الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de fin d'études Master

Filière : Informatique

Option : Ingénierie des Medias

Thème :

La nomination persistante sur des critères sémantiques

Encadré par :

Dr. Hakim SOUSSI

Présenté par :

Bahloul Mohamed amine

Kemerchou Brahim

Juin 2013

Remerciement

Nous tenons tout d'abord à remercier notre encadreur Dr. Hakim SOUSSI, qui nous a apporté son expertise avec la plus grande disponibilité tout au long de l'année et ses conseils et pertinentes remarques, sans lui ce travail n'aurait pas vu le jour.

Nous remercions également les membres du jury qui nous font honneur de juger notre travail.

Nous adressons également nos remerciements, à tous nos enseignants, qui nous ont donnés les bases de la science, et qui ont assisté à nos débuts en informatique pour leurs précieux conseils.

Enfin, nous adressons à nos plus sincères remerciements à tous nos proches, amis et collègues qui nous ont toujours soutenus et encouragés au cours de nos études.

Résumé :

L'utilisation des modeleurs paramétriques en CAO offre plusieurs atouts qui peuvent aider le concepteur dans la conception ainsi que dans la réévaluation des processus. Un bon système de nomination persistante est le pilier essentiel sur lequel repose un modeleur performant. Malheureusement et malgré les différents études entamées depuis plusieurs années, on ne peut pas affirmer qu'il existe aujourd'hui un système de nomination persistante qui soit assez robuste pour permettre à un modeleur d'assister le concepteur de manière efficace dans la conception des objets.

La voie explorée dans cette étude comporte deux tâches de base : Premièrement, offrir à l'utilisateur concepteur tous les résultats possibles lors de phase de réévaluation tout en essayant de préserver la topologie initiale de l'objet. Deuxièmement, classer ces résultats selon un ordre de préférence fondé sur la modification apportée par l'utilisateur à l'objet initial, c'est-à-dire que l'ordre de ces résultats change selon la modification ainsi que l'effet de cette dernière sur la topologie de l'objet.

Ce document s'efforce de montrer qu'une telle approche est incontournable pour améliorer l'assistance à la modélisation paramétrique. En effet, tenter de se limiter à une seule solution est insuffisant car plusieurs utilisateurs pourraient donner des résultats différents, ainsi le résultat obtenu pourrait ne pas être celui voulu par l'utilisateur.

MOTS-CLES. Cao, nomination persistante, modélisation paramétrique, critères sémantiques.

Sommaire

Introduction générale:

Au fur et à mesure que les modeleurs CAO (Conception Assistée par Ordinateur) évoluent et afin de satisfaire les critères sémantiques lors des phases de réévaluations sans avoir à reconstruire de nouveaux le processus, la notion de nomination persistante sur critères sémantiques a émergé.

Dans les années 70, les logiciels de CAO n'étaient que des logiciels de DAO (Dessin Assisté par Ordinateur). Ces derniers qui sont des systèmes de modélisation statiques, ne permettent pas une modification par réévaluation. Petit à petit il y a eu une évolution grâce, d'une part à l'augmentation des performances du matériel informatique, et d'autre part à la recherche dans le domaine du logiciel. Mais, aujourd'hui, ils ne sont pas encore véritablement des systèmes d'aide à la conception. Ces systèmes connus sous le nom de modeleurs dynamiques, sont plus encore connus sous le terme de **modeleurs paramétriques**.

Le principe de la modélisation paramétrique est d'enregistrer le processus de conception. Contrairement aux systèmes de modélisation statiques, les modeleurs dynamiques permettent la modification rapide par réévaluation. Cependant, les modifications trop fortes de la topologie donnent parfois des résultats différents de ceux attendus.

L'un des problèmes principaux pour la réévaluation paramétrique est de caractériser les entités géométriques et topologiques d'un modèle paramétrique en leur donnant un nom lors de la conception et à retrouver à quoi correspond ce nom lors de la réévaluation. Un autre problème concerne l'aspect sémantique, qui est relatif aux intentions du concepteur. En effet pour satisfaire ces derniers, un système de nomination robuste aux modifications topologiques s'avère nécessaire pour préserver, d'une réévaluation à l'autre, les références sur les entités topologiques. Ce problème est connu sous le nom de **nomination persistante** ou **nomination topologique**.

Les principales techniques de nomination persistante élaborées jusqu'à aujourd'hui, ne garantissent pas les critères sémantiques imposés par le concepteur lors de la phase de conception. Le problème qui se pose est donc le suivant ; Comment faire évoluer les techniques de nominations persistantes actuelles afin qu'elles puissent garantir, d'une part l'identification lors de la réévaluation, des entités nommées lors de la phase de construction, et cela de manière unique et déterministe, d'autre part, d'obtenir le résultat attendu par le concepteur.

Dans le but de répondre à ces questions, une étude est entamée. Cette étude se base sur le plan de travail suivant :

Le premier chapitre sera consacré à un historique concernant l'évolution de la CAO et sa contribution actuelle.

Dans le deuxième chapitre on présentera notre modèle lequel est basé sur des heuristiques permettant de prendre en compte un certain nombre d'intentions du concepteur lors de la phase de réévaluation. On verra aussi dans ce chapitre la partie expérimentation dans laquelle on exécutera quelques exemples via une application qui utilise les principes de notre modèle.
Finalement on clôturera ce travail avec une conclusion ainsi que quelques perspectives susceptibles d'améliorer ce travail.

CHAPITRE 1

La Bibliographie

Introduction :

La conception assistée par ordinateur (CAO) ou computer aided design (CAD) en Anglais est l'utilisation des systèmes informatiques pour aider à la création, la modification, l'analyse et l'optimisation d'une conception. Les logiciels de la CAO sont utilisés pour augmenter la productivité du concepteur et améliorer la qualité de la conception.

Généralement l'atteinte d'une solution ou d'un design n'est pas directe sauf pour des problèmes extrêmement simples. Le processus est plutôt itératif car on distingue d'abord le choix d'un modèle représentant le phénomène physique du problème et ensuite, un premier design est élaboré et, on vérifie si les contraintes sont satisfaites. On modifie le design et on répète jusqu'à ce que le design vérifie les contraintes et nous mène à la solution parfaite pour notre problème.

La CAO est un art industriel largement utilisés dans de nombreux domaines, l'automobile, la construction navale et les industries de l'aérospatiale, du design industriel et architectural, les rothèses, et beaucoup d'autre. La CAO est aussi largement utilisée pour produirc dc l'animation par ordinateur pour les effets spéciaux dans les films, la publicité et les manuels techniques, souvent appelé DCC (Digital Content Creation en Anglais) ou création de contenu numérique. L'omniprésence moderne et la puissance des ordinateurs signifie que même les bouteilles de parfum et shampoing sont conçues en utilisant des techniques inconnues par les ingénieurs des années 1960. En raison de son importance économique énorme, la CAO a été une force motrice majeure pour la recherche dans la géométrie algorithmique, l'infographie (matériel et logiciel) et la géométrie différentielle discrète [Pottmann].

Un logiciel de CAO pour la conception mécanique utilise des graphiques à base de vecteurs pour représenter les objets de dessin traditionnel, et peut également produire des images tramées montrant l'aspect global des objets conçus. Cependant, il s'agit plus que de simples formes. Comme dans la rédaction manuelle des dessins techniques et d'ingénierie, la sortie du CAD doit transmettre de l'information, tels que les matériaux, les procédés, les dimensions et tolérances, selon des conventions spécifiques à l'application.

La CAO peut être utilisée pour les courbes de conception en deux dimensions (2D) ou encore dans l'espace à trois dimensions (3D) [FarinG].

Bien que l'objectif des systèmcs CAO automatisés est d'accroître l'efficacité, ils ne sont pas nécessairement la meilleure façon de permettre aux nouveaux utilisateurs de comprendre les

principes de la géométrie et de la modélisation solide. Pour cela, les langages de script tels que Plasma (langage de programmation de la modélisation solide) , qui constitue le fondement de la conception assistée par ordinateur et des systèmes de CAO, sont plus adaptés puisque, contrairement à d'autres programmes de CAO, c'est un langage de script open source qui se concentre plus sur les scripts plutôt que l'interaction avec l'interface graphique, alors les utilisateurs peuvent créer des designs arbitrairement complexes en utilisant un grand nombre d'objets simples en 2D et 3D, des courbes et des surfaces courbes avancées, les opérations booléennes, et élémentaire ainsi que des transformations géométriques avancées.

1.1 Bref historique :

En suivant l'évolution des systèmes CAO, on constate qu'elles passent de simples outils de maquette à des systèmes graduellement complexes qui schématisent la géométrie des objets, en ajoutant et en conservant des informations qui accompagnent la modélisation et la conception des objets visés.

Le traitement informatique des informations et des données débute la fin des années 1950 et le début des années 1960. Comme mentionné ci-dessus, les premiers systèmes CAO étaient des simples systèmes de maquette, plan et représentations visuelles et graphiques. Le développement des modèles et systèmes CAO, associés à des outils typiques, ont été influencés par quatre domaines [Requicha 80] :

i) Les machines à commandes numériques: apparues au début des années 1950 au MIT. Ce sont des modèles de représentation numérique de pièces mécaniques capables de piloter des machines à commandes numériques.

ii) L'industrie aéronautique et automobile: parmi les premiers qui ont utilisés les machines à commande numérique. On cite les travaux de Bézier (Renault), De Casteljau (Citroën), Coon (Ford) et Ferguson (Boeing).

iii) L'infographie-: l'importance de manipuler visuellement et graphiquement les objets mènent à les représenter sur ordinateur (par exemple, Sketchpad développé par Sutherland en 1963 [Sutherland 63]).

iv) Les méthodes d'éléments finis: Les générateurs automatiques et semi-automatiques de maillage sont apparus en 1970. C'est un domaine développé indépendamment des autres domaines.

Le passage des modèles 2D aux modèles 3D ne sont pas fait selon une échelle de difficultés, puisque il y'a eu des travaux qui sont entamés en parallèle. La fin des années 1960 et le début des années 1970 ouvre la porte aux systèmes 3D qui succèdent aux travaux de Sutherland [Sutherland 63] pour les systèmes de dessin interactif en 2D. Alors, on a vu une immense évolution, du modèles géométriques BRep (Boundary Representation), CSG (Construction Solid Geometry) aux modeleurs à base de caractéristiques de forme permettant d'intégrer la conception au processus de fabrication, puis la naissance d'algorithmes d'extraction de caractéristiques de forme à partir de modèles géométriques, et l'introduction aux surfaces de formes libres dans les modèles BRep ainsi que l'utilisation de NURBS (Non-Uniform Rational B-Spline) entre 1970 et 1980 . En arrivant à la fin des années 1980 et le début des années 1990 connaissent le développement de différents prototypes de systèmes de modélisation par caractéristiques de forme dans les laboratoires de recherche universitaires et dans les bureaux d'études du monde industriel. La dernière décennie du XXème siècle est caractérisée par la représentation des objets non-rigides (fluides, nuages, objets élastiques, etc.) ainsi que l'installation et l'implémentation commerciale des systèmes de modélisation paramétrique.

A partir de XXI siècle, on se dirige vers la modélisation produit (product modeling), une représentation dans une structure globale des données qui concerne le produit et tous les détails des étapes et phases de son cycle de vie. Cette approche est caractérisée par la description du produit par des différents attributs portant des informations géométriques, fonctionnelles, matérielles, objectives, etc., qui permet l'extraction du produit convenable à chaque métier, profession et objectif [Agbod02].

1.2 Contribution de la CAO :

Dû à l'évolution de l'industrie, de l'économie, les contraintes de la vie public et les entités et organismes gouvernementaux, l'industrie qui cherche à rester compétitive se trouve face à un grand défi : la production des meilleurs produits à meilleur prix et l'augmentation de productivité des ingénieurs.

L'informatique a prouvé sa grande efficacité en offrant une économie dans le phases de conception assistée par ordinateur :

i) Création d'un modèle :

En utilisant les outils et systèmes disponibles actuellement sur le marché, on peut faire la création géométrique de n'importe quel objet aisément. En plus, on peut étudier l'objet de différents angles et extraire une copie de notre objet dans n'importe quelle phase ou étape de conception.

ii) Analyse :

Les caractéristiques de l'objet, une fois créé, sont immédiatement disponibles et prêtes pour des programmes d'analyse ou de simulation (éléments finis, vibrations, réponses en fréquence..) et, par conséquence, l'utilisateur reçoit les résultats de ces calculs sous forme graphique pour évaluer si l'objet est conforme aux contraintes.

iii) Modification :

Après l'analyse et la représentation graphique, la tâche de modification apportée à notre objet ou modèle sera significativement plus simple, claire et facile.

Donc, avec la CAO on peut considérer plusieurs alternatives et solutions et choisir la meilleure solution. La CAO accompagne la progression de la profession des ingénieurs pour avoir le meilleur résultat. En donnant un exemple de domaine des structures, les programmes d'analyses sont fiables, précis et complet graduellement, et l'analyse et le suivi de du comportement des éléments et des objets est plus fiable et pertinent comparé aux formules non scientifiques utilisées auparavant. Donc, l'optimisation du design est obtenue par l'utilisation itérative des outils CAO et révéler les comportements des objets tout au cours de toutes les phases de conception au lieu de les révéler seulement quand on présente le prototype ou le produit final.

Cette technique est adoptée depuis plusieurs années par des différents domaines technologiques hautement développé comme l'aéronautique, les activités et projets nucléaires, etc., puisque les méthodes traditionnelles ne sont plus utilisées. L'évolution de ces nouvelles techniques est étonnamment rapide et il y'a une tendance qu'elles seront déployées dans la majorité des différentes sociétés et entreprises.

Malgré les avantages de la CAO cités ci-dessus, les modèles géométriques statiques actuellement utilisés sont consacrés au seul but de la représentation et de la visualisation graphique des objets. Comme ces modèles sont dépourvus de capacité de la prise en charge de vœux et intentions de concepteur, et la capacité de la gestion de l'usinage des différentes pièces, ces modèles sont munis des informations additionnelles de fonction intégrées dans les caractéristiques. Ces nouveaux modèles à base de caractéristiques, particulièrement les caractéristiques de forme offre un vocabulaire de plus haut niveau (rainure, bossage, arrondi, etc.) et plus étendu (tolérance, matériau, métrologie, etc.) que les simples entités géométriques (points, cercle, cylindre, etc.) pour la description des étapes de construction des objets. Cette description, qui combine la géométrie, les paramètres, les contraintes, etc., et qui sert à la réutilisation dans le contexte de variations de paramètres et de la réévaluation des modèles géométriques, on a une représentation qui s'appelle modèle paramétrique.

