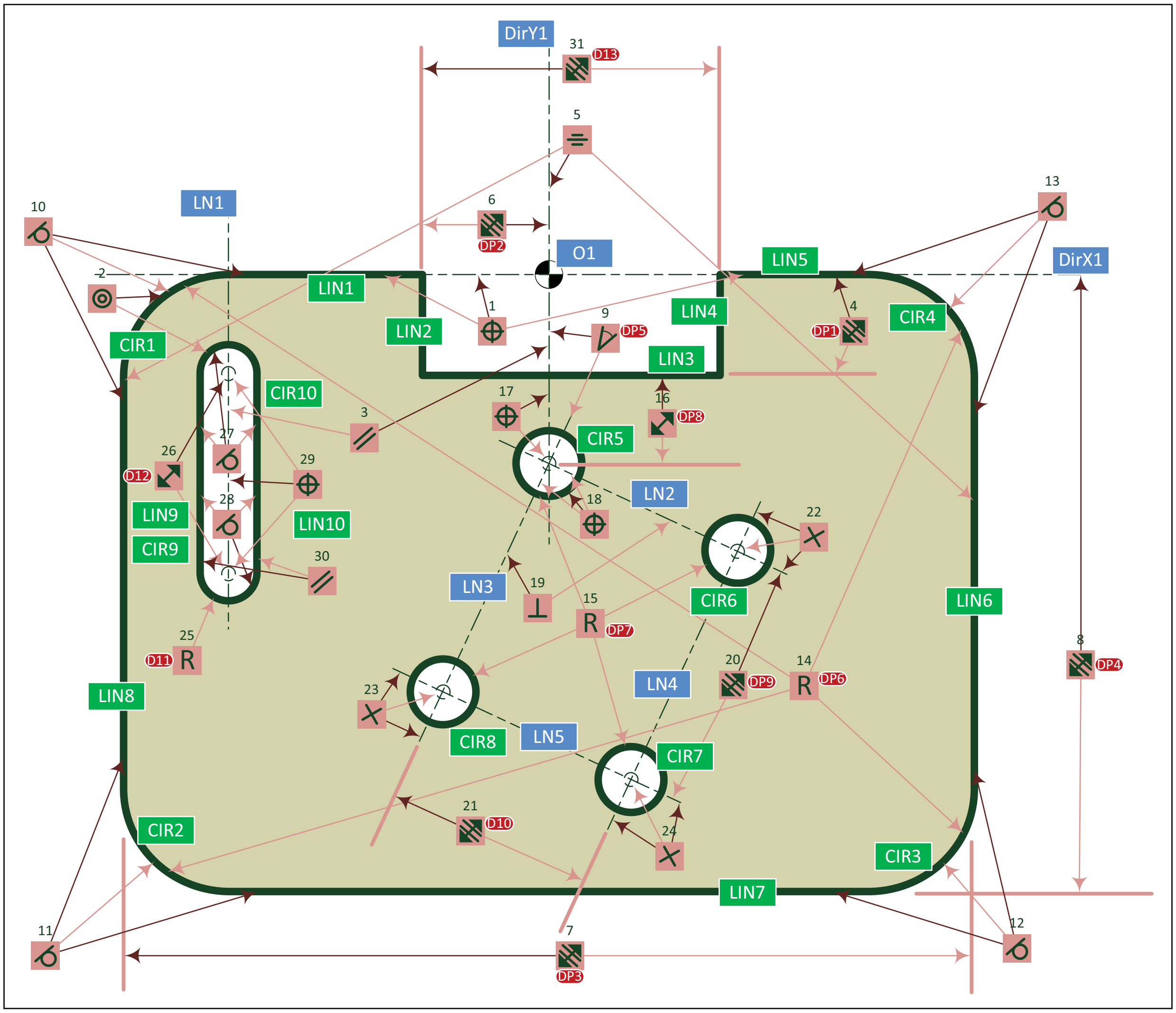


Figure A II-21 Troisième cas de figure : Forme de référence originale et schéma de contraintes géométriques explicites

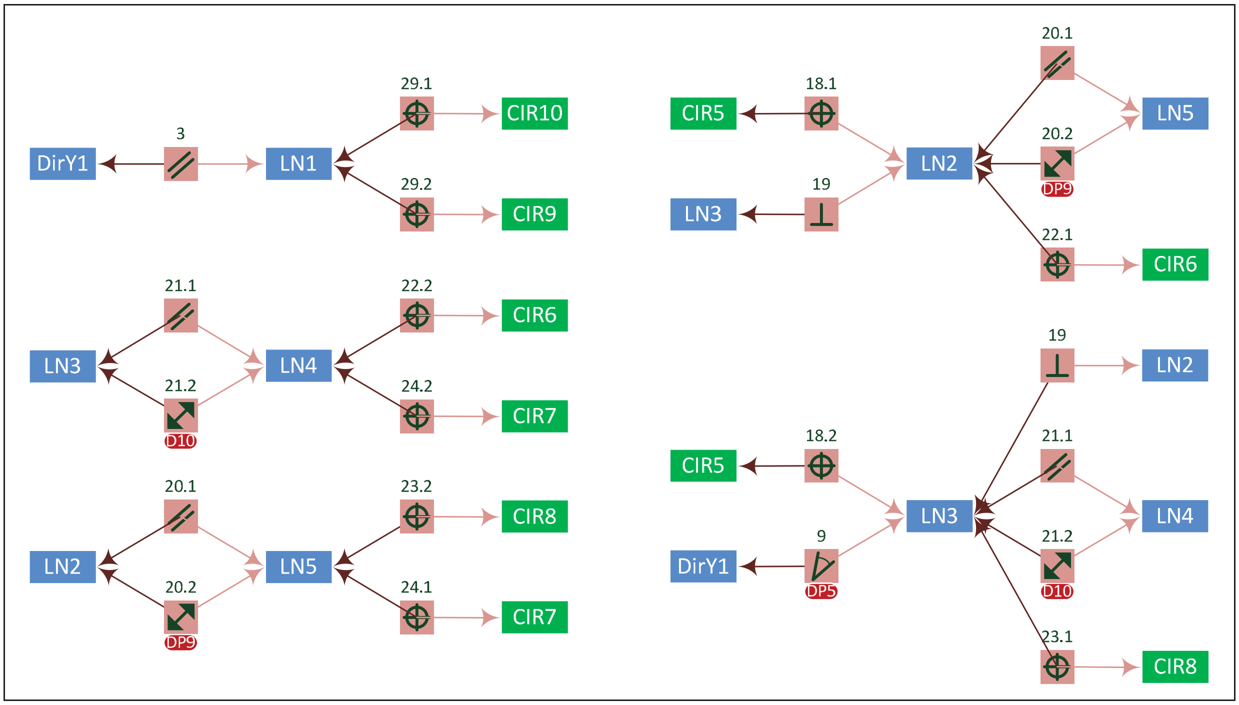


Figure A II-22 Troisième cas de figure : Schémas originaux de contraintes relatives aux éléments de forme auxiliaires à transposer

A II.3.1 Troisième cas de figure : Première forme cible modifiée

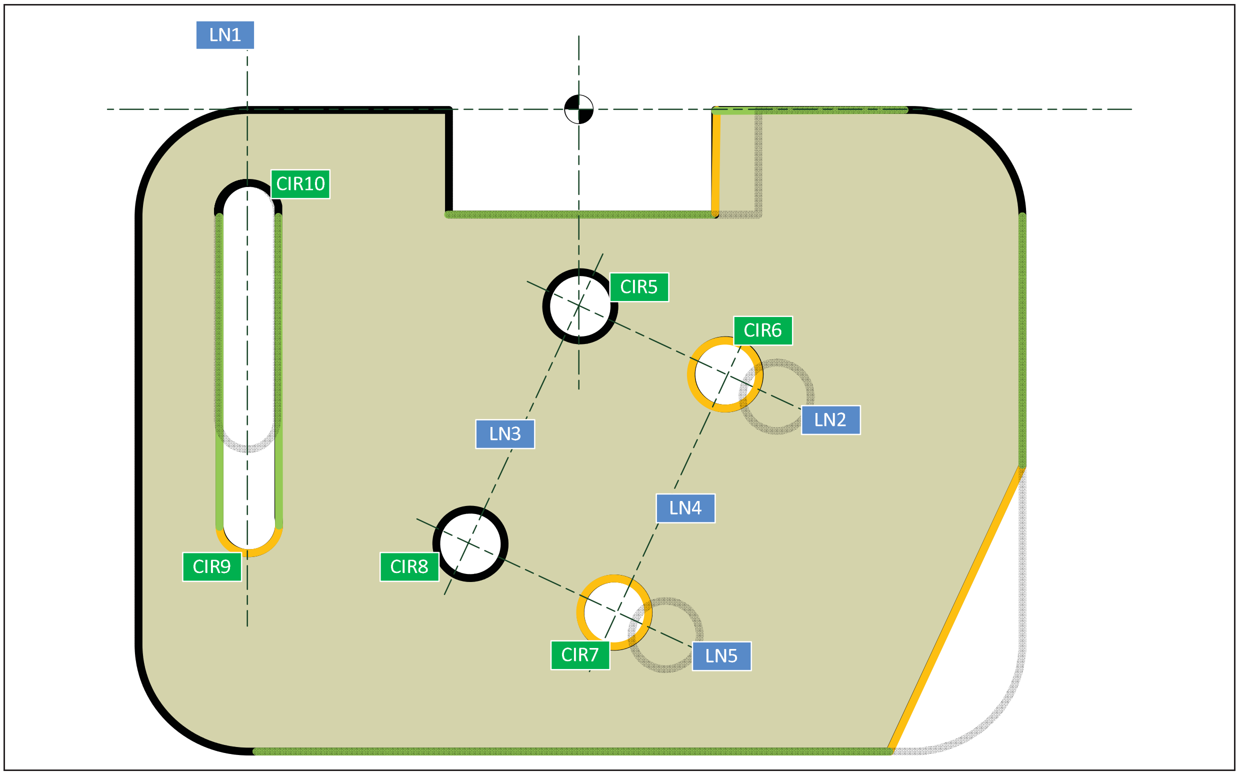


Figure A II-23 Troisième cas de figure : Première forme cible modifiée

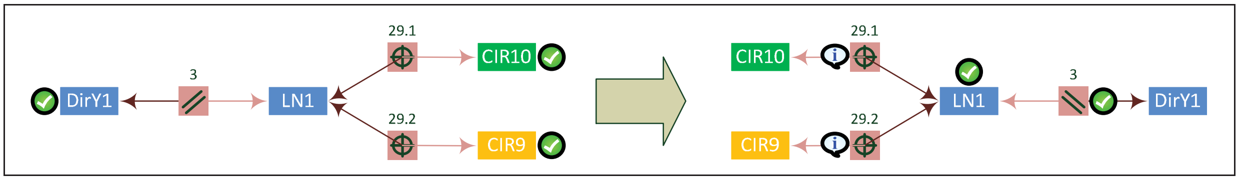


Figure A II-24 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln1' vers la première forme cible