Un modèle paramétrique regroupe la représentation géométrique et/ou topologique d'un objet, un ensemble de paramètres (caractéristiques) et un ensemble des contraintes (fonctions ou équations) qui s'applique sur l'objet. Alors, un modeleur paramétrique est un système de conception géométrique qui combine la géométrie de l'objet, plus la liste ou l'ensemble des commande qui nous permet à le créer. Autrement dit c'est le processus de conception, les gestes de construction ou encore de la spécification paramétrique.

Les systèmes paramétriques regroupent des systèmes de modélisation dynamique appelés "history-based", "constraint-based" ou encore "feature-based", qui remplacent les modèles géométriques statiques utilisés en CAO (BRep, CSG, etc.) et permettent à la fois d'exprimer et de mémoriser l'intention du concepteur, en plus des fonctions originales des systèmes géométriques qui permettent la représentation graphique des objets.

Un tel système paramétrique qui prend en charge l'intention du concepteur permet la modification rapide, instantané du modèle par le processus de réévaluation. Puisque la réévaluation cause des modifications topologiques, les références entres les entités exploitées dans la phase de conception sont souvent réévaluées de façon incorrecte, ce qui donne des résultats différents de ceux prévus. Ce problème qui porte le nom de **"nomination persistante"**

ou **"topological naming"** nécessite l'existence d'un système de nomination persistante vigoureux pour la préservation, d'une réévaluation à une réévaluation, des références entres les entités topologiques [Kripa95, Capoy96].

Le système de nomination persistante doit permettre une clarté non ambiguë dans l'identification des entités géométriques et topologiques du modèle paramétrique (1er problème) afin de nous permettre de les repérer dans le modèle réévalué (2ème problème), et doit nécessairement prendre en charge les intentions sémantiques exprimés par le concepteur (3ème problème) [Agbod99].

2. Problème de nomination persistante :

Notre problème consiste à identifier les entités géométriques et topologiques référencées par les contraintes. Cette indentification qui est le premier aspect de la nomination consiste à la caractérisation des entités en leur donnant un nom lors de la conception afin de pouvoir les « repérer » dans le modèle lors de la réévaluation.

En général, la caractérisation, c'est-à-dire le nom, peut être un numéro de création, un pointeur vers la géométrie, le nom de la fonction de construction, etc. Dans un modèle non-paramétrique, il suffit que le nom existe au moment de la désignation d'une entité, tandis que dans un modèle paramétrique ce n'est pas seulement la forme finale qui est conservée, mais tout le processus de construction. Toutes les entités désignées sont référencées dans la spécification paramétrique et doivent donc porter un nom, même si elles n'existent plus dans l'instance courante.

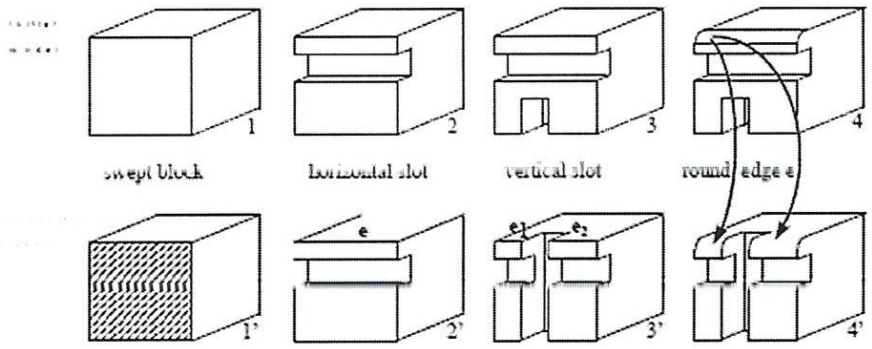


Figure 1.1 : *Problème de nomination [Agbod99].*

Pour montrer l'importance et la nécessité de l'identification par des noms, on prend l'exemple de la figure 1.1 ou on constate que le quatrième geste constructif est obtenu par l'arrondissement de l'arête *e*. Si le modèle initial est stocké ou bien échangé après cette quatrième étape, l'instance courante ne contient plus l'arête *e* : elle a été supprimée par la fonction d'arrondi. Donc la fonction d'arrondi qui a l'arête *e* comme paramètre d'entrée ne peut plus être représentée dans la spécification paramétrique du modèle par un pointeur sur la géométrie.

En plus, et en comparant les modèles CSG et BRep, on peut distinguer que les entités de CSG se sont situées totalement les unes par rapport aux autres et que chaque geste constructif produit une seule entité qui peut être désignée et nommée par la fonction de construction, tandis que le modèle BRep se caractérise par une structure hiérarchique qui doit être respectée et chaque geste constructif produit un nombre multiple, non-limité, non prédictible d'entités. Donc il est de grande nécessité d'avoir une structure des noms et un système ou un mécanisme de nomination pour identifier et nommer chaque entité de manière unique et non-ambiguë même si l'instance courante contient toutes les entités.

On conclut que les noms sont un aspect primordial dans l'identification des entités géométriques et topologiques dans les modèles paramétriques et ne peuvent pas être des simples pointeurs temporaires vers les entités du modèle.

Le deuxième aspect du problème est que le système de nomination doit être suffisamment puissant pour permettre d'effectuer un appariement « matching » robuste entre les entités du modèle initial et celles du modèle réévalué.

Pour illustrer ce problème, on revient à notre exemple de la figure 1.1, et plus précisément au modèle réévalué. On remarque qu'à l'étape 3, l'arête *e* a été coupée en deux arêtes e_1 et e_2. Mais à l'étape suivante, l'arête *e* doit être arrondie et ça nous donne 4 cas; sur quelle arête on doit appliquer l'arrondissement ?

(1) $e \leftrightarrow e_1$ & e_2 ; ou bien (2) $e \leftrightarrow e_1$; ou bien (3) $e \leftrightarrow e_2$; ou encore (4) $e \leftrightarrow \emptyset$.

Notre troisième principal problème que le système de nomination persistante doit y répondre, est la capacité de représenter et d'exprimer différentes sémantiques dans un même résultat (même instance courante) en utilisant des paramètres de haut niveau (agglomérat d'entités géométriques/topologiques) [Agbod02].

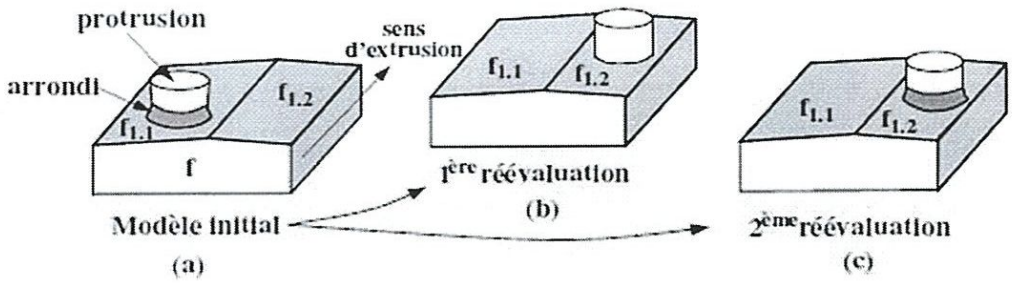


***Figure 1.2** : Différence de sémantique [Babali06].*

Pour illustrer ce dernier problème on considère l'exemple de la figure 1.2, où on trouve la fonction d'arrondi exprimée de deux façons, soit entre le cylindre et la face $f_{1.1}$, ou bien entre le cylindre et la coque composée des faces $f_{1.1}$ et $f_{1.2}$. Donc le modèle réévalué est différent puisque l'expression de geste constructif dans la spécification paramétrique est différente. Autrement dit, la **sémantique** du geste constructif n'est pas unique, mais elle varie.

Lorsque notre système traite les trois problèmes cités ci-dessus, on peut dire que c'est un système de nomination persistante sémantique vigoureux et robuste. Et ça ce qu'on va étudier en détail, avec quelques problèmes et solutions proposés (approche de Kripac, approche de Chen, etc.).

3. Solutions et approches proposés :

Plusieurs approches ont été proposés pour le problème de nomination persistante dans les modèles paramétriques qui sont caractérisés par des modèles de structure à deux niveaux puisque ils ont deux aspect : l'aspect instance représenté par la géométrie et la spécification paramétrique

qui englobe l'ensemble des contraintes qui guident et représentent l'historique gestes de construction du modèle. La modélisation paramétrique a été étudiée pendant plus de trois décennies. Un travail précurseur est celui de Hoffman et Juan [HJ92]. Parallèlement, plusieurs auteurs ont étudié la structure interne du modèle de données paramétriques. Ils ont contribué à l'apparition de différentes représentations [SV95, PAAB+96] et de représenter clairement leurs structures mathématiques sous-adjacentes [RS98, WN05]. D'autres ont présenté et discuté divers problèmes liés soit à la sémantique des opérations de modélisation [Che95, Mar06] ou à la gestion des contraintes de modélisation [BFH+95]. Parmi les approches proposées pour résoudre le problème de nomination persistante on trouve l'approche de Kripac, l'approche de Chen… etc.

3.1. Approche de Kripac :

Kripac [Kripa94a] a traité le problème d'appariement ou du matching des noms avec son API qu'il a proposé qui encapsule son système d'identification topologique et qui garantit la persistance des noms à l'aide d'une table de correspondance entre une entité du modèle initiale et ses entités correspondantes du modèle réévalué. Il a proposé une structure pour identifier les entités topologiques selon l'historique des faces et un algorithme de matching pour retrouver et repérer les entités après modification.

Parmi les désavantages et les limitations de l'approche de Kripac sont le mécanisme de matching proposé établit dans certains cas des règles de priorités tout à fait indépendante de l'action du concepteur, ce que rend le résultat de la réévaluation imprévisible. L'utilisation non optimale de la mémoire puisque il conserve une copie du modèle géométrique à chaque étape du processus de construction et plus considérablement il ne prend pas en charge le problème de la représentation sémantiques.

3.2. Approche de Chen :

Son modèle [Chen95a] est composé de deux représentations. La première est appelée Erep [Hoffm93a], non évaluée et indépendante du modeleur géométrique et basée sur un format textuel neutre. Par contre la deuxième représentation est évaluée et dépendante du modeleur géométrique et contient la géométrie ou l'instance courante par un schéma de nom qui forme un

lien entre les références géométriques du modèle géométrique et les noms génériques persistants du modèle non évalué.

Il a étudié la nomination dans la phase de construction du modèle paramétrique et le mécanisme de matching n'est pas suffisamment clair et défini.

3.3. Approche de Agbodan :

Etant donné que l'approche d'Agbodan est parmi les plus récentes qui utilisent la sémantique, on lui donne plus d'attention. Bien que son approche soit à la base relativement similaire à celle de Kripac, elle définit un mécanisme de nomination permettant de résoudre simultanément les trois problèmes énoncés en section 2. Pour définir de tels noms robustes, deux types d'entités géométriques et topologiques ont été distinguées dans [Agbod99] :

- Les **entités invariantes**. Une entité invariante est une entité géométrique ou topologique qui peut être, complètement et sans ambiguïté, caractérisée par la structure d'un geste constructif et de ses paramètres d'entrée, indépendamment des valeurs impliquées. Dans la Figure 1, les entités invariantes incluent la face d'extrémité du bloc extrudé, la coque latérale du slot horizontal avec ses face initiale et finale (qui peuvent ou ne pas exister), la face résultant de l'arrondi, etc. Pour caractériser, c'est-à-dire "nommer", de telles entités, il faut définir un mécanisme de nomination qui associe ces entités aux gestes constructifs et à leurs paramètres d'entrées.

- Les **entités contingentes**. Outre ces entités invariantes, il existe des entités qui dépendent du contexte d'un geste constructif. Nous appelons entité contingente une entité géométrique ou topologique qui résulte d'une interaction entre le modèle géométrique existant et les entités invariantes résultant d'un geste constructif particulier.

  Par exemple, dans la Figure 1.1, le nombre de faces latérales du slot vertical dans le modèle initial (étape 3) et dans le modèle réévalue (étape 3) n'est pas identique. Un mécanisme de nomination est donc également nécessaire pour définir les noms ces entités contingentes. Nous proposons donc d'identifier, de manière unique et non ambiguë, d'abord les entités invariantes (voir section 3.3.4.1), puis les entités contingentes (voir

section 3.3.4.2) en fonction des invariants. Pour chaque caractéristique de forme, nous allons extraire certains invariants et établir une classification (voir section 3.3.1), ce qui nous permettra de représenter les caractéristiques de forme comme des agrégats de coques (afin de répondre au problème d'expression de différentes sémantiques). Ces agrégats de coques représenteront les entrées du graphe des coques (voir section 3.3.2).

Nous tracerons l'évolution des coques à l'aide de ce graphe ce qui nous permettra d'identifier les coques contingentes.

3.3.1. Classification des caractéristiques de forme :

Afin, d'une part, de faciliter l'identification des entités lors de la réévaluation, et, d'autre part de permettre la représentation de différentes sémantiques, il est possible d'extraire un certain nombre d'invariants contenus dans les caractéristiques de forme. En effet, ces invariants ne changent pas d'une instanciation à l'autre ou d'une réévaluation à l'autre et sont donc facilement identifiables dans le modèle. La plupart des classifications actuelles de caractéristiques de forme proposent d'interpréter les données CAO d'un point de vue usinage ou d'un point de vue métier. Ces classifications ne dégagent pas d'invariants structurels et ne sont donc pas adaptées à notre problème.

L'idée est de proposer une classification des caractéristiques de forme basée, non pas sur des caractéristiques de forme métiers, mais plutôt sur leur structure géométrique invariante intrinsèque. Il s'agit à la fois de la structure initiale de la caractéristique de forme avant sont immersion dans la géométrie, mais aussi de la structure issue de l'interaction avec cette géométrie. Par exemple, il est considéré comme invariant le fait que systématiquement la face terminale d'un "trou débouchant" disparaisse lors de l'interaction avec l'objet dans lequel ce "trou débouchant" est effectué. La structure invariante, dépend de la manière dont la caractéristique de forme est conçue. Par exemple, pour l'extrusion et pour la révolution les invariants sont la coque initiale (ci), la coque latérale (cl) et la coque finale (cf).

La coque latérale (compte tenu de la topologie particulière du contour extrudé) peut être structurée en sous-coques (cl1, cl2, ..., cln) qui elles-mêmes peuvent être structurées. Par exemple, en coque latérale droite, gauche et fond (cld, clg, clfd) pour un "crossing_flat_slot". Le cas du "crossing_flat_slot" représenté sur la Figure 1.3 illustre bien la structure invariante d'une caractéristique de forme dont certaines entités (ci, cl1, cf) disparaissent systématiquement

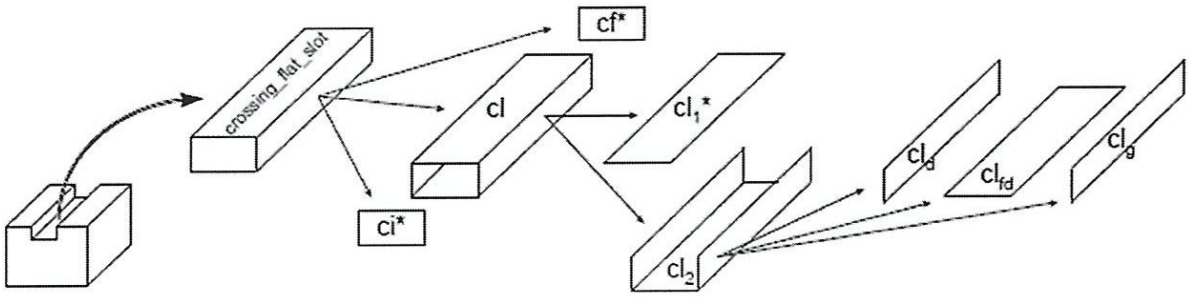


Figure 1.3 : *Exemple de structure invariante de caractéristique de forme : Crossing_flat_slot [Agbod99].*

Pour réaliser cette classification, nous nous sommes basé sur des classifications de caractéristiques de forme proposées par Shah et Mäntylä [Shah95] et celle proposée dans la norme STEP (AP 224) dont laquelle des caractéristiques de forme sont classées suivant les invariants qu'il a pu extraire pour chacune d'elles (cf Figure 1.4).

Il est possible de distinguer deux catégories de caractéristiques de forme, selon qu'elles peuvent ou non s'exprimer en fonction d'invariants de coques. Il a distingué, par rapport aux invariants de coques, 4 classes de caractéristiques de forme :

- **Les primitive features** : Ce sont des primitives de construction (bloc, cylindre, etc.) qui ne sont définies ni par extrusion, ni par révolution. Elles sont caractérisées par leur coque globale et leurs faces.

- **Les transition features** : Ce sont les caractéristiques de forme qui joignent plusieurs coques telles que le chanfrein et l'arrondi. Elles sont caractérisées par leur coque globale.

- **Les basic volume features** : Elles sont caractérisées par leur coque initiale, latérale et finale. Ces caractéristiques de forme sont issues d'une extrusion ou d'une révolution. Elles sont classées suivant le résultat de leur interaction avec la géométrie. Par exemple

un trou non débouchant ("blind_hole"), obtenu par extrusion d'un cercle, est caractérisé par ces trois coques (ci, cl, cf) et par le fait que la coque initiale disparaît dans la géométrie.

- **Les basic surface features** : Elles sont caractérisées par leur fil de fer initial et final et leur coque latérale. Ce sont toutes les surfaces obtenues par balayage d'un fil de fer. A ces quatre classes de caractéristiques de forme il faut ajouter deux autres classes qui ne s'expriment pas en particuliers ("blind_counter_sunk_hole", ...), agrégats répétitifs ("pattern") ou agrégats quelconques ("assembly").

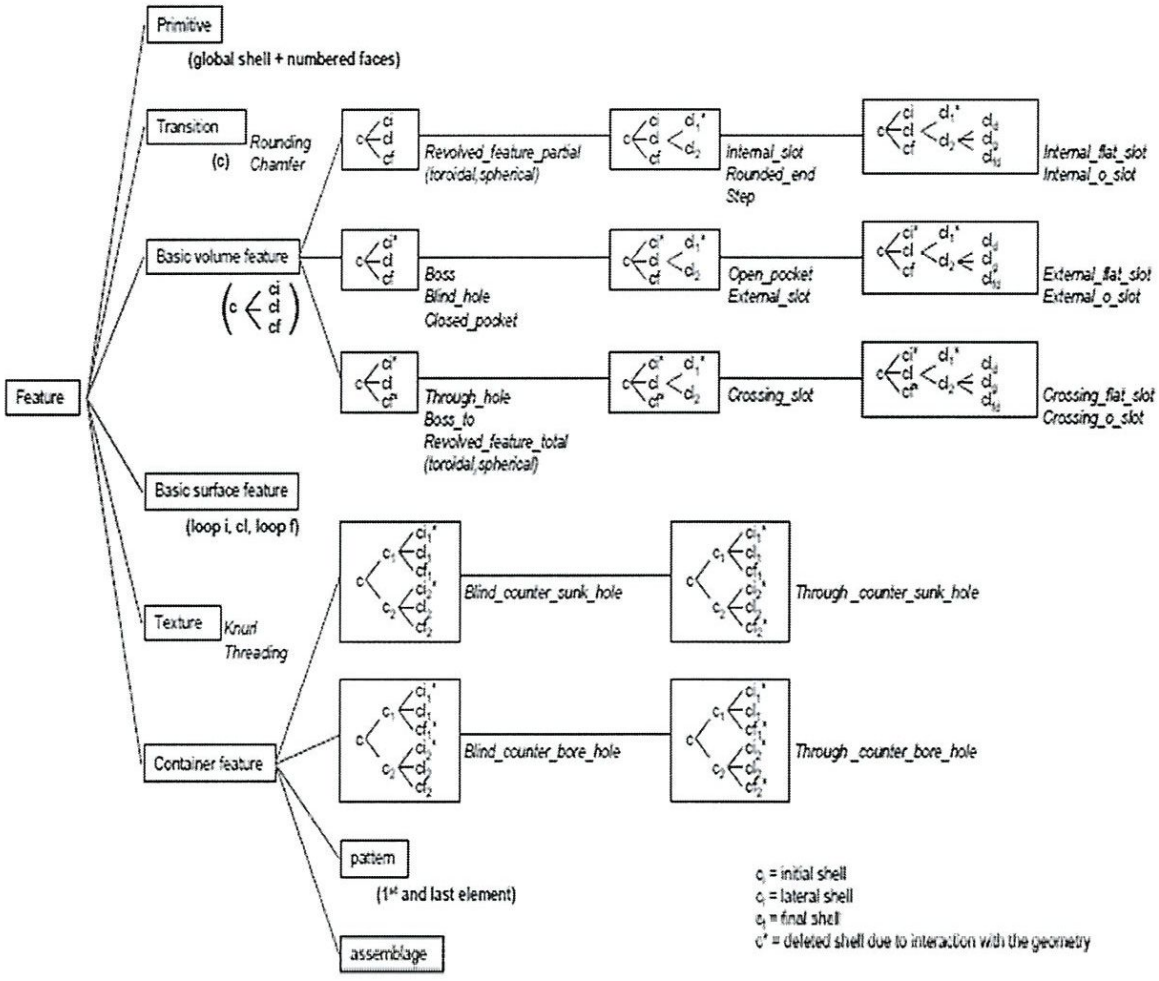


Figure 1.4 : *Classifications de caractéristiques de formes par rapport à leur structure [Agbod99]*

3.3.2. Le graphe des coques :

Les coques sont définies comme des agrégats de coques. Au niveau le plus bas de la hiérarchie, chaque coque représente une face. Les coques sont :

- **hiérarchiques** : elles sont structurées en coques, sous-coques, etc. afin de pouvoir appréhender la géométrie à différents niveaux de granularité et ainsi exprimer différentes sémantiques. Cette structure est définie pour chaque classe de caractéristique de forme

(une coque globale + les coques invariantes issues de notre classification de caractéristiques de forme)

- **connexes** : elles sont composées de coques jointives (dans la structure hiérarchique). Le graphe des coques a pour objectif de représenter l'évolution (scission et disparition, la fusion n'étant pas autorisée pour l'instant) de la structure invariante initiale, elle même connexe. Un nouveau noeud de ce graphe représente ensuite chacune des parties connexe qui apparaît lors d'une scission et constitue son identification. Il est ainsi possible de tracer l'évolution de chaque noeud et d'en définir la sémantique.

- **recouvrantes** : une sous-coque peut appartenir à différentes coques. La structure hiérarchique recouvrante permet la création d'agrégats quelconques au cours de la construction ce qui offre un mécanisme de désignation à la fois souple et extrêmement puissant. En particulier, comme nous le verrons dans la section 3.3.2.2.1, cela s'avère fondamental lorsque l'on souhaite référencer des agrégats issus de la fusion d'une caractéristique de forme avec certaines parties de l'objet.

3.3.2.1. Intérêts du graphe des coques

Il y a essentiellement deux intérêts à utiliser des coques et donc un graphe de coques : l'agrégation et la traçabilité :

- **L'agrégation** (intérêt des coques). Le but est de pouvoir appréhender la géométrie à différents niveaux de granularité, c'est-à-dire d'agréger les entités de bas niveaux. Les faces sont des entités de bas niveau, n'ayant souvent pour le concepteur que très peu de sémantique. Les coques permettent d'abstraire certains invariants de caractéristique de forme. Par exemple, lors d'une extrusion on a une unique coque latérale qui est composée de plusieurs faces latérales. Ces coques, qu'il est possible de structurer (en fonction de ces invariants), permettent donc de représenter dans chaque cas le niveau de sémantique pertinent pour le concepteur.
- **La traçabilité** (intérêt du graphe des coques). Le but est de pouvoir suivre l'évolution des coques, afin d'identifier lors de la construction les coques mises en jeu, puis lors des réévaluations les coques effectives (dans l'instance courante) correspondant à ces coques

référencées. Outre la possibilité de pouvoir référencer chaque coque ainsi représentée dans le graphe, ce dernier permet un autre niveau d'agrégation : l'agrégation de toutes les coques ayant eu un ancêtre commun. Ainsi, par exemple, la notion de coque latérale existera même en cas de scission de celle-ci, puisqu'à chaque niveau d'agrégation l'historique est conservé.

3.3.2.2. Structure du graphe :

Le graphe conserve à la fois la structure des caractéristiques de forme (agrégation), l'historique des coques (traçabilité) et certaines relations d'attache entre les coques (agrégation et traçabilité).

Chaque geste constructif peut être décomposé en 2 étapes. La 1ère étape est la spécification de la caractéristique de forme brute. Elle correspond dans le graphe à la structure invariante (défini dans notre classification de caractéristiques de forme). La structure invariante des caractéristiques de forme est représentée par des liens hiérarchiques entre coques. Cette structure hiérarchique initiale invariante de coques représente les entrées du graphe des coques. La 2ème étape est l'interaction avec l'objet qui produit des entités contingentes qui sont issues de l'évolution de la structure hiérarchique initiale. L'évolution de cette structure hiérarchique (conservée par des liens hiérarchiques) et en particulier l'évolution des coques est décrite par des liens historiques. La Figure 1.5 illustre la structure du graphe dans le cas d'un slot posé sur un cube.

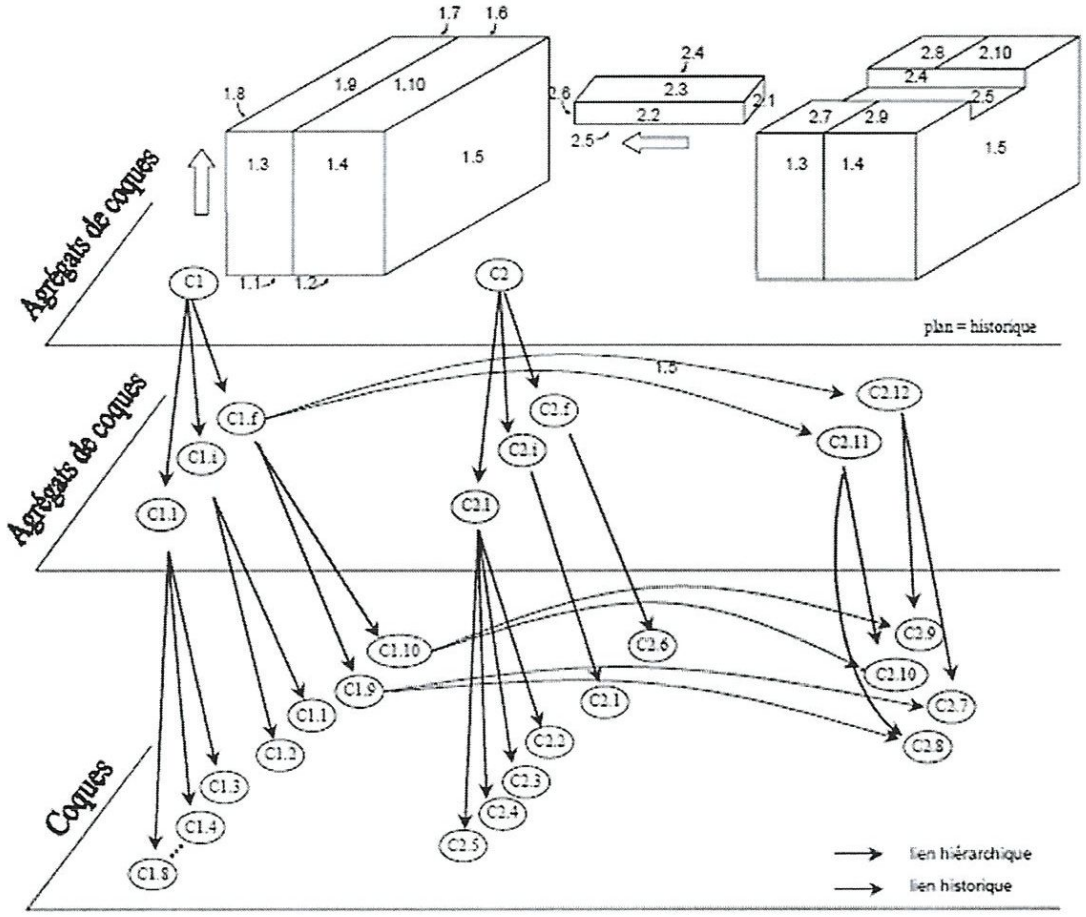


Figure 1.5 : *Exemple des graphes de coques (partiel) [Agbod99].*

Dans cet exemple, le cube est une 3D variété construit par extrusion d'une 2D variété composée des deux coques C1.1 et C1.2. Chaque coque du graphe est identifiée par deux éléments. Le premier est le numéro d'ordre du geste constructif, le deuxième est défini à la scction 3.3.4.