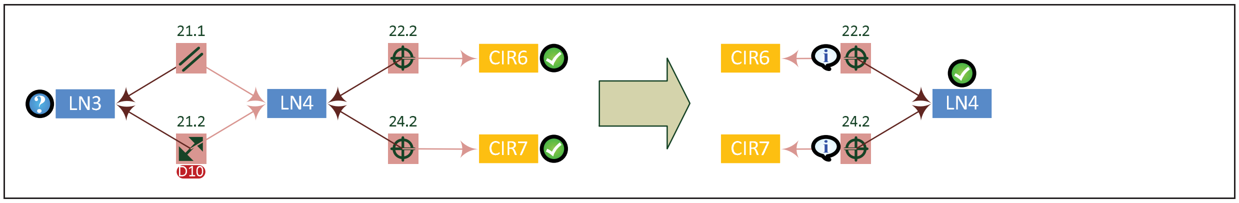


Figure A II-25 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln4' vers la première forme cible

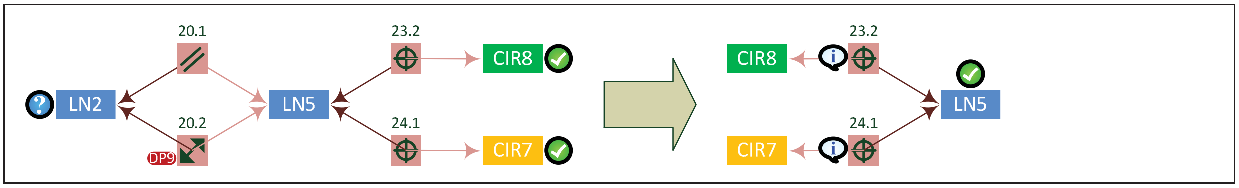


Figure A II-26 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln5' vers la première forme cible

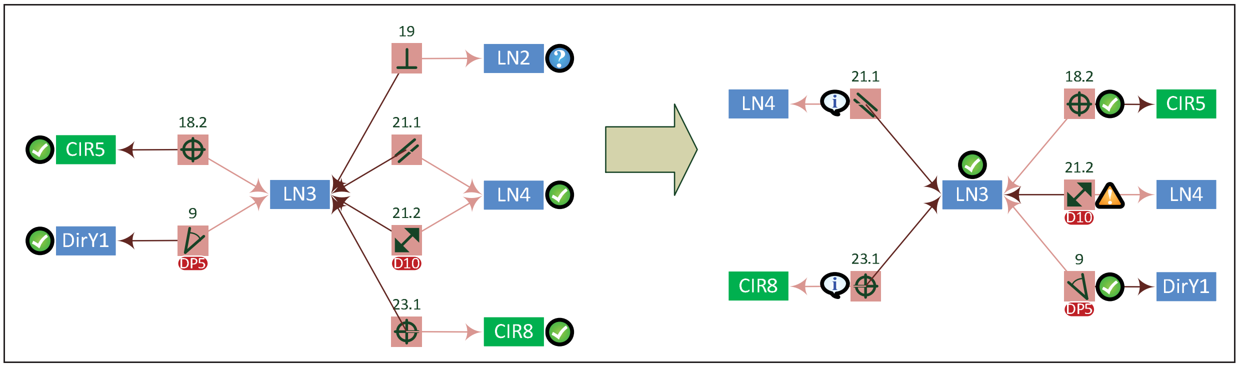


Figure A II-27 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln3' vers la première forme cible

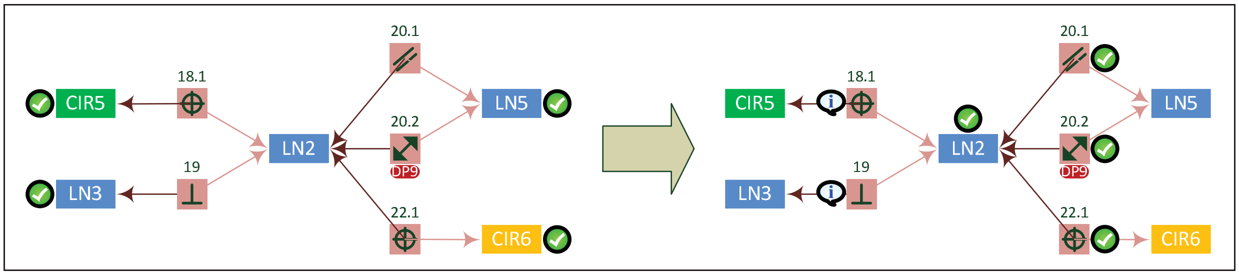


Figure A II-28 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln2' vers la première forme cible

A II.3.2 Troisième cas de figure : Deuxième forme cible modifiée

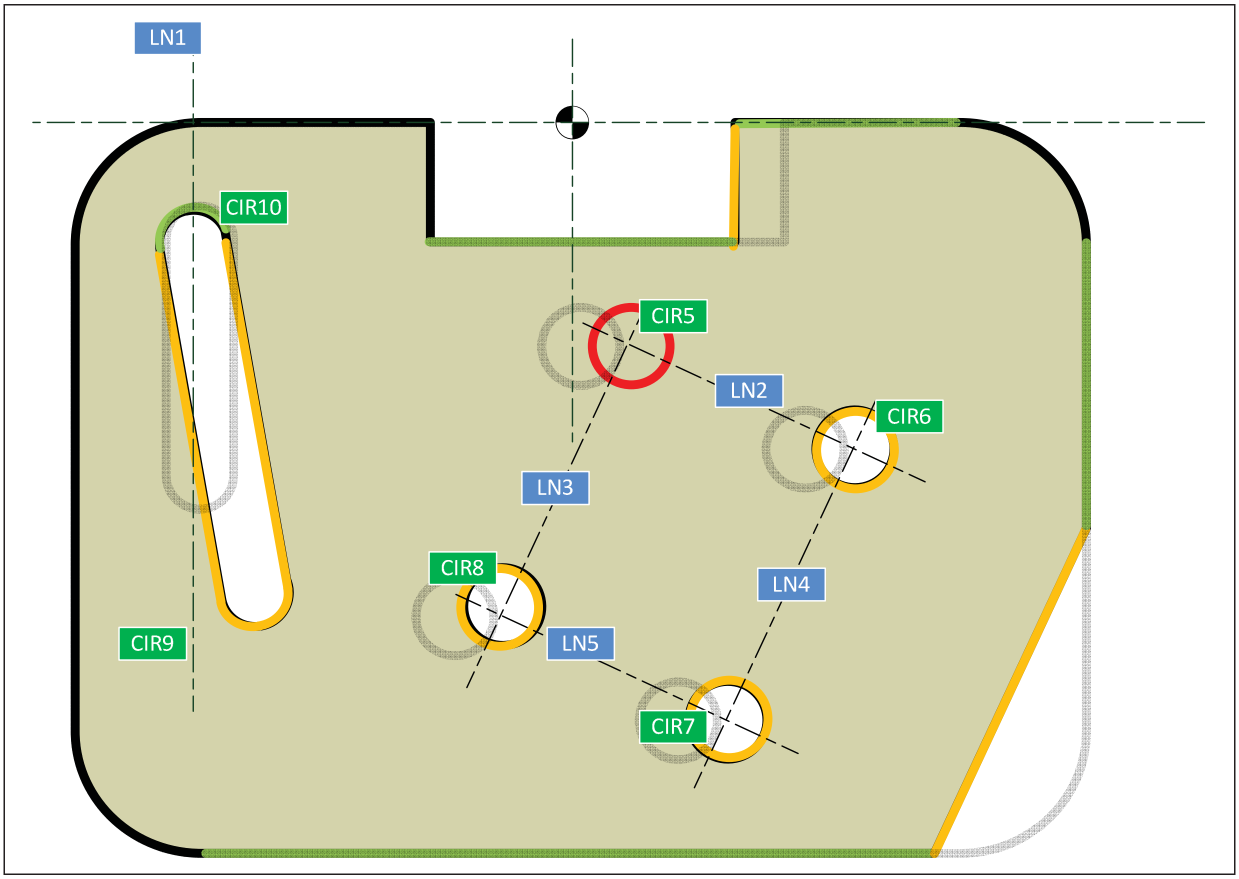


Figure A II-29 Troisième cas de figure : Deuxième forme cible modifiée

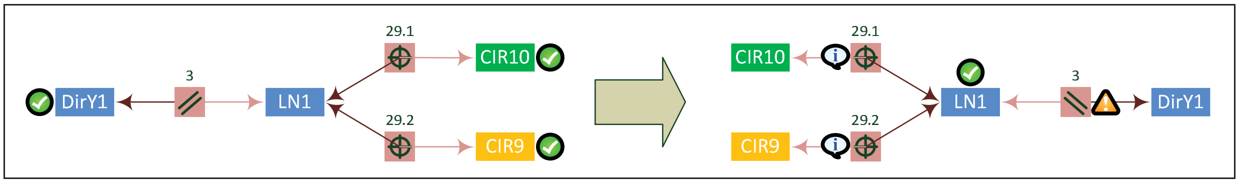


Figure A II-30 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln1' vers la deuxième forme cible

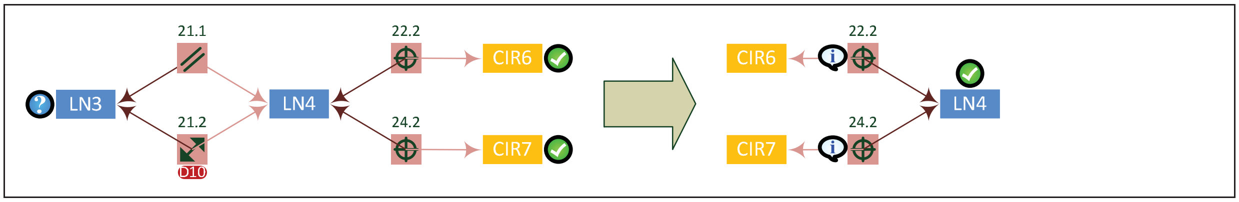


Figure A II-31 Troisième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln4' vers la deuxième forme cible

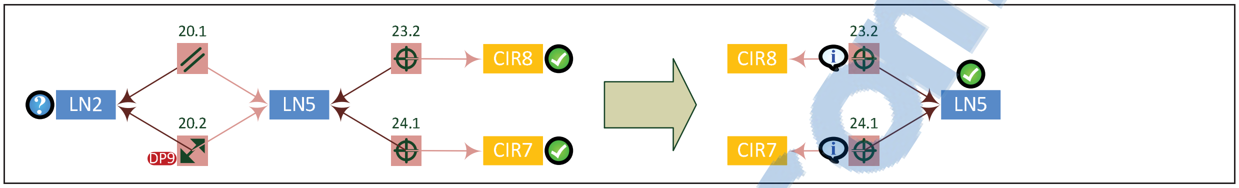


Figure A II-32 Troisième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln5’ vers la deuxième forme cible