3.3.2.2.1. Liens hiérarchiques

La structure d'une caractéristique de forme (décrite dans la section 3.3.1) est représentée dans le graphe par les liens hiérarchiques. Ces liens qui sont définis entre agrégats d'entités, permettent de structurer les données et de dégager des agrégats invariants. Outre la représentation de la structure invariante des caractéristiques de forme, ces liens hiérarchiques permettent de représenter de nouvelles agrégations apparaissant au cours de la modélisation. La Figure 6 en donne un exemple. Une protrusion est posée sur la face supérieure d'un cube construit par extrusion. Une coque C3 peut être générée (par exemple sur demande explicite du concepteur qui souhaite manipuler cet agrégat) afin de pouvoir nommer et donc adresser l'agrégat correspondant à l'union de la protrusion (C2) et de la coque portant cette protrusion (C1f). C3 possède donc deux liens hiérarchiques vers C1f et C2. Nous pouvons de plus remarquer que cet exemple illustre le cas d'un recouvrement de coques présenté dans la section 3.3.2.1 puisque la coque C1f appartient maintenant à deux coques distinctes C1 et C3.

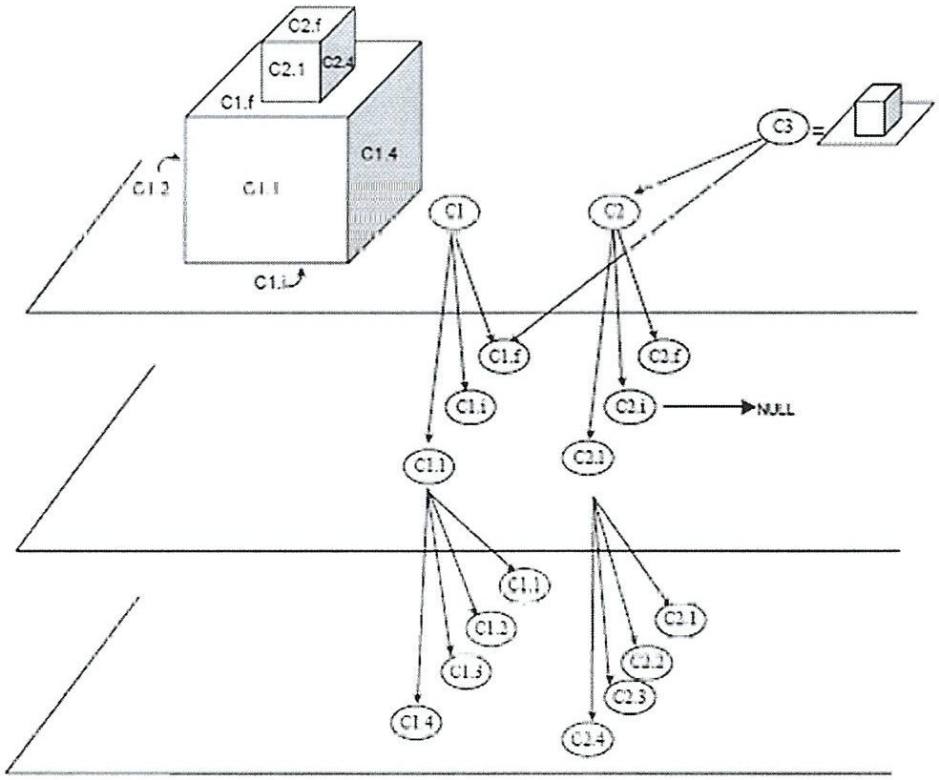
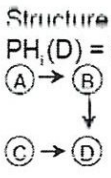


Figure 1.6 : *Exemples des liens hiérarchiques [Agbod99]*

La structure hiérarchique évolue avec les modifications des coques. Ainsi dans l'exemple de graphe, illustré dans la Figure 1.5, la coque du dessus (c1f) qui est composée des coques c1.9 et c1.10 est scindée en deux coques c2.11 et c2.12. Ces dernières sont à leur tour composées du résultat de la scission des coques c1.9 et c1.10, de sorte que l'on retrouve à chaque étape une structure hiérarchique (modifiée). Une propriété de ce graphe de coques, qui en simplifie considérablement la gestion, est que les liens hiérarchiques peuvent n'être conservés et décrits qu'au niveau des feuilles historiques du graphe. Pour prouver cette propriété nous avons étudié pour deux étapes successives et pour deux niveaux hiérarchiques successifs (ici coques de plus bas niveau –faces- et premier niveau d'agrégation, mais cela s'étend à 2 niveaux hiérarchiques successifs quelconques) toutes les combinaisons possibles d'historiques d'entités. L'historique se résume à la scission. En effet la fusion n'est actuellement pas autorisée et la disparition des entités (en fait, pointeurs vers NULL) ne change pas les liens historiques du graphe. Il est possible de déduire de ces différents cas de figure que la connaissance de la structure hiérarchique de l'étape i suffit pour connaître celle de l'étape i -1. En effet, il suffit de comparer l'ordre des numéros d'étape (ordre de création) de deux coques liées hiérarchiquement. Par exemple, dans le 2ème graphe de la Figure 1.7, la coque f2.1 est une sous-coque de c1.1. La coque f2.1 qui a été créée après c1.1 est donc issue d'une coque (f1.1) qui était sous-coque de c1.1.

Ce raisonnement, appliqué à chacun des 4 cas, et étendu à des étapes i -1 et i quelconques, permet de généraliser la propriété. Par exemple, dans le graphe représenté ci-dessous, si la coque B est le Père Hiérarchique de la coque D (PHi (D) = B) à l'étape i, il est possible de déduire de l'ordre de création de D et C, quel était le Père Hiérarchique de D (ou C selon le cas) à l'étape I -1.



Structure à l'étape i			Déduction de la structure à l'étape i -1		
Si	n° d'étape (D) < n° d'étape (B)	alors	$PH_{i-1}(D) = A.$		
Si	n° d'étape (D) = n° d'étape (B)	alors	$PH_{i-1}(C) = A$		
Si	n° d'étape (D) > n° d'étape (B)	alors	$PH_{i-1}(C) = B.$		

Il est possible de retrouver la structure hiérarchique de l'étape i -1, connaissant celle de l'étape i. Par récurrence on peut en conclure qu'il est possible de conserver les liens hiérarchiques uniquement au niveau des feuilles historiques. Notons par contre, que cette propriété devrait être étendue si la fusion devait être introduite dans le graphe. Cette connaissance de l'état de l'ancien graphe à une étape quelconque est indispensable pour faire le matching avec le graphe issu de la réévaluation.

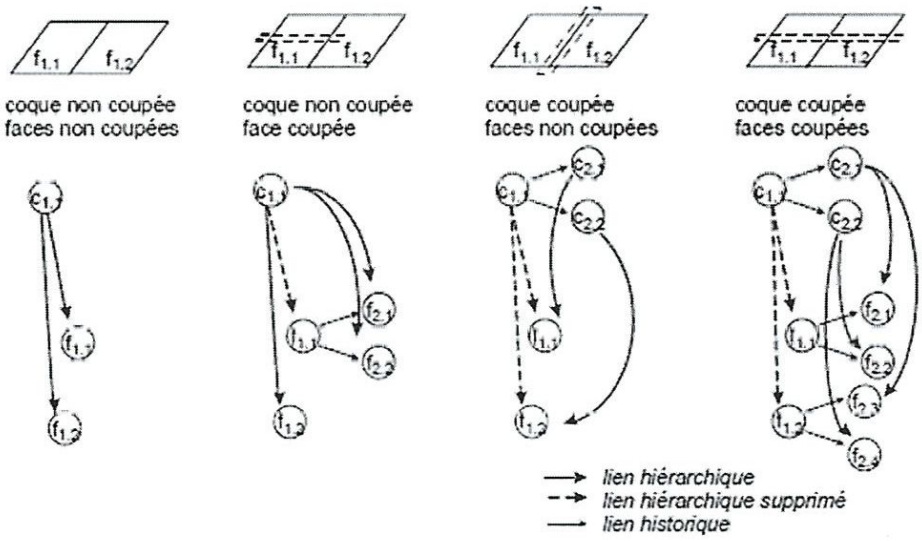


Figure 1.7 : *Evolution des liens hiérarchiques [Agbod99].*

3.3.2.2.2. Liens historiques :

Pour chaque noeud du graphe et en particulier pour chaque feuille de la structure hiérarchique, on va tracer son évolution. Il y a 3 possibilités pour cette modification (la fusion n'est pas autorisée) :

- **La disparition** : Lors de l'interaction entre la géométrie de la caractéristique de forme et la géométrie de l'objet dans lequel elle est plongée, un certain nombre de coques (de la caractéristique de forme et/ou de l'objet) disparaissent. Ces coques restent présentes dans le graphe, mais sans lien vers la géométrie (le lien historique abouti à NULL).
- **La modification** : Par modification, Agbodan a entendu toutes évolutions qui conservent la connexité des coques. Nous estimons que toute coque modifiée est égale à la coque avant modification. En effet, si à une étape i une coque est parfaitement identifiée, et si à l'étape i+1 cette coque est modifiée (sans scission), elle reste parfaitement identifiée dans le graphe. Les modifications de coques ne sont donc pas représentées dans le graphe.
- **La scission** : Il y a scission lorsqu'une coque est coupée en plusieurs coques non connexes. Cette scission est représentée dans le graphe par les liens historiques. Ainsi, une coque coupée va pointer sur les sous-coques résultat et inversement. Les liens historiques permettent de tracer l'évolution (la scission) des entités topologiques. L'historique est conservé à tous les niveaux de granularité (faces, coques, agrégats de coques, ...), ce qui donne au modèle une puissance d'expression supérieure à celui de Kripac. En effet, lorsqu'une coque est coupée en plusieurs morceaux lors d'une opération, il est possible d'avoir accès non seulement à la coque (qui est représentée dans la hiérarchie) mais aussi à chacun des morceaux de coques (qui sont représentés dans l'historique). C'est par rapport à ces liens historiques que se fera le parcours du graphe lors du matching.

3.3.3. Structure d'un noeud du graphe :

Chaque noeud représente une coque qui existe ou a existé dans le modèle. Toutes les coques sans liens historiques sortant sont présentes dans la géométrie.

3.3.4. Nomination des entités :

L'identification des différentes entités (sommets, arêtes, coques) se fait par rapport aux faces (coque de plus bas niveau, c'est-à-dire les feuilles de la structure hiérarchique) invariantes. Il faut donc pouvoir nommer ces faces de manière unique et totalement déterministe. La nomination des coques aussi bien invariantes que contingentes se fait par rapport au numéro d'étape (ordre de création) et par rapport à un identifiant qui les caractérise de manière unique. Pour chaque geste constructif, nous considérons qu'il y a 2 étapes. Premièrement, la spécification de la caractéristique de forme ; c'est-à-dire que toutes les entités (invariantes) de la

caractéristique de forme et en particulier les coques de plus bas niveau doivent être nommées. Deuxièmement, le plongement de la caractéristique de forme dans l'objet existant. L'interaction avec la géométrie existante entraîne la disparition et la modification d'entités et l'apparition de nouvelles entités. Ces entités contingentes doivent être nommées. Il y a donc deux types de numérotation à mettre en oeuvre : une pour les coques invariantes et une pour les coques contingentes.

3.3.4.1 Coques invariantes :

Pour les coques invariantes, deux cas sont à distinguer. Il y a d'une part les coques de plus bas niveau (feuilles de la structure hiérarchique) et d'autre part les coques de niveau supérieur.

3.3.4.1.1 Coques invariantes de plus bas niveau :

Pour identifier les coques invariantes de plus bas niveau (faces invariantes), nous allons nous baser sur la topologie. Cette nomination est robuste par rapport aux modifications géométriques. En effet, les modifications géométriques n'entraînent pas de modification au niveau de la topologie. Comme les modifications géométriques (dimension, position, ...) sont les seules qui s'appliquent aux caractéristiques de forme et donc aux coques invariantes, cette nomination par rapport à la topologie est robuste par rapport aux réévaluations. L'idée est de partir d'une demi arête de départ, l'hypothèse étant que les caractéristiques de forme sont définies, en plus de leurs paramètres, par une demi arête (définissant le point de départ de la numérotation) et de faire un parcours radial autour des arêtes, puis un parcours suivant les arêtes des contours de la face. Le parcours qui est unique (par rapport à la demi arête de départ) et qui couvre tout l'objet permet d'assigner un numéro à chaque face. Le nom des coques invariantes de plus bas niveau est donc composé du numéro d'étape et du numéro de face issue du parcours. Ce parcours se fait de la manière suivante.

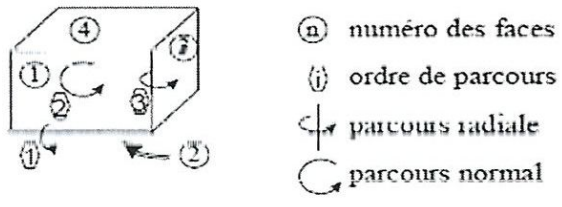


Figure 1.8 : *Parcours topologique lors de la nomination [Agbod99].*

- On parcourt de manière orientée le circuit de demi arête entourant la face en cours
- Pour chaque demi arête rencontrée on parcourt de manière radiale toutes les faces adjacentes.
- Si ces faces ne le sont pas déjà numérotées, on leur assigne un numéro et on les met dans la file.
- Lorsque le parcours du circuit est terminé on prend la première face de la file et on recommence le parcours précédant en partant de la demi arête "commune" à cette face et la face de plus petit numéro.

Ces différents points peuvent être traduit dans l'algorithme suivant.

```
Face_courante = face adjacente à la demi arête              {il n'y en a qu'une}
Répéter
   Répéter
      Répéter
         Face_tmp = face suivante dans le parcours radial  {autour de demi arête}
         Si (Face_tmp pas numéroté) alors
         {
               numéroter Face_tmp
               file <- Face_tmp
         }
      Jusqu'à (Face_tmp=Face_courante)
      Demi_arete = demi arête suivante dans parcours normal {autour de Face_courante}
   Jusqu'à fin du parcours du circuit d'arête
   Face_courante = Oter 1ère face de la file
   Demi_arête = demi arête commune à Face_courante et à la face déjà numérotée de plus petit
numéro
Jusqu'à file vide
```

Notons que cet algorithme de parcours, illustré dans le cas de polyèdres sans frontières internes, est utilisable pour la nomination de n'importe quelle 3D variété y compris celles possédant des frontières internes.

3.3.4.1.2 Coques invariantes de niveau supérieur :

Les coques invariantes, autres que celles de plus bas niveau, sont des agrégations de coques invariantes. Les coques invariantes peuvent être nommées par rapport à la liste des coques qui les composent, et par transitivité, par rapport à la liste des coques de plus bas niveau qui les composent. Ces dernières étant parfaitement identifiées et la liste étant unique, les coques sont donc identifiées de manière unique par la liste de coques de plus bas niveau. Le nom est donc composé du numéro d'étape et de la liste des noms des faces (coque de plus bas niveau) qui composent la coque.

3.3.4.2 Coques contingentes :

Le nom des coques contingentes est constitué du numéro d'étape et d'un numéro itératif (quelconque, mais unique par étape de construction). Ce nom, insuffisant pour permettre le "matching" ultérieur, est associé à des informations sur le contour de la coque permettant de distinguer deux sous-coques issues par subdivision de la même coque. Le contour est défini par les coques de plus bas niveau (feuilles de la structure hiérarchique -face-) qui entourent la coque scindée. Ces informations seront utilisées en réévaluation, lors du "matching" ultérieur des coques contingentes. Bien que l'algorithme complet de "matching" dépasse le cadre de cet article son principe consiste à conserver le graphe de coques initial sur lequel s'appuient toutes réévaluations. En effet, lors d'une réévaluation, un deuxième graphe de coques est construit parallèlement au graphe initial. A chaque étape de cette construction, les nouvelles entrées apparaissant dans le deuxième graphe sont comparées grâce à un parcours en retour arrière à certaines coques présentes dans le graphe initial. Lorsqu'il y a identification, les nouvelles entrées sont dotées du même numéro itératif. Dans le cas contraire, elles reçoivent des numéros itératifs inexistants dans le graphe initial. Finalement, une référence dans la spécification paramétrique à une coque inexistante dans le graphe courant est remplacée par une référence à une ou plusieurs coques calculées par "matching" du graphe initial et du graphe courant par un algorithme similaire à celui proposé par Kripac.

4. Conclusion :

Agbodan a proposé un nouveau mécanisme de nomination persistante associé à une structure hiérarchique permettant de tracer l'évolution historique d'invariants facilement identifiables dans chaque geste constructif. Afin de dégager de tels invariants, une nouvelle classification basée sur la structure intrinsèque et sémantique de chaque caractéristique de forme à été introduite. Cette approche offre trois principaux avantages. Premièrement cela permet d'identifier de manière non ambiguë les entités topologiques d'un modèle paramétrique initial. Deuxièmement cela permet de faire du "matching" afin de retrouver ces mêmes entités dans un modèle réévalué, malgré les multiples variations topologiques. Finalement, un tel mécanisme autorise la capture et la désignation **persistante** de différentes **sémantiques** exprimables par le concepteur [Agbod02].

Dans le chapitre suivant on va traiter précisément le premier aspect du problème qui est l'identification non ambiguë des objets et le troisième aspect du problème qui est la prise en charge de la sémantique, autrement dit, tenir en compte les intentions du concepteur.

CHAPITRE 2

Modèle et Expérimentation

I. Modèle :

1. Introduction :

On a vu dans le chapitre précédent le problème de nomination persistante dans les modèles paramétriques, et les approches et solutions proposées à ce problème, comme l'approche de Kripac, l'approche de Chen et l'approche d'Agbodan.

Nous avons vu que Kripac a abordé principalement le problème du matching alors que Chen a traité essentiellement le problème de la définition d'un mécanisme de nomination non ambiguë pour la phase de construction.

L'approche d'Agbodan qui est relativement similaire à celle de Kripac , son mécanisme autorise la capture et la désignation persistante de différentes sémantiques exprimables par le concepteur.

De notre part, on essaye dans ce chapitre de proposer un modèle simplifié qui prend en charge les intentions du concepteur lors de la conception. Autrement dit, notre modèle va traiter et résoudre le problème de la sémantique lors de la reconstruction et la réévaluation d'un objet qui a subi une modification à une certaine étape de conception. Ce modèle sera caractérisé par sa simplicité, sa clarté et son efficacité pour le traitement du problème de la nomination persistante en tenant compte de la sémantique.

2. Le modèle :

L'étude de la sémantique est quelque chose de délicat, car elle fait intervenir les intentions de concepteur, et puisque ses intentions peuvent théoriquement aller jusqu'à l'infini, on doit mettre en quelque sorte des conditions ou des intervalles pour limiter ces intentions.

Donc on doit proposer une liste des conditions, des contraintes ou un simple intervalle pour limiter les cas possibles issus de la reconstruction ou de la réévaluation automatique d'un modèle en tenant compte de la sémantique.

2.1. Les heuristiques :

On doit d'abord considérer un objet initial qui se compose d'un ou plusieurs sous objets.

Une fois que la modification faite, lors de la réévaluation on peut avoir plusieurs résultats possibles. Ces résultats sont classés dans un ordre qui essaie de préserver au plus la forme topologique globale de l'objet initial.

Donc notre modèle on propose quelques heuristiques où chacune d'elles permet de satisfaire une forme topologique globale de l'objet après la modification:

2.1.1. Première heuristique :

La première heuristique concerne le déplacement du sous objet qui a subit une modification par rapport aux sous objets non modifié. Cela donne un nouvel objet ayant une certaine forme topologique qui peut être celle souhaitée par le concepteur.

On prend comme exemple un objet initial composé d'un cylindre superposé à un cuboïde (voir la figure 2.1).

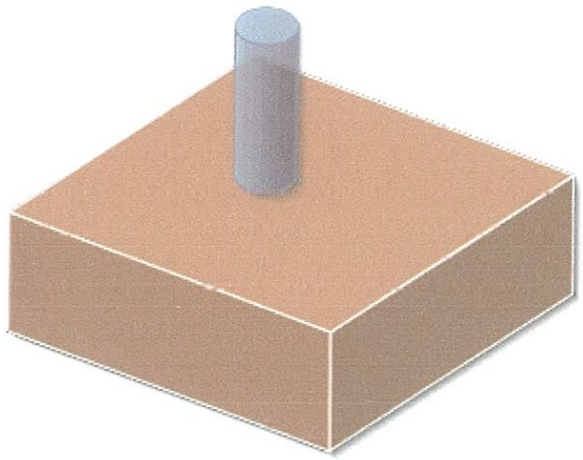


Figure 2.1 : *Objet initial.*

La réévaluation ne se fait qu'après que l'objet initial subit une modification à une certaine étape de sa conception.

Dans notre exemple, cette modification se fait sur le cuboïde, précisément sur sa hauteur qui est réduite en un demi de sa valeur initiale comme montré dans la figure 2.2.
La modification de la hauteur du cuboïde génère un vide entre lui et le cylindre, ce vide sera considéré lors de la réévaluation de l'objet initial (cas ambigüe).

La modification peut être autrement appliquée non pas sur la hauteur du cuboïde mais sur sa largeur, et cette modification n'entraine pas un vide entre le cuboïde et le cylindre, puisque les deux restent superposés comme montré dans la figure 1, mais il y a une modification de la position du cylindre par rapport au centre du cuboïde, et donc le déplacement du cuboïde sera horizontalement pour que le cylindre soit placé à son centre.

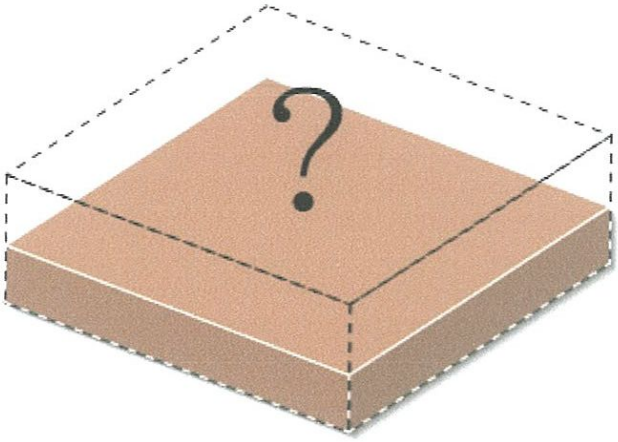


Figure 2.2 **:** *Réévaluation avant l'apparition du cylindre.*

En considérons la première heuristique mentionnée ci-dessous, le cuboïde modifié se déplace vers le cylindre et forment les deux ensemble un seul objet (voir figure 2.3).

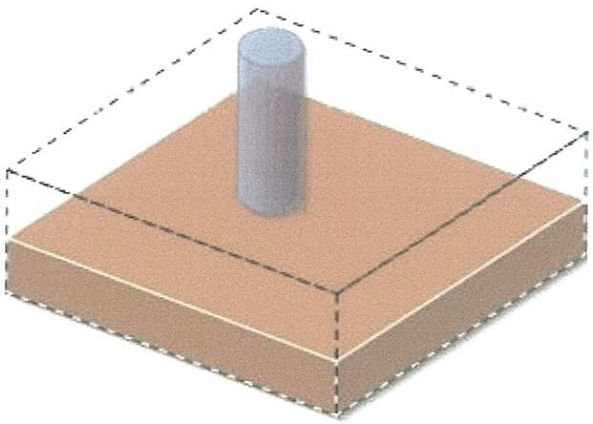


Figure 2.3 : *Déplacement du cuboïde vers le cylindre.*

2.1.2. Deuxième heuristique

La deuxième heuristique est le déplacement du sous objet non modifié vers le sous objet modifié, et les deux formeront un seul objet différent du premier objet initial en dimension et en topologie.

On considère le même exemple du cylindre et cuboïde montré dans la figure 2.1.
D'abord le cuboïde est toujours le sous objet qui subit la modification sur sa hauteur qui sera réduite (voir figure 2.2).

On considérant cette fois-ci la deuxième heuristique, le cylindre se déplace vers le cuboïde modifié et formeront les deux ensembles un seul objet (voir figure 2.4).

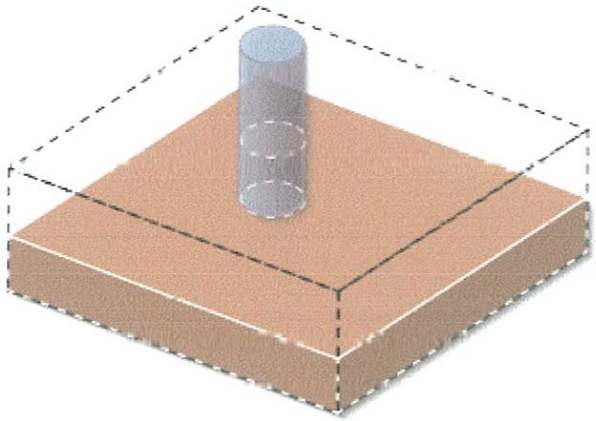


Figure 2.4 : *Déplacement du cylindre vers le cuboïde.*

2.1.3. Troisième heuristique :

La troisième heuristique n'est pas un déplacement mais un autre type de modification qui est la modification topologique, où le sous objet non modifié se transforme (prolongement, élargissement, etc.) vers le sous objet modifié, et les deux formeront un seul objet différent du premier objet initial en dimension et en topologie.

On considère le même exemple du cylindre et cuboïde montré dans la figure 2.1.
Le cuboïde est toujours le sous objet qui subit la modification et sa hauteur sera réduite (voir figure 2.2).

On considérant cette heuristique, il n'y a pas de déplacement ni du cylindre, ni du cuboïde, mais une transformation de cylindre qui se prolonge vers le cuboïde modifié (voir figure 2.5).

Ni le cylindre, ni le cuboïde ne se déplacent dans ce cas d'heuristique.

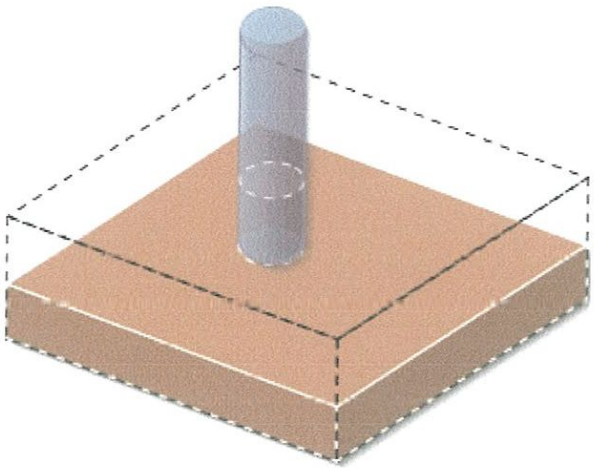


Figure 2.5 : *Transformation et prolongement du cylindre vers le cuboïde.*

II. Expérimentation :

1. Introduction :

Tout d'abord, il convient de définir ce que l'on entend par la réévaluation du ... paramétrique. Bien entendu, cette technique à pour objectif la reconstruction de l'objet modélisé après la modification d'un certain nombre du paramètre du modèle initial.