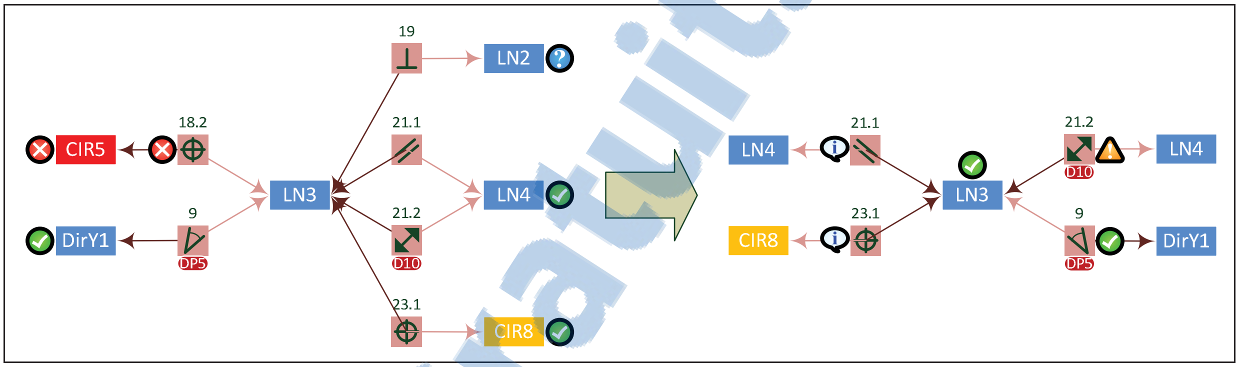


Figure A II-33 Troisième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln3’ vers la deuxième forme cible

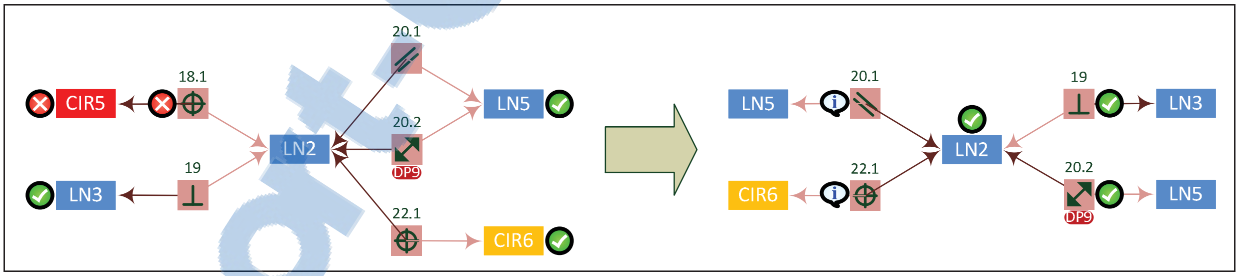


Figure A II-34 Troisième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln2’ vers la deuxième forme cible

A II.4 Quatrième cas de figure

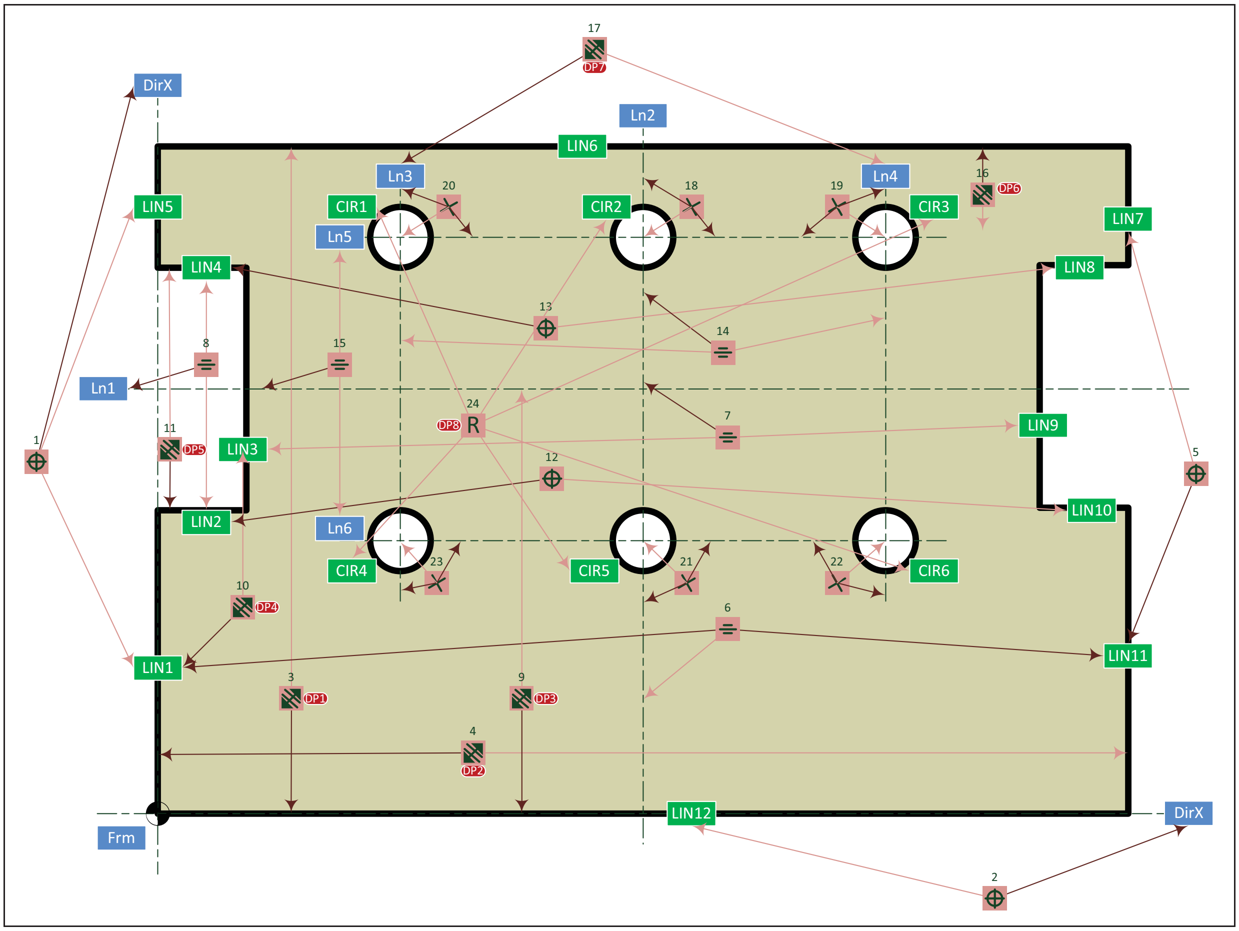


Figure A II-35 Quatrième cas de figure : Forme de référence originale et schéma de contraintes géométriques explicites

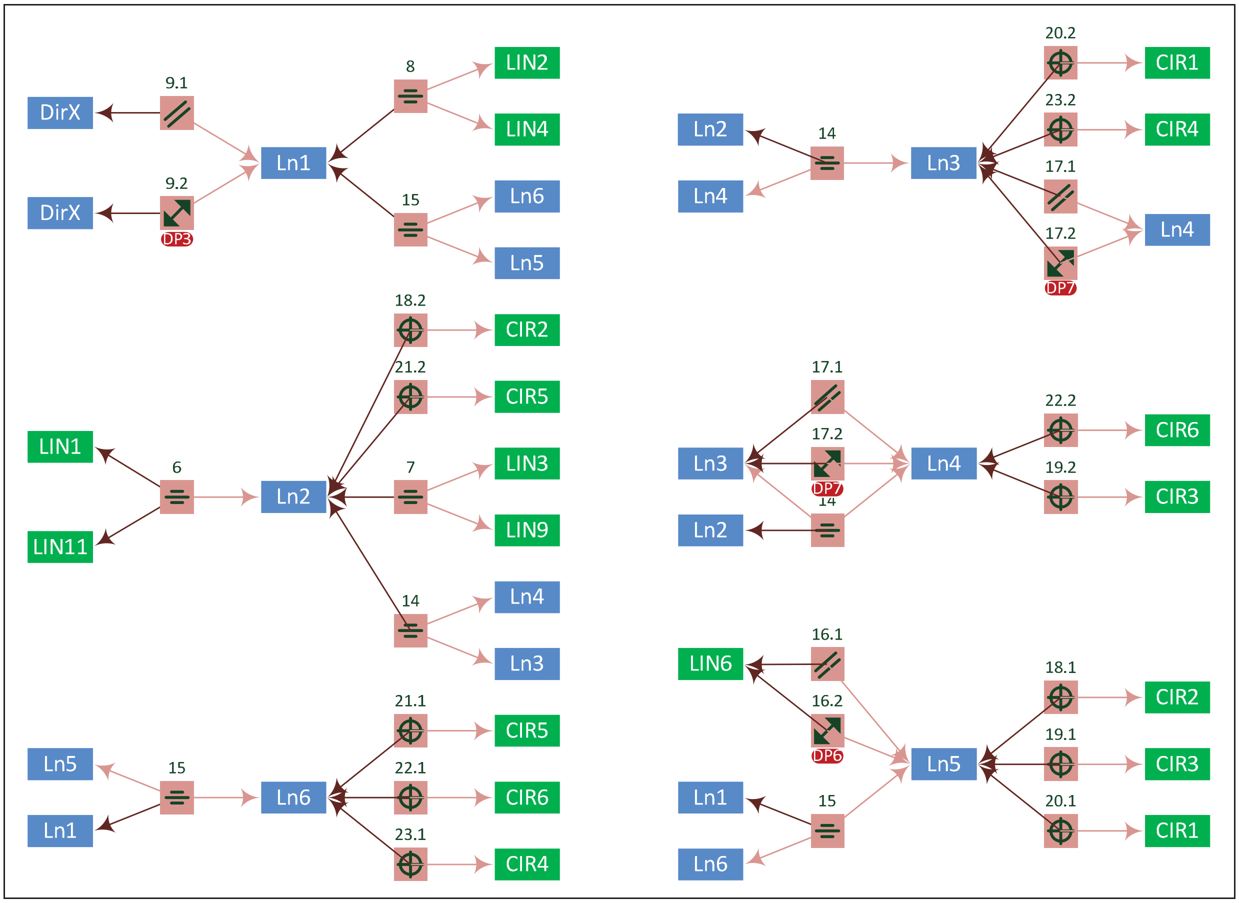


Figure A II-36 Quatrième cas de figure : Schémas originaux de contraintes relatives aux éléments de forme auxiliaires à transposer

A II.4.1 Quatrième cas de figure : Forme cible modifiée

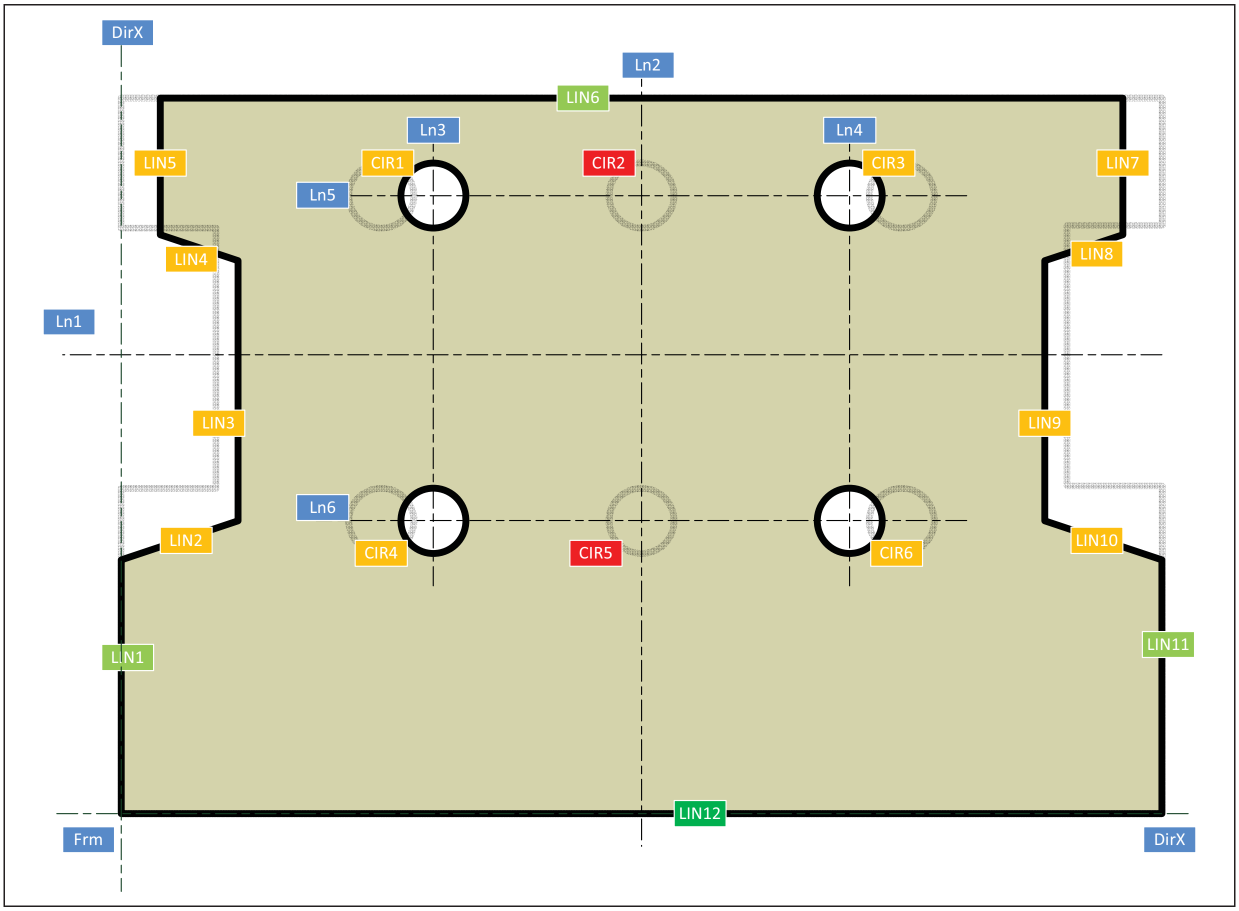


Figure A II-37 Quatrième cas de figure : Forme cible modifiée

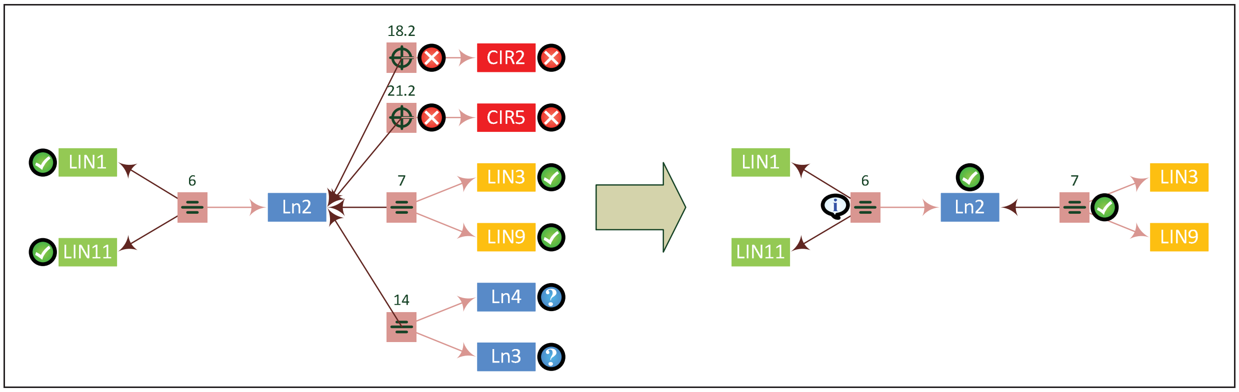


Figure A II-38 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln2’ vers la forme cible

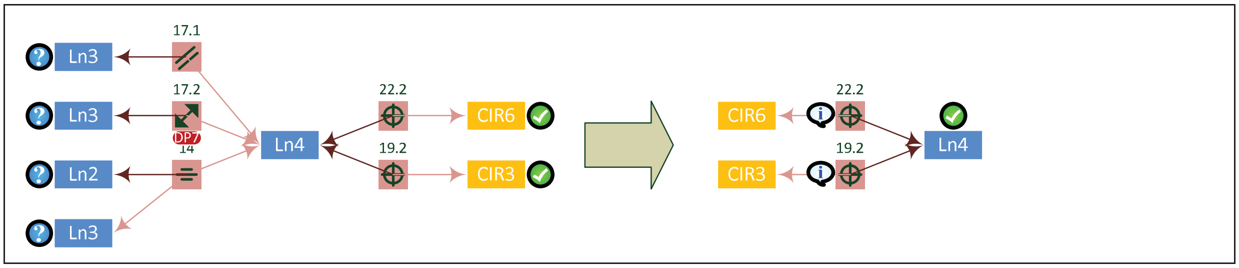


Figure A II-39 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln4’ vers la forme cible

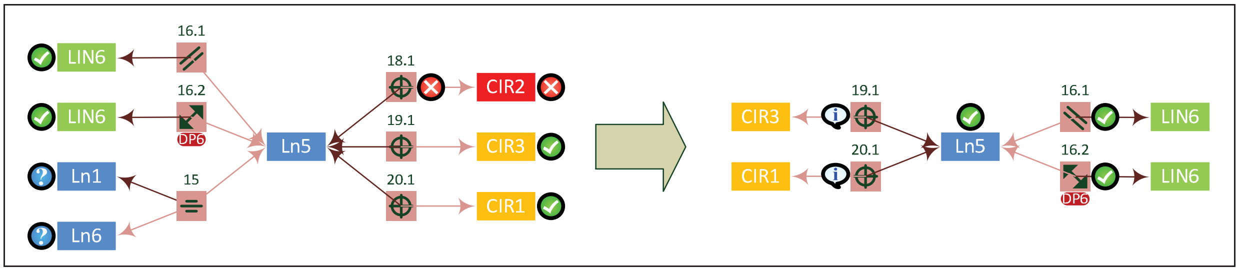


Figure A II-40 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln5’ vers la forme cible

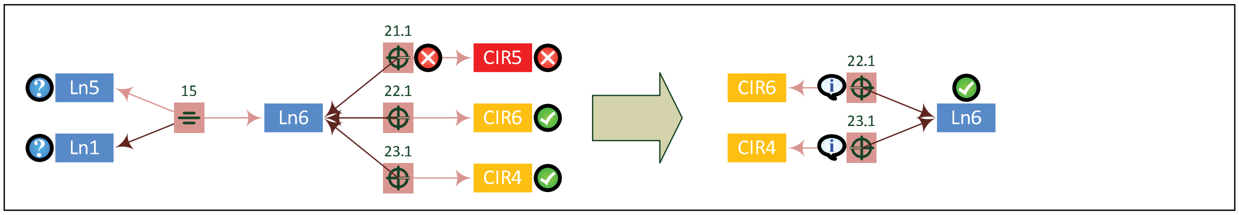


Figure A II-41 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln6’ vers la forme cible

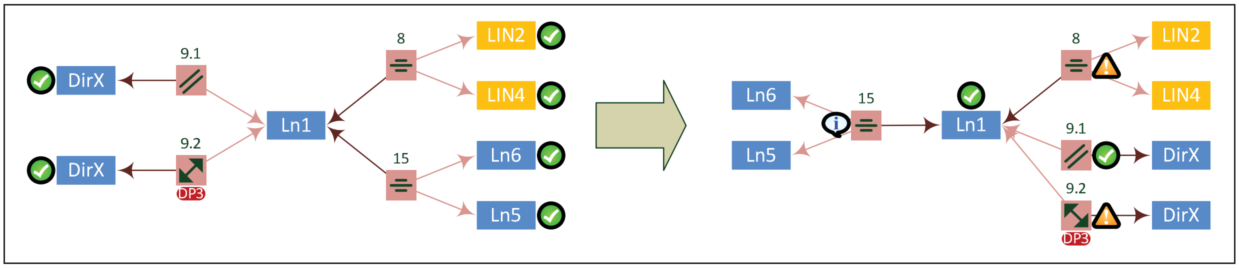


Figure A II-42 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln1’ vers la forme cible

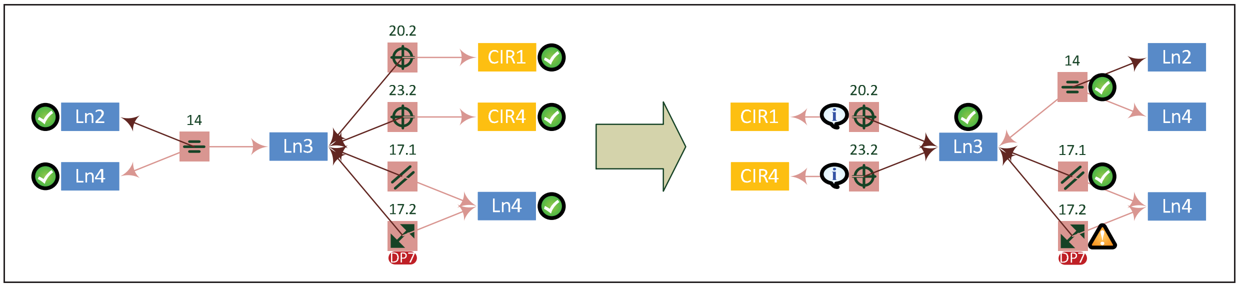


Figure A II-43 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln3’ vers la forme cible