Dans partie nous concentrons nos efforts pour l'implémentation d'une application qui permet de réaliser un exemple de réévaluation d'un modèle paramétrique (2D) sans et avec la sémantique. Grace à cette application on peut modifier l'objet à l'étape de conception de notre choix la réévaluation du modèle ce fait soit en ne tenant pas compte de la sémantique soit le contraire en se basant sur les différents heuristiques proposée.

2. Conception générale du système:

Le processus de notre application consiste à créer deux rectangles superposés en deux étapes pour qu'on puisse obtenir un modèle paramétrique final.

Le but du système est de réévaluer ce modèle selon des spécifications paramétriques et les différentes heuristiques.

2.1. Les étapes de la réévaluation :

La figure suivante (Figure 3.1) illustre les différentes étapes de la réévaluation:

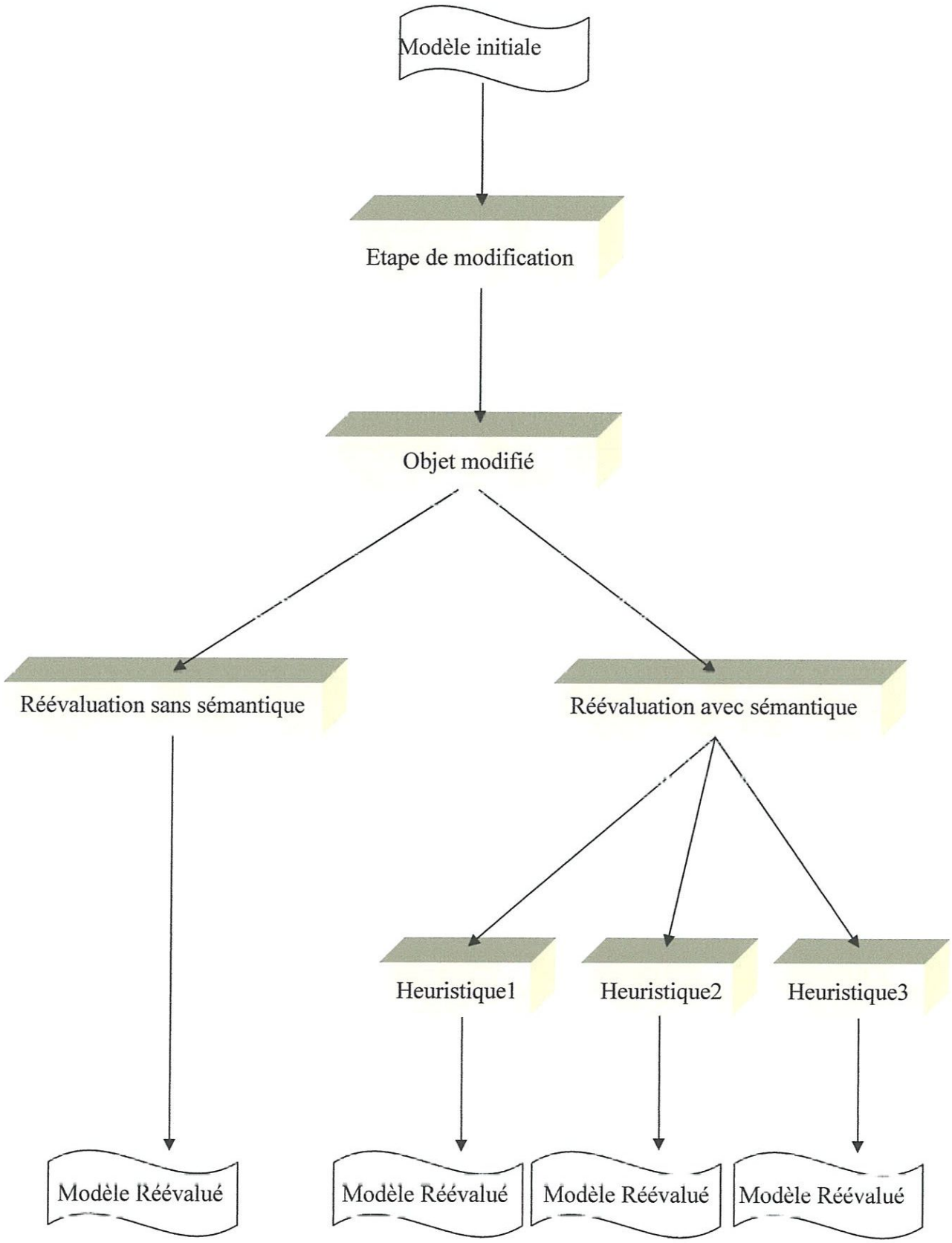


Figure 2.6 : *Etapes de la réévaluation.*

2.1.1. L'étape de modification :

L'étape de modification ce fait selon des coordonnées de séparation.

Coordonnées de séparation

Les coordonnées de séparation permettent au concepteur de modifier l'une des pièces de l'objet (modification en hauteur), ce qui fait qu'il y aura une des deux objets.

Pour cela nous avons laissé au concepteur de choisir l'étape de modification puis il entre les valeurs de séparation pour obtenir **l'objet modifier**.

2.1.2. Réévaluation sans sémantique :

Cette étape permet de reconstruire l'objet après la modification apportée.

Dans cette phase la réévaluation se fait sans tenir en compte les intentions du concepteur comme conséquence on obtient le modèle réévalué ayant une forme topologique ambigu.

2.1.3. Réévaluation avec sémantique :

Contrairement à la réévaluation précédente, celle-ci se fait en tenant en compte des intentions du concepteur, et puisque ses intentions ne sont pas quelque chose de concret, elles peuvent varier, bien sûr en se limitant à l'intervalle des heuristiques considérées dans notre modèle.

On a proposé trois heuristiques :

1. L'objet modifié se repositionne pour être superposé à l'objet non modifié, ce qui nous donne un modèle ayant une forme topologique proche de celle de l'objet initial.
2. L'objet non modifié se repositionne pour être superposé à l'objet modifié, ce qui nous donne un modèle ayant une forme topologique proche de celle de l'objet initial.
3. L'objet non modifié se transforme (se rallonge) pour que l'objet modifié lui soit superposé, ce qui nous donne un modèle ayant une forme topologique assez proche de celle de l'objet initial.

L'interface graphique de l'application est la suivante (Figure 3.2):

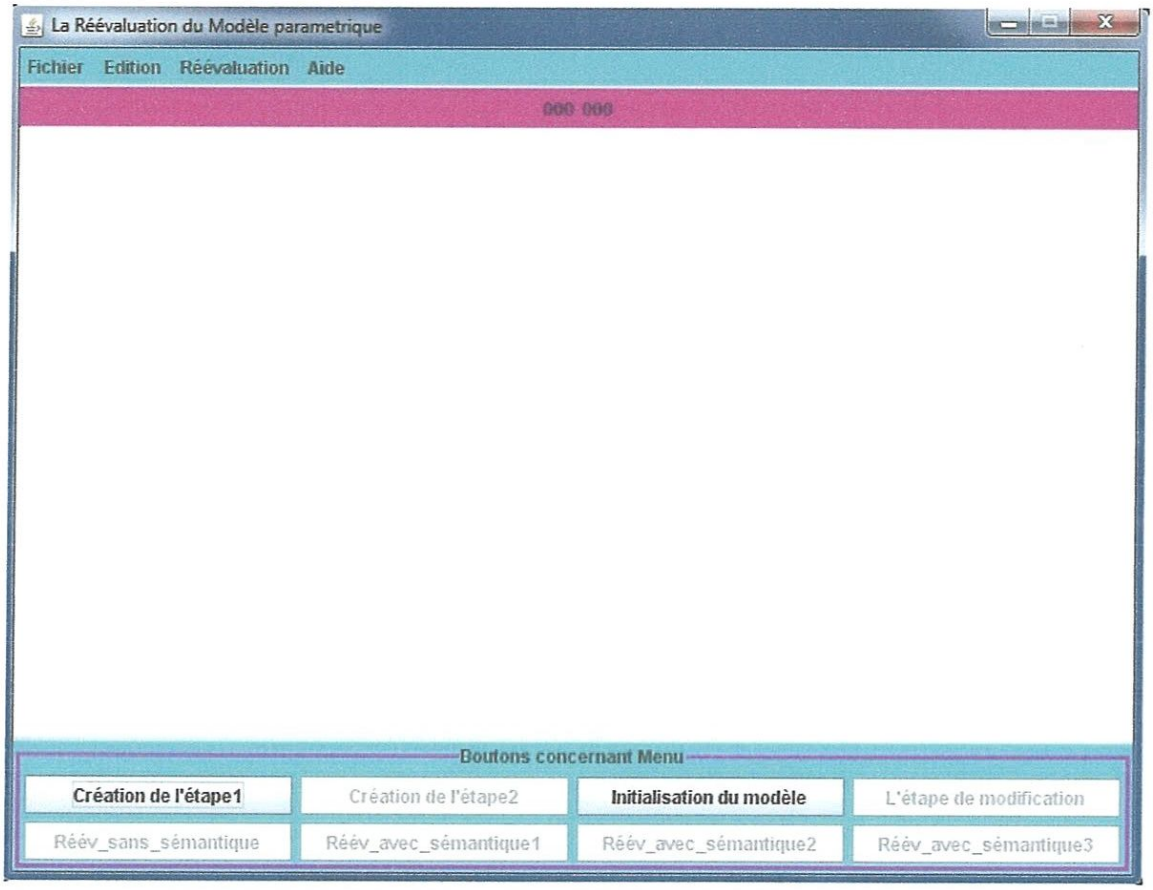


Figure 2.7 : *L'interface graphique de l'application.*

Dans la suite de notre démonstration nous allons utiliser le l'objet de la (Figure 3.3) comme exemple de test.

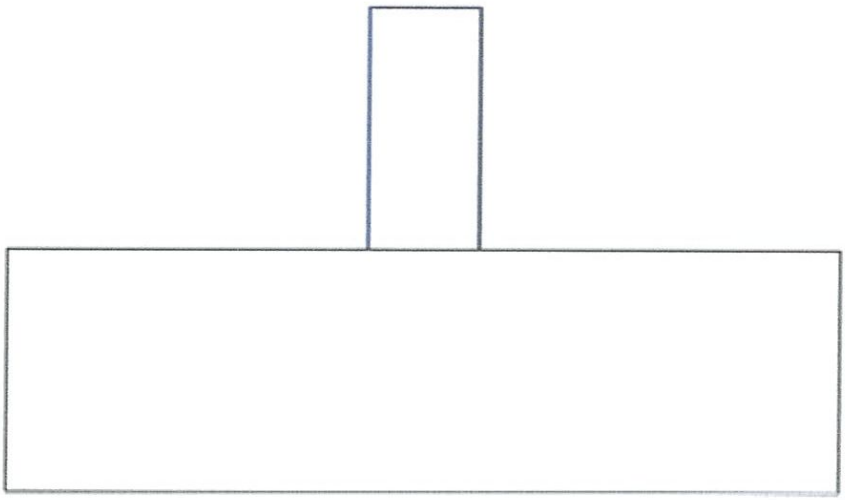


Figure 2.8 : *Le modèle paramétrique.*

3. Réévaluation de notre modèle paramétrique :

Avant d'aborder l'étape de réévaluation, nous avons créé un objet constitué de deux sous objets (donc en deux étapes).

L'étape1 :

La première étape consiste à créé un rectangle (Figure 3.4) avec les coordonnées suivantes : (X, Y)=(330,80).

-la taille de l'objet1 est : (largeur, hauteur)=(40,120).

-remarque **:** les paramètres d'objets sont fixes pour avoir un modèle final superposé.

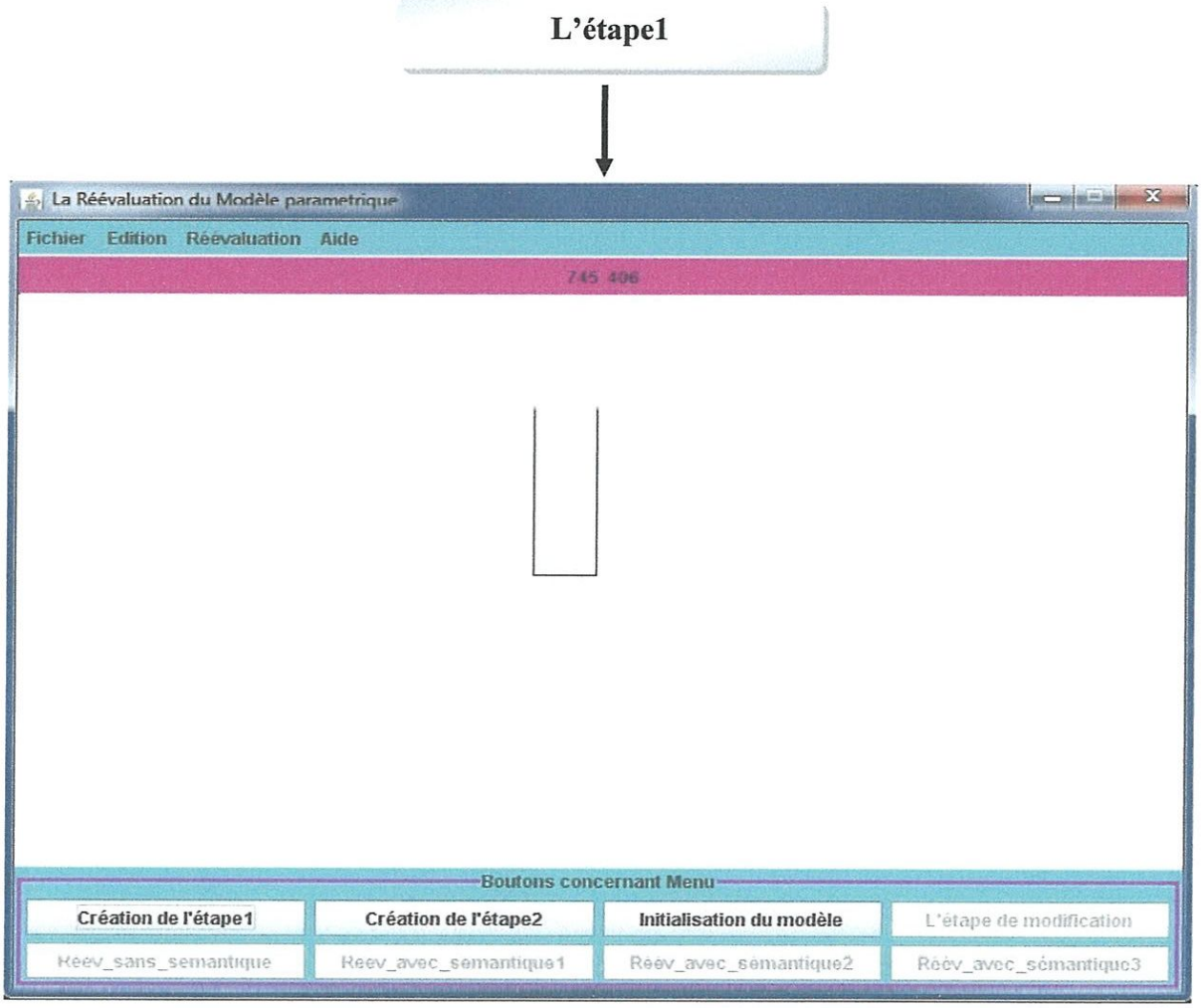


Figure 2.9 : *Création de l'objet1.*

L'étape2 :

La deuxième étape consiste à créé un second rectangle avec les coordonnées suivantes :

(X, Y)=(200,200).