A II.5 Cinquième cas de figure

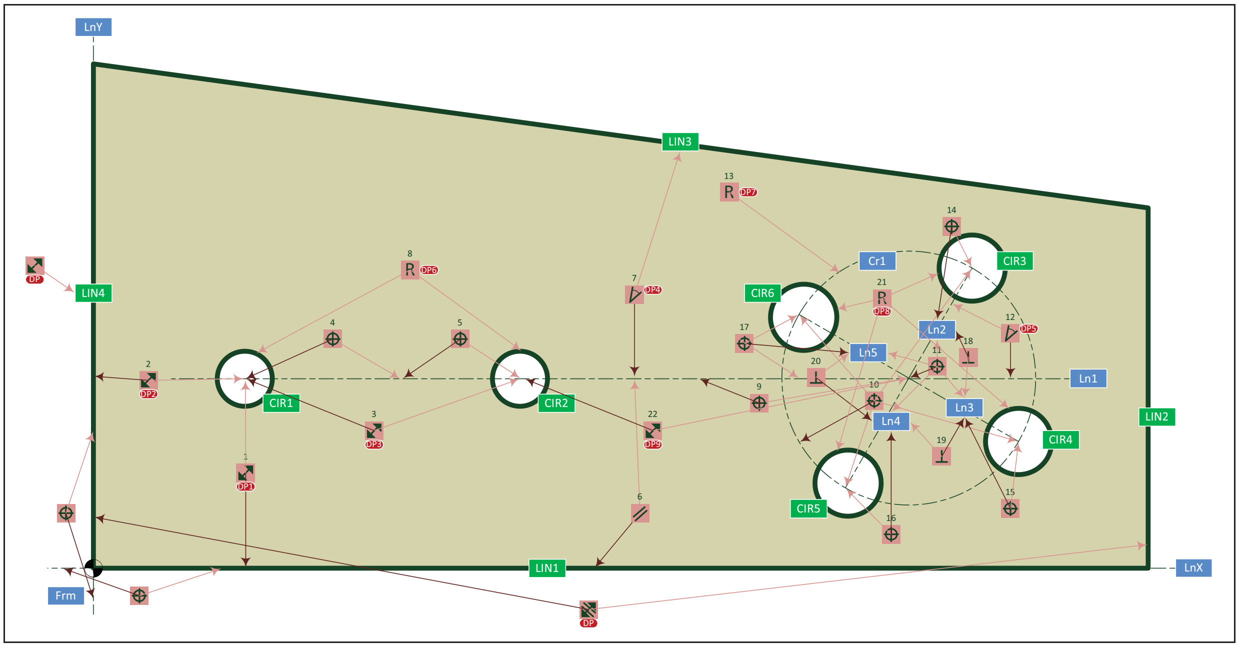


Figure A II-44 Cinquième cas de figure : Forme de référence originale et schéma de contraintes géométriques explicites

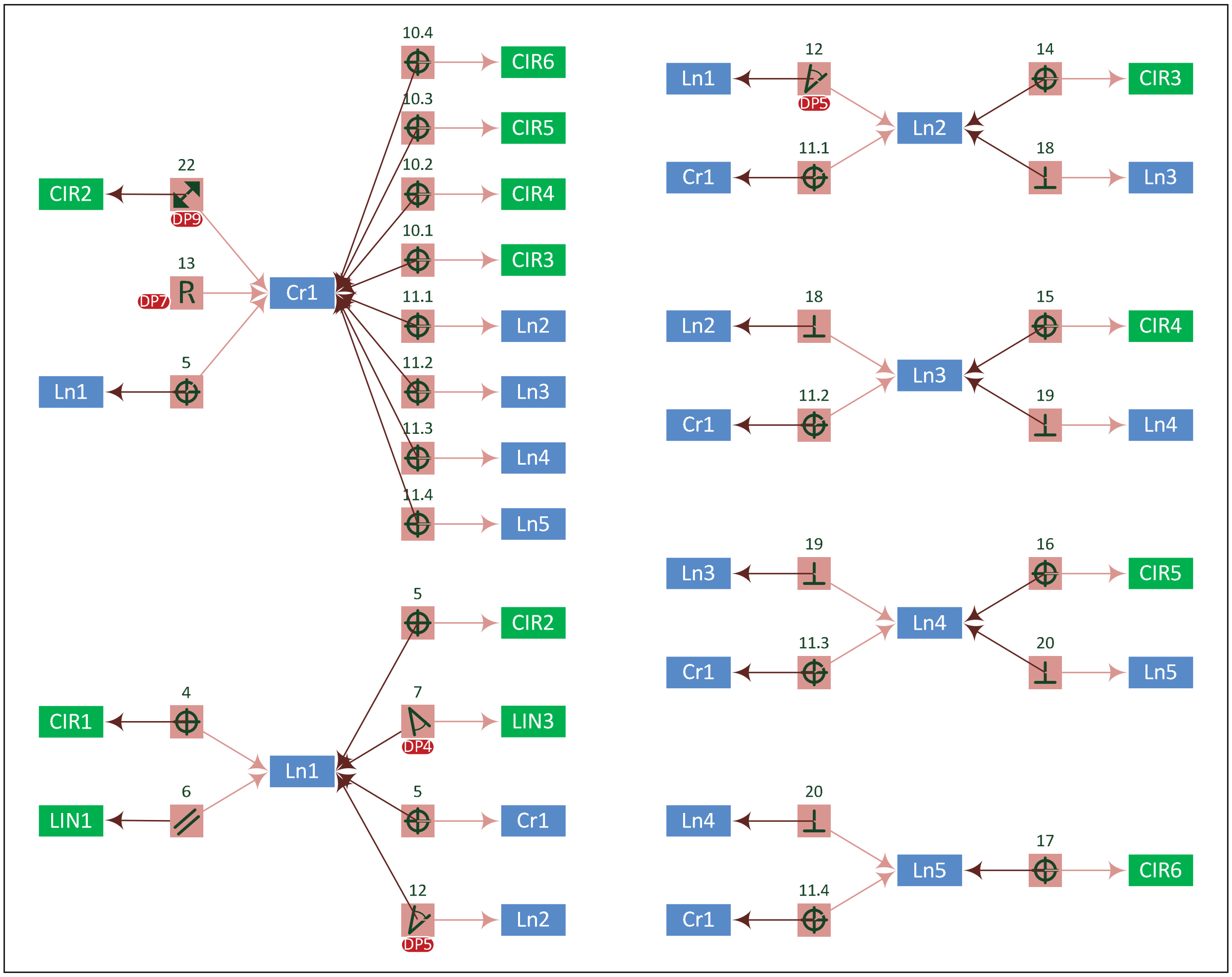


Figure A II-45 Cinquième cas de figure : Schémas originaux de contraintes relatives aux éléments de forme auxiliaires à transposer

A II.5.1 Cinquième cas de figure : Forme cible modifiée

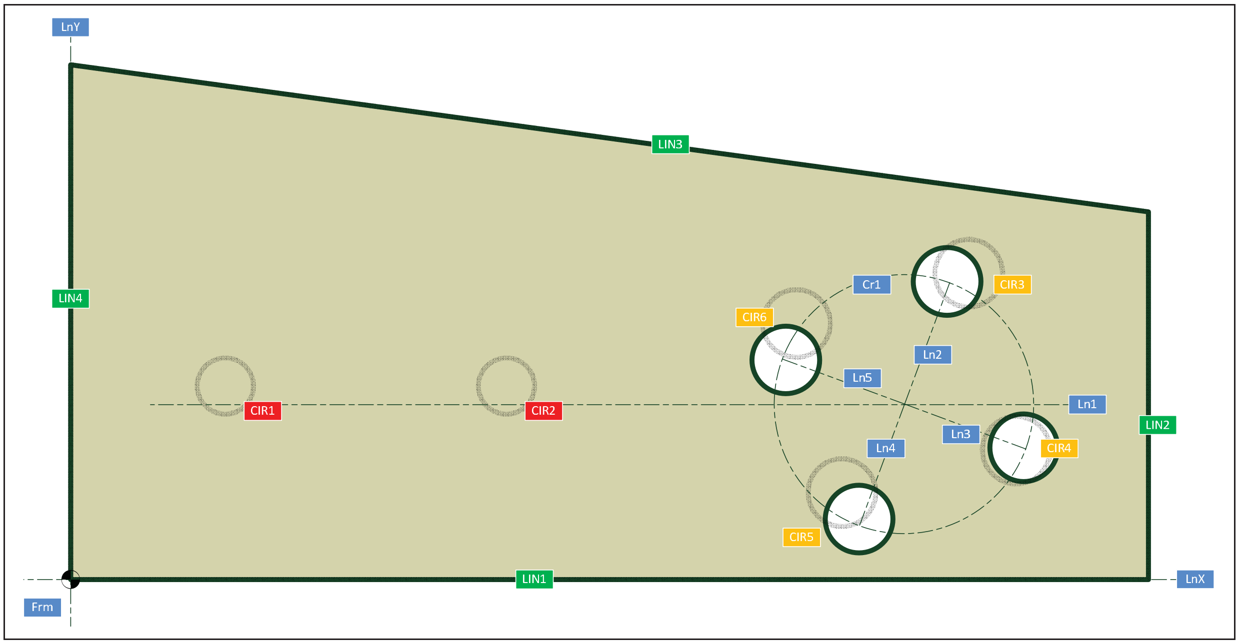


Figure A II-46 Cinquième cas de figure : Forme cible modifiée

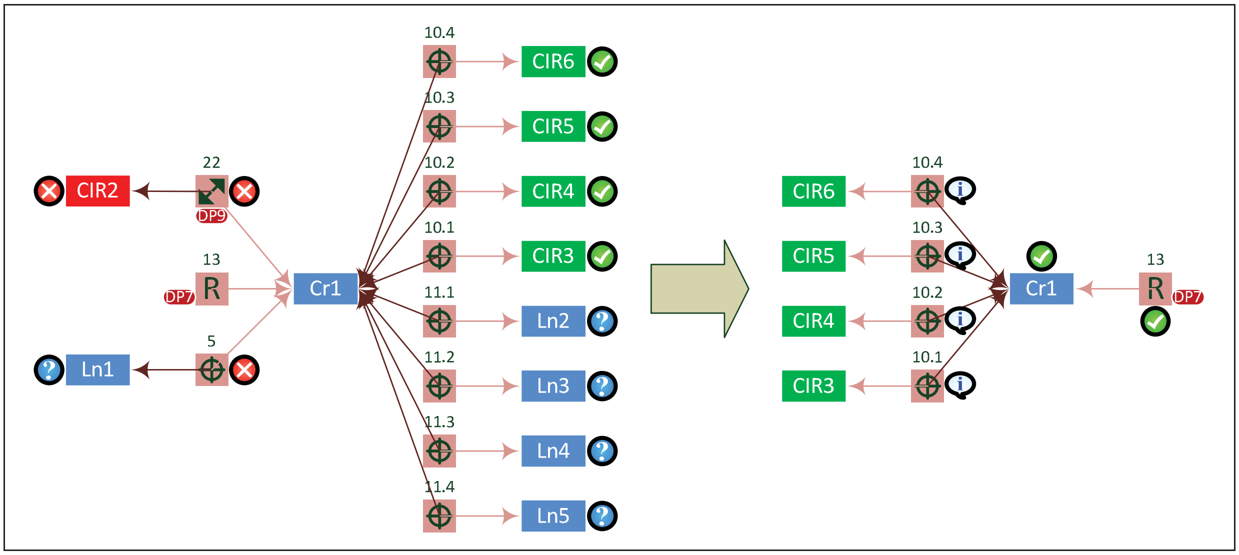


Figure A II-47 Quatrième cas de figure : Transposition de l’élément de forme auxiliaire ‘Cr1’ vers la forme cible

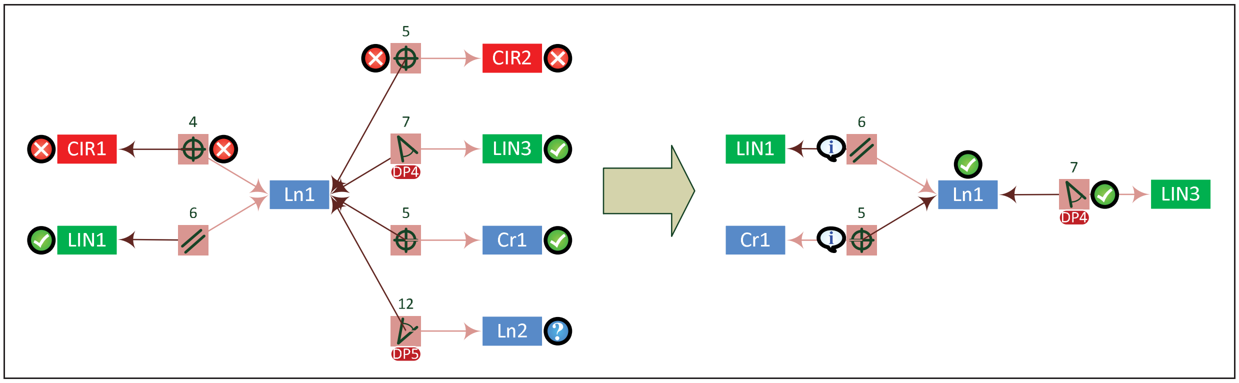


Figure A II-48 Cinquième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln1' vers la forme cible

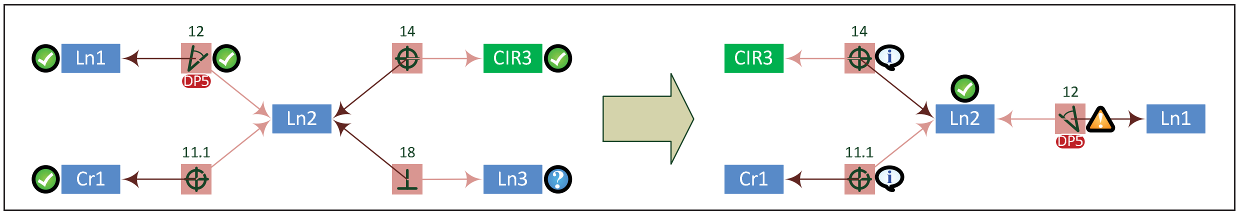


Figure A II-49 Cinquième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln2' vers la forme cible

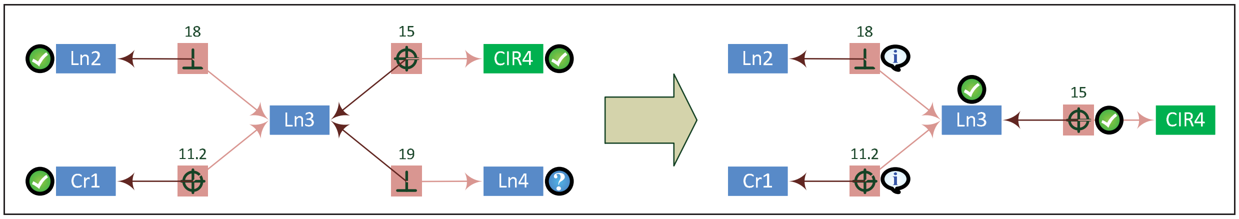


Figure A II-50 Cinquième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln3' vers la forme cible

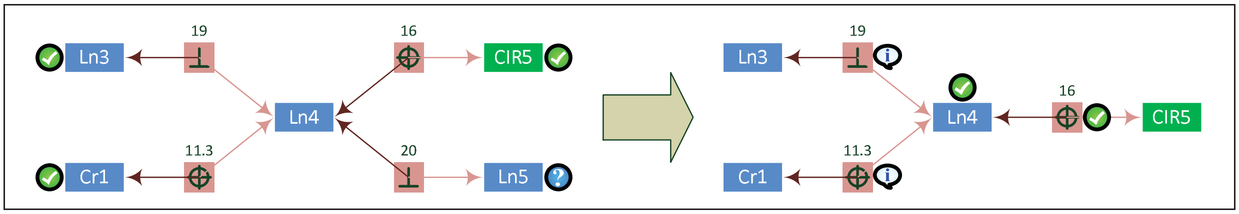


Figure A II-51 Cinquième cas de figure : Transposition de l'élément de forme auxiliaire 'Ln4' vers la forme cible

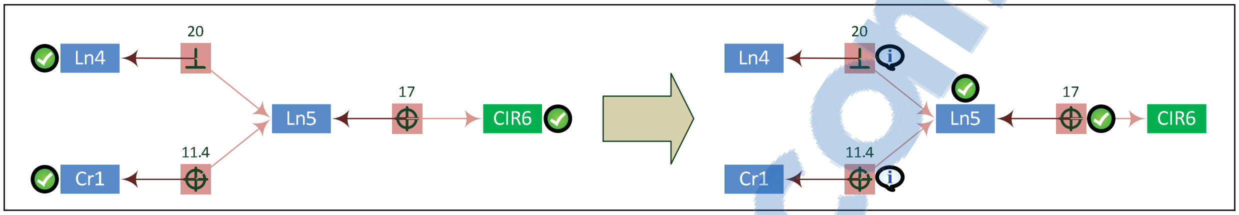


Figure A II-52 Cinquième cas de figure : Transposition de l’élément de forme auxiliaire ‘Ln5’ vers la forme cible

BIBLIOGRAPHIE

3DSemantix inc. 2011. « 3DPartFinder: Geometric Search Engine ». In *3DPartFinder: Geometric Search Engine*. < http://www.3dpartfinder.com/en/ >. Consulté le 18 janvier 2012.

Actify inc. 2010. *Spinfire.*(Version 9.0). Logiciel d'évaluation. San Fransisco, CA (USA): Actify inc. < http://www.actify.com/products/spinfire-professional/2d-3d-cad-system-viewer-file-converter-download >. Consulté le 20 octobre 2010.

Adobe Systems inc. 2008. *Adobe Acrobat 9.0 Pro Extended: 3D Reviewer.*(Version 9.4.0.112). Logiciel commercial. San Jose, CA: Adobe Systems inc. < http://www.adobe.com/products/acrobat.html >.

Agbodan, D., D. Marcheix, G. Pierra et C. Thabaud. 2003. « A topological entity matching technique for geometric parametric models ». In *Proceedings. Shape Modeling International 2003, 12-15 May 2003*. p. 235-44. IEEE Comput. Soc. < http://dx.doi.org/10.1109/SMI.2003.1199623 >. Consulté le 6 mars 2008.

Akenine-Möller, Tomas. 2005. « Fast 3D triangle-box overlap testing ». In *ACM SIGGRAPH 2005 Courses*. Los Angeles, California: ACM. < http://doi.acm.org/10.1145/1198555.1198747 >. Consulté le 17 novembre 2008.

Al-Sabeh, Mazen. 2004. « Algorithme de comparaison de modèles géometriques en évolution : une application avec CATIA V5 ». Mémoire de maîtrise en génie mécanique en ligne. Montréal, École de technologie supérieure, 111 p. In ProQuest Dissertations and Theses. Ann Harbour, MI : ProQuest LLC. < http://proquest.umi.com/pqdlink?did=766718491&Fmt=7&clientId=46962&RQT=309&VName=PQD >. Consulté le 3 juin 2008.

American Society of Mechanical Engineers. 2009. *Geometric dimensioning and tolerancing*. ASME Y14.5-2009. New York: American Society of Mechanical Engineers. < http://www.asme.org/products/codes---standards/dimensioning-and-tolerancing >.

American Society of Mechanical Engineers. 2012. *Digital Product Definition Data Practices*. ASME Y14.41-2012. New York, NY: American Society of Mechanical Engineers. < https://www.asme.org/products/codes-standards/y1441-2012-digital-product-definition-data >.

Aspert, N., D. Santa-Cruz et T. Ebrahimi. 2002. « MESH: measuring errors between surfaces using the Hausdorff distance ». In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 26-29 Aug. 2002*. Vol. vol.1, p. 705-8. IEEE. < http://dx.doi.org/10.1109/ICME.2002.1035879 >. Consulté le 26 novembre 2008.

Autodesk inc. 2010a. *Autodesk Inventor*. Prototype. San Rafael, CA (USA): Autodesk inc. < http://usa.autodesk.com/autodesk-inventor/ >.

Autodesk inc. 2010b. *Autodesk Inventor Fusion Technology Preview.*(Version 0.3.0.29). Prototype. San Rafael, CA: Autodesk inc. < http://labs.autodesk.com/technologies/fusion/ >.

Bai, Jing, Shuming Gao, Weihua Tang, Yusheng Liu et Song Guo. 2009. « Semantic-based partial retrieval of CAD models for design reuse ». In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*. (San Francisco, California), p. 271-276. 1629289: ACM.

Bettig, Bernhard, et J. Shah. 2001. « Derivation of a standard set of geometric constraints for parametric modeling and data exchange ». *Computer-Aided Design,* vol. 33, n° 1, p. 17-33.

Bouikni, Nadjib. 2005. « Ingénierie simultanée et gestion du cycle de vie du produit : Modèle de validation des évolutions des caractéristiques du produit ». Thèse de doctorat en génie en ligne. Sherbrooke, Université de Sherbrooke, 205 p. In ProQuest Dissertations and Theses. Ann Harbour, MI : ProQuest LLC. < http://proquest.umi.com/pqdweb?index=0&did=1155550871&SrchMode=2&sid=1&Fmt=6&VInst=PROD&VType=PQD&RQT=309&VName=PQD&TS=1260821285&clientId=46962 >. Consulté le 14 décembre 2009.

Brière-Côté, Antoine, Louis Rivest et Roland Maranzana. 2011. « A three-step approach to structuring 3D CAD model comparison scenarios ». In *IFIP WG5.1 8th International Conference on Product Lifecycle Management*, sous la dir. de Pels, Hank Jan, Abdelaziz Bouras et Chris McMahon. p. 429-443. IFIP WG5.1 on Global Product Development for the Whole Lifecycle. < http://www.phi-kpe.nl/contents-_page-37.html >.

Brière-Côté, Antoine, Louis Rivest et Roland Maranzana. 2012a. « Comparing 3D CAD models: Uses, methods, tools and perspectives ». *Computer-Aided Design and Applications,* vol. 9, n° 6, p. 771-794.

Brière-Côté, Antoine, Louis Rivest et Roland Maranzana. 2012b. « Integration of explicit geometric constraints in the comparison of 3D CAD models for part design reuse ». In *IFIP WG5.1 9th International Conference on Product Lifecycle Management*, sous la dir. de Rivest, Louis, Abdelaziz Bouras et Borhen Louhichi. Vol. 388, p. 445-457. New York: Springer.

Brière-Côté, Antoine, Louis Rivest et Roland Maranzana. 2013. « 3D CAD model comparison: An evaluation of model similarity and difference assessment technologies ». *Computer-Aided Design and Applications,* vol. 10, n° 2, p. 173-195.

Bronsvoort, W. F., R. Bidarra, M. Dohmen, W. van Holland et K. J. de Kraker. 1997. « Multiple-view feature modelling and conversion ». In *Proceedings of International Conference on Theory and Practice of Geometric Modelling, 14-18 Oct. 1996*. p. 159-74. Springer-Verlag. Consulté le 12 juin 2008.

Bronsvoort, W. F., R. Bidarra et P. J. Nyirenda. 2006. « Developments in feature modelling ». *Computer-Aided Design and Applications,* vol. 3, n^{o} 5, p. 655-64.