-la taille de l'objet2 est : (largeur, hauteur)=(300,100).

- le premier rectangle qu'on a créé sera superposé au deuxième (Figure 3.5).

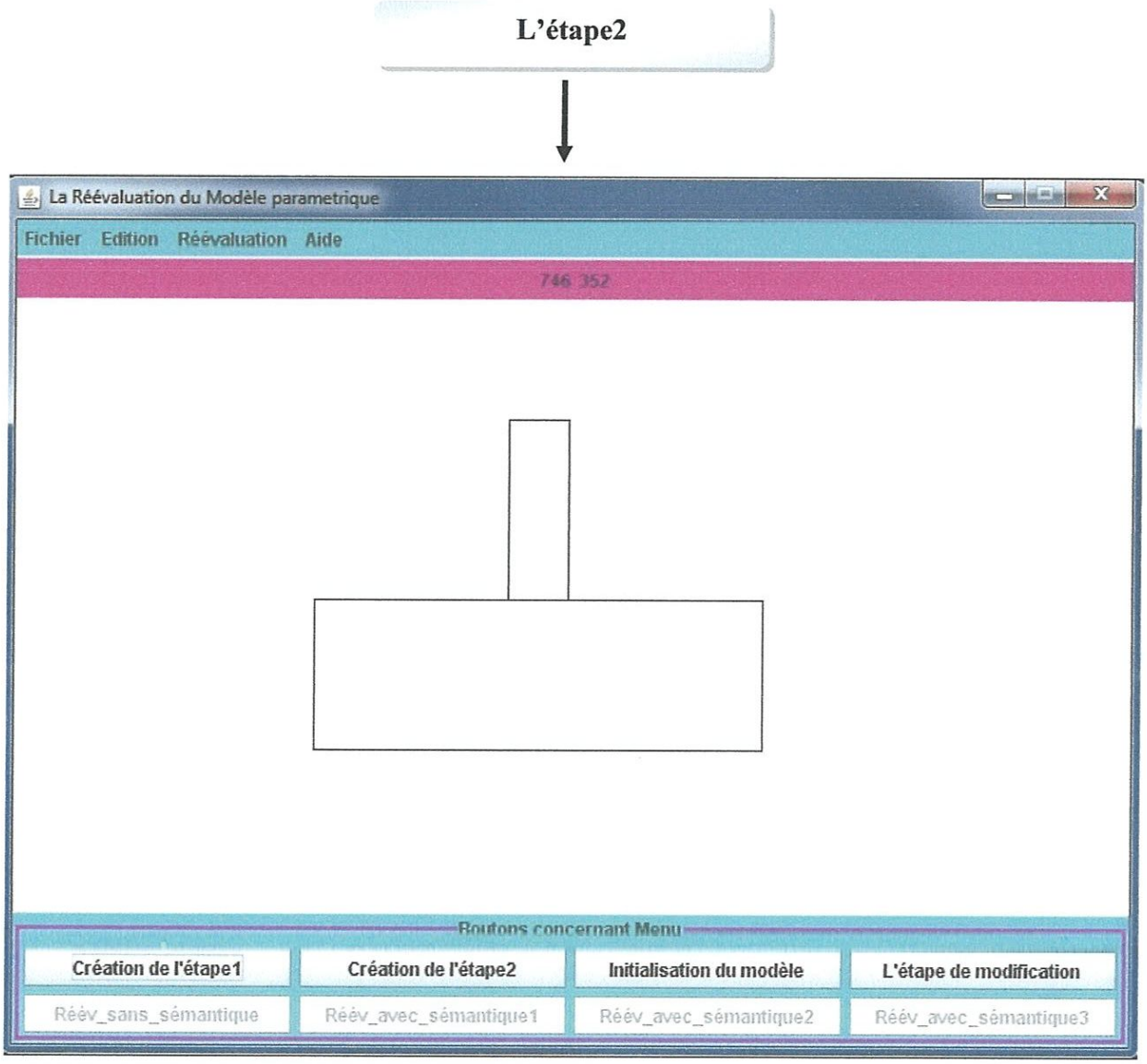


Figure 2.10 : *Création de l'objet2.*

3.1. L'étape de modification :

-Au niveau de l'étape de modification, l'utilisateur ou le concepteur peut choisir l'étape qu'il veut pour réaliser la modification de l'objet initial.

-la modification en largeur ne montre pas une ambiguïté sur le modèle.

-Par contre, si l'utilisateur fait une modification sur l'hauteur de l'objet, alors le processus de modification coupe l'objet en bas (Figure 3.6).

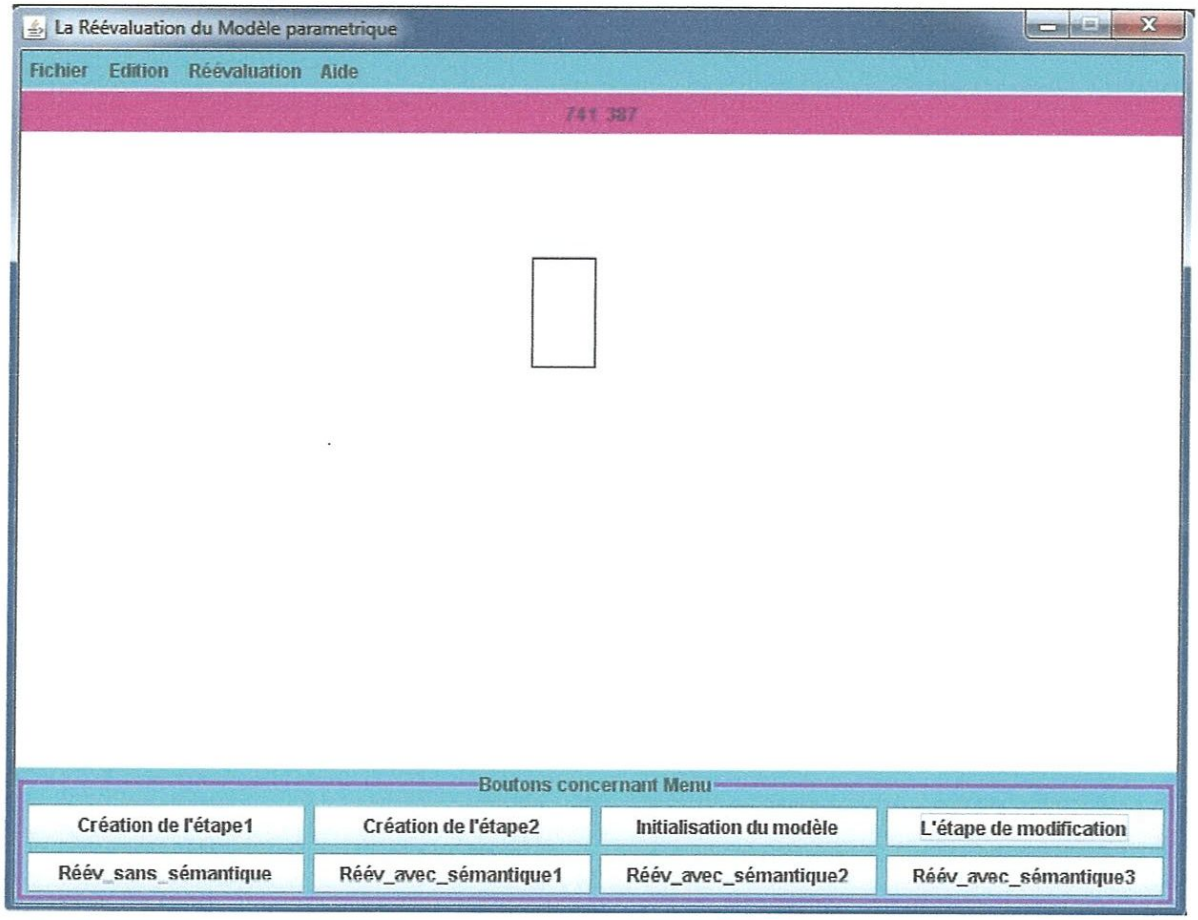


Figure 2.11 : L'étape de modification.

3.2. Réalisation de la réévaluation sans sémantique :

-La réévaluation sans sémantique fait la reconstruction directe de notre modèle après la modification du concepteur (c'est-à-dire, sans tenir compte des intentions de l'utilisateur).

-On remarque que le modèle réévalué présente une certaine ambigüité, c'est-à-dire, que les deux objets sont séparé comme s'ils flottaient parfaitement dans l'espace (Figure 3.7).

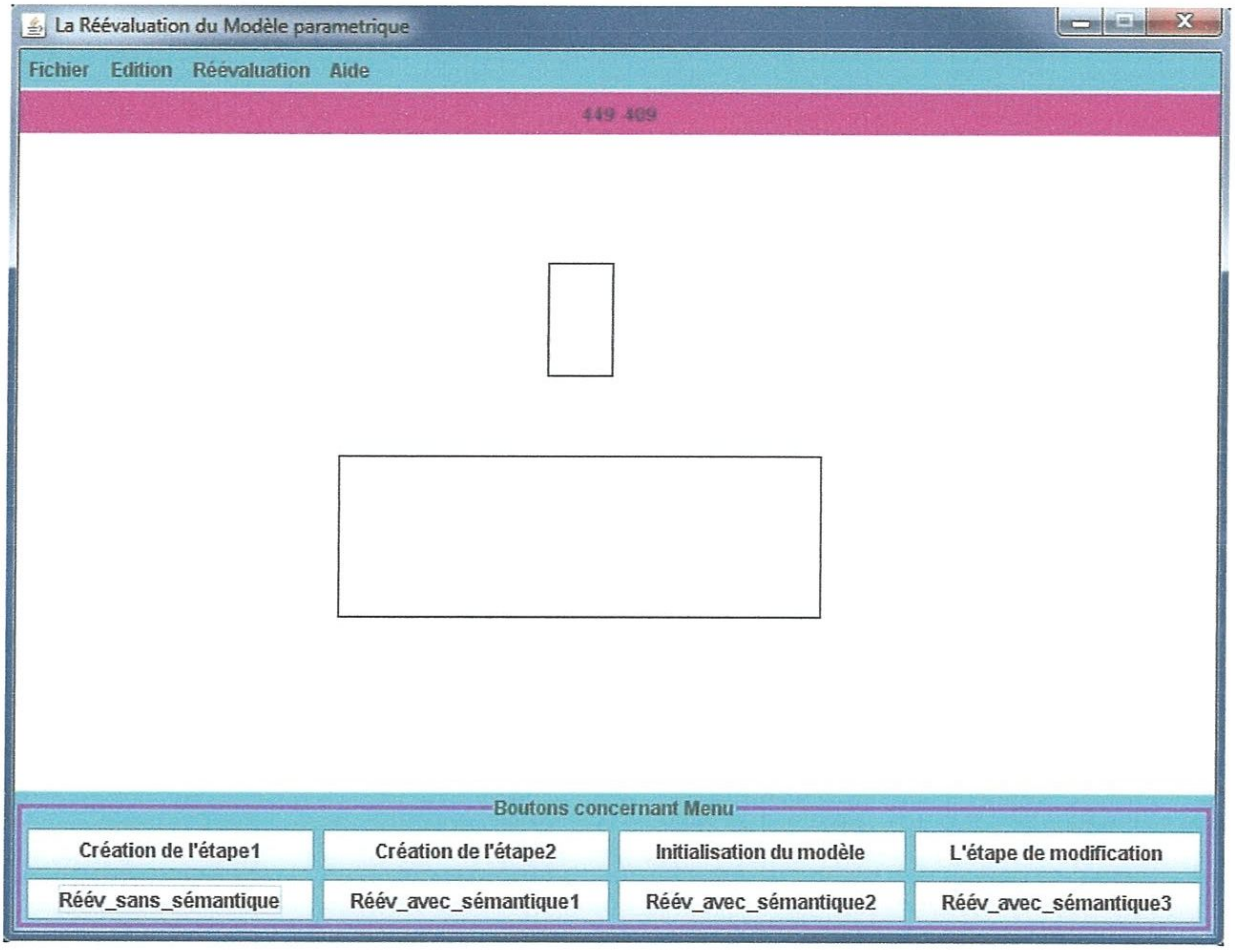


Figure 2.12 : *La réévaluation sans sémantique.*

3.3. Réalisation de la réévaluation avec sémantique :

3.3. 1. Heuristique1 :

-Ici l'objet1 se déplace vers le bas pour éliminer l'ambiguïté (Figure 3.8).