Brunetti, G., et S. Grimm. 2005. « Feature ontologies for the explicit representation of shape semantics ». *International Journal of Computer Applications in Technology,* vol. 23, n^{o} 2-4, p. 192-202.

C4W. 2008. *3D Shop ModelScan.*(Version 2.8.4). Logiciel de visualisation. Montpellier (France): C4W. < http://www.c4wviewer.com/accueil-fr.html >. Consulté le 22 octobre 2010.

CADCAM-E.Com, Inc. 2010. « CCE EnSuite: A Solution for Sharing Detailed Product Knowledge with Ease ». En ligne. < http://www.cadcam-e.com/products/ensuite2/ >.

Caplat, Guy. 2008. *Modèles et métamodèles*. Lausanne: Presses polytechniques et universitaires romandes.

CapVidia NV. 2010. *CompareVidia.*(Version 1.0.4019). Logiciel d'évaluation. Leuven, Belgium: CapVidia NV. < http://www.capvidia.com/products/comparevidia >. Consulté le 26 novembre 2010.

Cardone, Antonio, Satyandra K. Gupta et Mukul Karnik. 2003. « A survey of shape similarity assessment algorithms for product design and manufacturing applications ». *Journal of Computing and Information Science in Engineering,* vol. 3, n^{o} 2, p. 109-118.

CAx Implementor Forum. 2008. *Recommended practices for geometric validation properties*. Online. CAx Implementor Forum, 85 p. < http://www.cax-if.de/joint_testing_info.html#recpracs >.

Cerezuela, C., S. Limam et D. Riopel. 2001. « Processus de développement de produit et ingénierie concourante ». *Journal européen des systèmes automatisés,* vol. 35, n^{o} 6, p. 699-726.

Cessna Aircraft Company. 2010. « Cessna Supplier Guidelines and Requirements for Engineering Certification Projects ». En ligne. 27 p. < https://www.supplier.cessna.com/quality/requirements_for_engineering_projects.pdf >.

Chatelain, Jean-François, Roland Maranzana et Serge St-Martin. 2002. « A solid model comparison approach based on a model tree analysis ». In *4th WSEAS International Conference on Mechanical Engineering*.

Cheney, Doug. 2008. « 3D CAD Model Validation ». In *3D Collaboration & Interoperability '08*. (Denver, Colorado, May 15-16 2008). Consulté le 30 juin 2010.

Chowdhury, Sagar, et Zahed Siddique. 2009. « Development and implementation of shape comparison of 3D CAD models ». In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2009, DETC2009*, PART B. Vol. 2, p. 1085-1094. New York, NY: ASME.

Chu, Chih-Hsing, et Yung-Chang Hsu. 2006. « Similarity assessment of 3D mechanical components for design reuse ». *Robotics and Computer-Integrated Manufacturing,* vol. 22, n° 4, p. 332-341.

Cicchetti, A., D. Di Ruscio et A. Pierantonio. 2007. « A metamodel independent approach to difference representation ». *Journal of Object Technology,* vol. 6, n° 9, p. 165-185.

Cicchetti, A., D. Di Ruscio et A. Pierantonio. 2008. « Managing model conflicts in distributed development ». In *Model Driven Engineering Languages and Systems. 11th International Conference, MoDELS 2008, 28 Sept.-3 Oct. 2008*. (Berlin, Germany), p. 311-25. Coll. « Model Driven Engineering Languages and Systems. Proceedings 11th International Conference, MoDELS 2008 »: Springer-Verlag.

Cicirello, V., et W. C. Regli. 2001. « Machining feature-based comparisons of mechanical parts ». In *Proceedings International Conference on Shape Modeling and Applications*. p. 176-85. IEEE Computer Society. < http://dx.doi.org/10.1109/SMA.2001.923388 >. Consulté le 29 mai 2008.

CoCreate Software GmbH. 2008. *CoCreate Modeling : Personal Edition 2.0.*(Version 20.00). Logiciel d'évaluation. Unterschleissheim (Germany): Parametric Technology GmbH. < http://www.cocreate.com/products/PE2/ModelingPE2.aspx >.

CoCreate Software GmbH, et Steve Gutierrez. 3 juin 2010. *Method for comparing a first computer-aided 3D model with a second computer-aided 3D model*. En ligne US 2010/0135535. Consulté le 13 mai 2008.

CT CoreTechonologies Group. 2008. « 3D_Evolution Product Suite ». En ligne. < http://www.coretechnologie.de/p_enp.htm >.

Cuillière, Jean-Christophe, Vincent François, Khaled Souaissa, Abdelmajid Benamara et Hedi BelHadjSalah. 2009. « Automatic CAD Models Comparison and Re-meshing in the Context of Mechanical Design Optimization ». In *Proceedings of the 18th*

International Meshing Roundtable, sous la dir. de Clark, Brett W., p. 231-245. Berlin (Germany): Springer-Verlag. < http://www.imr.sandia.gov/papers/abstracts/Cu500.html >.

Cuillière, Jean-Christophe, Vincent François, Khaled Souaissa, Abdelmajid Benamara et Hedi BelHadjSalah. 2011. « Automatic comparison and remeshing applied to CAD model modification ». *Computer-Aided Design,* vol. 43, n° 12, p. 1545-1560.

Dassault Systèmes. 2007. *CATIA V5.*(Version R17 SP8.0). Logiciel commercial. Vélizy Villacoublay, France: Dassault Systemes. < http://www.3ds.com/products/catia/ >.

Dassault Systèmes. 2010. *SolidWorks 2010 : Premium Edition.*(Version 2010 SP4.0). Logiciel commercial. Vélizy Villacoublay (France): Dassault Systèmes. < www.solidworks.com >.

De Martino, T., B. Falcidieno et S. Hassinger. 1998. « Design and engineering process integration through a multiple view intermediate modeller in a distributed object oriented system environment ». *Computer-Aided Design,* vol. 30, n° 6, p. 437-52.

Del Fabro, Marcos Didonet, Jean Bézivin et Patrick Valduriez. 2006. « Weaving models with the Eclipse AMW plugin ». In *2006 Eclipse Modeling Symposium and Eclipse Summit Europe*.

Elysium Co., Ltd. 2010. « CADDoctor: Product Info ». En ligne. < http://www.elysium-global.com/productinfo/caddoctor/ >.

François, V., et J. C. Cuillière. 2000. « 3D automatic remeshing applied to model modification ». *Computer-Aided Design,* vol. 32, n° 7, p. 433-444.

Frechette, Simon P. 1996. *Interoperability Requirements for CAD Data Transfer in the AutoSTEP Project*. En ligne. NISTIR 5844. Gaithersburg, MD: U.S. Departement of Commerce, Technology Administration, 28 p. < http://www.mel.nist.gov/msidlibrary/doc/ReqCadProj96.pdf >. Consulté le 25 août 2010.

Geometric Ltd. 2010. « GeomDiff ». En ligne. < http://geomdiff.geometricglobal.com/ >.

Hoffmann, C. M., et R. Joan-Arinyo. 2000. « Distributed maintenance of multiple product views ». *Computer-Aided Design,* vol. 32, n° 7, p. 421-431.

Hoffmann, Christoph M., et R. Joan-Arinyo. 1998. « CAD and the product master model ». *Computer-Aided Design,* vol. 30, n° 11, p. 905-918.

Huang, Zhengdong, Shaopeng Tian et Ji Zhou. 2004. « Solving CSG equations for checking equivalency between two different geometric models ». *Computer-Aided Design,* vol. 36, n° 10, p. 975-92.

InnovMetric Software Inc. 2009. *PolyWorks: Inspector.*(Version 11.0). Logiciel commercial. Québec, QC (Canada): InnovMetric Softwares Inc. < http://www.innovmetric.com/polyworks/3D-scanners/home.aspx?lang=en >. Consulté le 10 décembre 2008.

International Organisation for Standardization. 2005. *Industrial automation systems and integration -- Product data representation and exchange -- Part 108: Integrated application resource: Parameterization and constraints for explicit geometric product models*. International standard, ISO 10303-108:2005. Geneva: International Organisation for Standardization.

International Organisation for Standardization. 2010. *Industrial automation systems and integration -- Product data representation and exchange -- Part 214: Application protocol: Core data for automotive mechanical design processes*. International standard, ISO 10303-214:2010. Geneva: International Organisation for Standardization.

International Organisation for Standardization. 2011. *Industrial automation systems and integration -- Product data representation and exchange -- Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies*. International standard, ISO 10303-203:2011. Geneva: International Organisation for Standardization.

Iraqi Houssaini, Mehdi, Mathias Kleiner et Lionel Roucoules. 2012. « Tools interoperability in engineering design using model-based engineering ». In *Proceedings of the 11th Biennial Conference on Engineering Systems Design and Analysis*. New York: ASME.

ITI TranscenData. 2010a. « CAD Model Validation - CADIQ ». En ligne. < http://www.transcendata.com/products/cadiq/ >.

ITI TranscenData. 2010b. « CADIQ 6.1 Product Specification: Release Summary ». En ligne. Milford, OH (USA): International TechneGroup Inc., 7 p. < http://www.transcendata.com/support/cadiq/CADIQ_6_1_Product_Specification.pdf >. Consulté le 26 novembre 2010.

Iyer, N., S. Jayanti, Lou Kuiyang, Y. Kalyanaraman et K. Ramani. 2005. « Three-dimensional shape searching: state-of-the-art review and future trends ». *Computer-Aided Design,* vol. 37, n° 5, p. 509-30.

Jackson, Chad, et Maura Buxton. 2007. *The design reuse benchmark report: Seizing the opportunity to shorten product development*. En ligne. Boston, MA: Aberdeen Group, Inc. < http://www.aberdeen.com/aberdeen-library/3908/DesRes_Out_3908.aspx >. Consulté le 1 avril 2008.

Kim, Junhwan, Michael J. Pratt, Raj G. Iyer et Ram D. Sriram. 2008. « Standardized data exchange of CAD models with design intent ». *Computer-Aided Design,* vol. 40, n° 7, p. 760-777.

Kolovos, D. S., D. Di Ruscio, A. Pierantonio et R. F. Paige. 2009. « Different models for model matching: an analysis of approaches to support model differencing ». In *2009 ICSE Workshop on Comparison and Versioning of Software Models*. p. 1-6. Coll. « 2009 ICSE Workshop on Comparison and Versioning of Software Models (CVSM 2009) ». Piscataway, NJ: IEEE. < http://dx.doi.org/10.1109/CVSM.2009.5071714 >.

Kripac, J. 1997. « A mechanism for persistently naming topological entities in history-based parametric solid models ». *Computer Aided Design,* vol. 29, n° 2, p. 113-122.

Kubotek USA. 2010. « Kubotek Validation Tool ». En ligne. < http://blog.kubotekusa.com/validatecad/ >.

Lattice Technology Co. 2010. *XVL Studio Professional.*(Version 9.1a). Logiciel d'évaluation. San Fransisco, CA: Lattice Technology inc. < http://www.lattice3d.com/products/products_studio_3d_software.html >. Consulté le 3 novembre 2010.

Lee, Kunwoo. 1999. *Principles of CAD/CAM/CAE systems*. Reading, Mass. ; Don Mills, Ont.: Addison-Wesley, xviii, 582 p. p.

Li, Baofu, et Jin Liu. 2002. « Detail feature recognition and decomposition in solid model ». *Computer-Aided Design,* vol. 34, n° 5, p. 405-14.

Li, Zhanjun, Min Liu et Karthik Ramani. 2004. « Review of product information retrieval: Representation and indexing ». In *2004 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Sep 28-Oct 2 2004*. Vol. 4, p. 971-979. American Society of Mechanical Engineers, New York, NY 10016-5990, United States. Consulté le 18 mars 2008.

Lockett, Helen, et Marin Guenov. 2008. « Similarity measures for mid-surface quality evaluation ». *Computer-Aided Design,* vol. 40, n° 3, p. 368-380.

Maurino, Michel. 1995. *La gestion des données techniques : technologie du concurrent engineering*. Coll. « Organisation industrielle ». Paris, France: Masson, xi, 188 p.

Missler Software. 2008. *TopSolid 2008.*(Version 6.9.200). Logiciel commercial. Evry (France): Missler Software. < http://www.topsolid.fr/ >.

Msaaf, Omar, Roland Maranzana et Louis Rivest. 2007. « Part data mining for information re-use in a PLM context ». In *Proceedings of GT2007*. New York: American Society of Mechanical Engineers.

Object Management Group. 2011. « MetaObject Facility ». In *Object Management Group Website*. < www.omg.org/mof >. Consulté le September 2011.

Office québécois de la langue française. 2011. « Le grand dictionnaire terminologique ». En ligne. < http://gdt.oqlf.gouv.qc.ca/ >. Consulté le 8 novembre 2012.

Oracle Corporation. 2010. *AutoVue Electro-Mechanical Professional : Desktop Version.*(Version 20.0). Logiciel d'évaluation. Redwood Shores, CA: Oracle Corporation. < http://www.oracle.com/us/products/applications/autoVue/autovue-electro-mechanical-professional/index.html >. Consulté le Octobre 2010.

Pan, Wanbin, Shuming Gao, Hua Zhu, Ming Li et Jinming Chen. 2011. « Identifying different entities for minor model modification based on common primary subpart ». *Computer-Aided Design and Applications,* vol. 8, n° 3, p. 345-356.

Parametric Technology Corporation. 23 septembre 2003. *Method of comparing parts*. Brevet américain US 6,625,607. < http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=US6625607&F=0&QPN=US6625607 >. Consulté le 20 mai 2008.

Parametric Technology Corporation. 2009. *PRO/Engineer Wildfire : Education Edition.*(Version 5.0). Logiciel commercial. Needham, MA: Parametric Technology Corporation. < http://www.ptc.com >.

Pottman, H., S. Leopoldseder et H. Zhao. 2003. *The d²-Tree: A Hierarchical Representation of the Squared Distance Function*. Tech. Rep. 101: Institute of Geometry, Vienna University of Technology. < http://www.geometrie.tuwien.ac.at/leopoldseder/t_rep101.pdf >. Consulté le 17 novembre 2008.

Pottmann, Helmut, Stefan Leopoldseder et Michael Hofer. 2004. « Registration without ICP ». *Computer Vision and Image Understanding,* vol. 95, n° 1, p. 54-71.

Prieto, F., T. Redarce, R. Lepage et P. Boulanger. 1999. « A non-contact CAD based inspection system ». In *Proceedings of the Conference on Quality Control by Artificial Vision, 18-21 May 1999*. p. 133-8. Canadian Image Process. & Pattern Recognition Soc. Consulté le 10 juillet 2008.

Purdue Center for Information Systems in Engineering. 2007. « Engineering Shape Benchmark ». In *3D Sketch-Based System for Conceptual Design*. En ligne. < http://shapelab.ecn.purdue.edu/Benchmark.aspx >.

Quintana, Virgilio. 2011. « Vers l'élimination des dessins d'ingénierie des processus de modification d'ingénierie en aéronautique ». Thèse de doctorat en génie. Montréal, Québec, École de technologie supérieure, XXII; 266 p.

Quintana, Virgilio, Louis Rivest et Robert Pellerin. 2012. « Measuring and improving the process of engineering change orders in a model-based definition context ». *International Journal of Product Lifecycle Management,* vol. 6, n° 2, p. 138-160.

Ramesh, M., D. Yip-Hoi et D. Dutta. 2001. « Feature Based Shape Similarity Measurement for Retrieval of Mechanical Parts ». *Journal of Computing and Information Science in Engineering,* vol. 1, p. 245.

Sheffer, A., et A. Ungor. 2001. « Efficient adaptive meshing of parametric models ». *Transactions of the ASME. Journal of Computing and Information Science in Engineering,* vol. 1, n° 4, p. 366-75.

Siemens PLM Software inc. 2009a. *NX.*(Version 7.0.1.7 MP1). Logiciel commercial. Plano (TX): Siemens PLM Software inc. < http://www.plm.automation.siemens.com/fr_fr/products/nx/index.shtml >.

Siemens PLM Software inc. 2009b. *Solid Egde with Synchronous Technology 2.*(Version 102.00.07.04). Logiciel de CAO. Plano, TX: Siemens PLM Software. < http://www.plm.automation.siemens.com/en_us/products/velocity/solidedge/index.shtml >. Consulté le 28 novembre 2011.

Siemens PLM Software inc. 2011a. « Geolus Search ». In *PLM - Product Lifecycle Management: Siemens PLM Software*. En ligne. < http://www.plm.automation.siemens.com/en_us/products/open/geolus/index.shtml >.

Siemens PLM Software inc. 2011b. *Solid Egde with Synchronous Technology 4.*(Version 104). Logiciel de CAO. Plano (TX): Siemens PLM Software. < http://www.plm.automation.siemens.com/en_us/products/velocity/solidedge/index.shtml >. Consulté le juillet 2013.

Souaissa, Khaled, Jean-Christophe Cuillière, Vincent François, Abdelmajid Benamara et Hedi BelHadjSalah. 2010. « Automatic modification retrieval between CAD parts ». *Mécanique & Industries,* vol. 11, n° 2, p. 85-92.

SpaceClaim Corporation. 2011. *SpaceClaim Engineer 2011+.*(Version 2011.1 SP1). Logiciel CAO, direct modeling. Concord, MA: SpaceClaim Corporation. < www.spaceclaim.com >. Consulté le 28 novembre 2011.

St-Martin, Serge. 2001. « Maîtrise du changement dans les modèles CAO ». Mémoire de maîtrise en technologie des systèmes en ligne. Montréal, École de technologie supérieure., 102 p. In ProQuest Dissertations and Theses. Ann Harbour, MI : ProQuest LLC. < http://proquest.umi.com/pqdlink?did=729280741&Fmt=7&clientId=46962&RQT=309&VName=PQD >. Consulté le 3 juin 2008.

Subramani, S., et B. Gurumoorthy. 2005. « Maintaining associativity between form feature models ». *Computer-Aided Design,* vol. 37, n^o 13, p. 1319-1334.

Synergis Software. 2010. « Adept ». Synergis Software. < http://www.synergissoftware.com >. Consulté le December 2010.

Sypkens Smit, Matthijs, et Willem F. Bronsvoort. 2007. « The difference between two feature models ». *Computer-Aided Design and Applications,* vol. 4, n^o 6, p. 843-851.

Sypkens Smit, Matthijs, et Willem F. Bronsvoort. 2009a. « Efficient tetrahedral remeshing of feature models for finite element analysis ». *Engineering with Computers,* vol. 25, n^o 4, p. 327-44.

Sypkens Smit, Matthijs, et Willem F. Bronsvoort. 2009b. « Integration of design and analysis models ». *Computer-Aided Design and Applications,* vol. 6, n^o 6, p. 795-808.

Tangelder, J. W. H., et R. C. Veltkamp. 2004. « A survey of content based 3D shape retrieval methods ». In *Proceedings. Shape Modeling International 2004, 7-9 June 2004*. p. 145-56. IEEE Comput. Soc.

Tarbox, G. H., S. N. Gottschlich et L. A. Gerhardt. 1993. « Registration of dissimilar featureless models for inspection ». In *Proceedings of 1993 IEEE International Conference on Robotics and Automation, 2-6 May 1993*. p. 686-91. IEEE Comput. Soc. Press. < http://dx.doi.org/10.1109/ROBOT.1993.292058 >.

Theorem Solutions. 2010. « Data Exchange Automation via Theorem Process Manager (TPM) ». En ligne. < http://www.theorem.co.uk/products/tpm.htm >.

Translation Technologies Inc. 12 décembre 2006. *Geometric model comparator and method.* Brevet américain US 7,149,677. < http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=US2003135846&F=0 >. Consulté le 21 mai 2008.

Translation Technologies Inc. 2010. « Mirror Model Comparator ». In *Translation Technologies: CAD File Translation.* En ligne. < http://www.translationtech.com/mmc.asp >.

Tremblay, Thomas Guillaume, Louis Rivest, Omar Msaaf et Roland Maranzana. 2006. « The role of associations in CAD and PLM for handling change propagation during product development ». In *Proceedings of the 2006 conference on Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering*. p. 507-514. IOS Press.

Tsai, Y. R. 2002. « Rapid and accurate computation of the distance function using grids ». *Journal of Computational Physics,* vol. 178, nº 1, p. 175-95.

Vezzetti, Enrico. 2009. « Product lifecycle data sharing and visualisation: Web-based approaches ». *International Journal of Advanced Manufacturing Technology,* vol. 41, nº 5-6, p. 613-630.

Viswanathan, Karthik, Sagar Chowdhury et Zahed Siddique. 2008. « Shape comparison of 3D models based on features and parameters ». In *2008 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC 2008, August 3, 2008 - August 6, 2008*. Vol. 3, p. 647-657. New York City, NY, United states: ASME.

Wei, L., et H. Yuanjun. 2008. « Representation and retrieval of 3D CAD models in parts library ». *International Journal of Advanced Manufacturing Technology,* vol. 36, nº 9-10, p. 950-958.

Wolfe, L. Stephen. 2006. « Shape Indexing May Help Part Reuse ». *CADCAMNet*. < www.cadcamnet.com >. Consulté le September 14, 2006.

Wu, J., T. Zhang, X. Zhang et J. Zhou. 2001. « A face-based mechanism for naming, recording and retrieving topological entities ». *Computer Aided Design,* vol. 33, nº 10, p. 687-698.

Yang, Yubin, Hui Lin et Yao Zhang. 2007. « Content-based 3-D model retrieval: A survey ». *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews,* vol. 37, nº 6, p. 1081-1098.

Zhang, Tong, et Qingjin Peng. 2009. « Shape distribution-based 3D shape retrieval methods: Review and evaluation ». *Computer-Aided Design and Applications,* vol. 6, nº Compendex, p. 721-735.

Zuray, Rick, et Jean-Yves Delaunay. 2009. « LOTAR 101: Overview of Long Term Archival and Retrieval of digital product & technical data ». En ligne. 134 p. < http://www.slideshare.net/zurayrs/LOTAR101OverviewCurrentJan2009 >. Consulté le 12 août 2010.