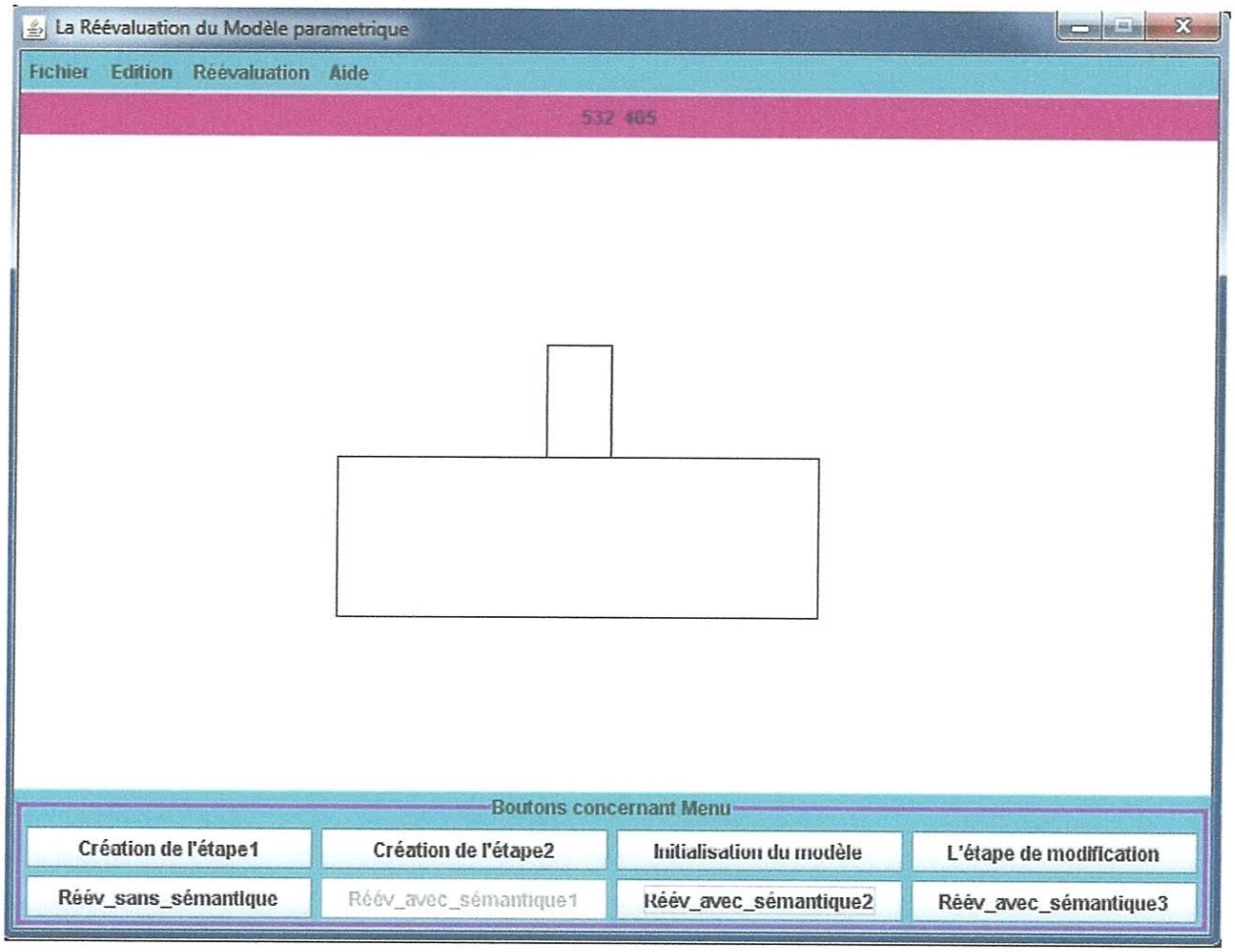


Figure 2.13 : *La réévaluation avec l'heuristique1.*

3.3. 2. Heuristique2 :

-Dans l'heuristique2, l'objet2 (non modifié) se déplace vers le haut pour éliminer l'ambiguïté de notre objet réévalué (Figure 3.9).

-Dans les deux heuristiques (1 et 2), on remarque que les deux objets deviennent superposés, ce qui nous permet de satisfaire à un certain degré les propriétés topologiques de l'objet initial.

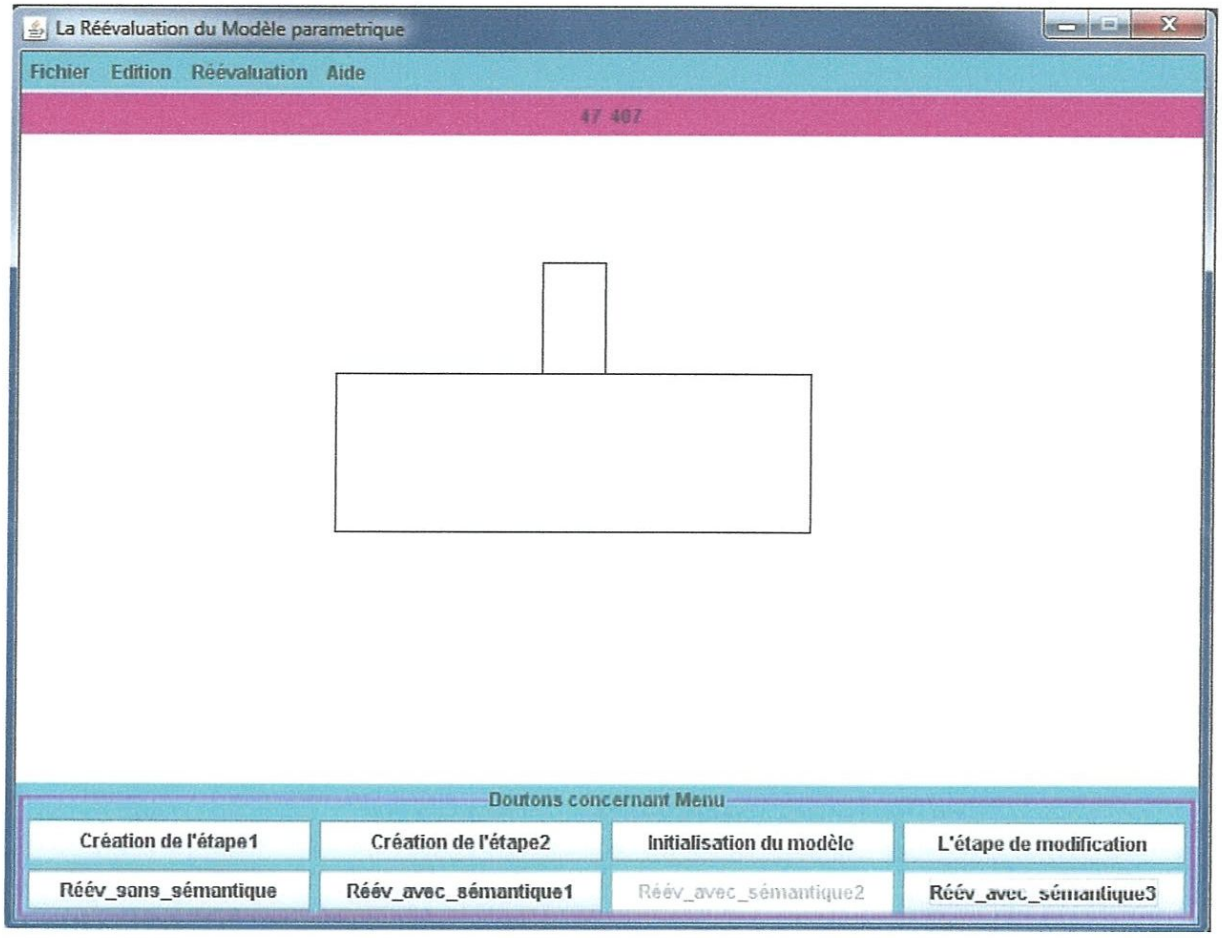


Figure 2.14 : *La réévaluation avec l'heuristique2.*

3.3. 3. Heuristique3 :

-Cette heuristique est différentes par rapport les heuristiques précédentes, elle consiste à modifier la topologie l'objet2 (non modifié).

-Ici l'objet 2 garde sa position initiale et il se prolonge vers l'objet modifié (Figure 3.10).

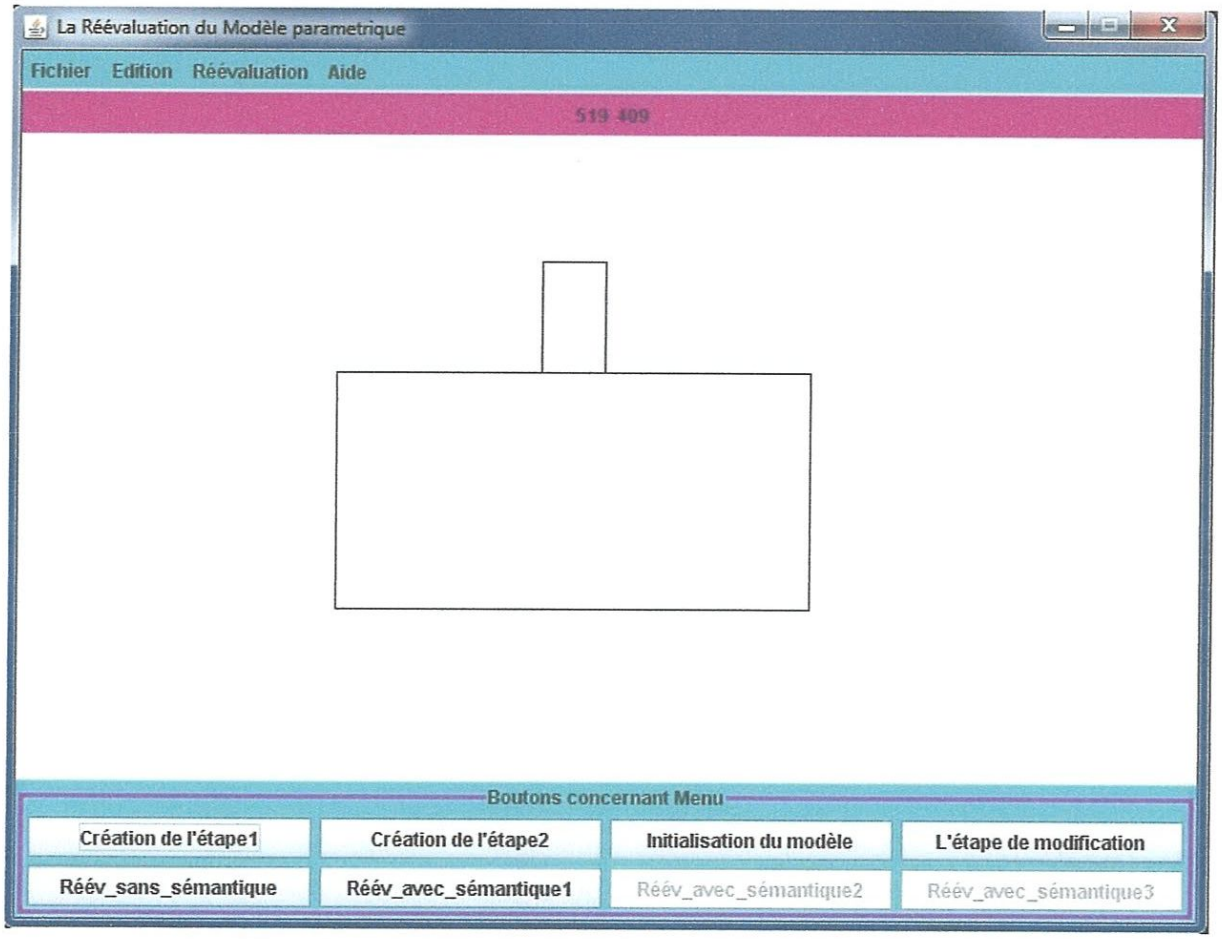


Figure 2.15 : *La réévaluation avec l'heuristique3.*

4. Conclusion :

Les approches proposées permettent de réévaluer le modèle paramétrique en utilisant les différentes heuristiques qu'on a vues, le processus de réévaluation permet de reconstruire l'objet soit en considérant la sémantique ou non).

Les heuristiques qu'on a proposées permettent dans certains cas comme on la vue dans les différents exemples de notre expérimentation d'éliminer l'ambiguïté qui peut survenir au niveau de l'objet réévalué.

Conclusion générale :

Aujourd'hui on ne discute plus l'importance et la difficulté de construire des systèmes de nomination persistante performants capables de rendre les modeleurs CAO (Conception Assistée par Ordinateur) capables d'assister l'utilisateur dans la construction des objets solides. Mais on cherche surtout à les améliorer en offrant des outils qui rendent les modeleurs CAO plus performants.

L'analyse de ce problème nous a amené à poser plusieurs questions sur la modélisation paramétrique : Faut-il proposer une nouvelle technique de nomination persistante ? Est-ce que la nouvelle technique pourrait garantir une réévaluation parfaitement adéquate aux souhaits du concepteur ?

Le but de ce travail est de montrer qu'il est possible d'arriver à proposer une solution sans avoir à redéfinir une nouvelle structure de nomination persistante. C'est-à-dire, ajouter à celles déjà existantes des heuristiques afin de remédier aux insuffisances citées précédemment

Trois grandes parties composent cette recherche:

i. La sauvegarde du processus de conception de l'objet initial.

ii. La génération toutes les solutions possibles susceptibles de sauvegarder la topologie globale de l'objet, en considérant les critères sémantiques de l'objet initial et lors de la réévaluation.

iii. L'affichage de ces solutions par ordre de pertinence décroissante, en utilisant des heuristiques fondées sur les attentes supposées du concepteur

La première partie de ce travail, qui est la base de tous les modeleurs paramétriques fondés sur l'historique des contraintes de conceptions, va sauvegarder tout le processus de conception afin de permettre à l'utilisateur concepteur de revenir au niveau de n'importe qu'elle étape de conception sans avoir besoin de reconstruire tout le processus.

La deuxième et la troisième partie, incluent quelques heuristiques permettant d'une part d'extraire tous les résultats possibles lors de la phase de réévaluation, et d'une autre part d'ordonner ces résultats. Ceci inclut un certain respect de la sémantique en conservant la topologie globale de l'objet.

En plus des avantages cités précédemment offerts par cette solution, il y a un autre concernant un gain important de temps grâce aux différentes simulations offertes par le système. Cependant les différentes études faites jusqu'à aujourd'hui concernent simplement les objets linéaires et n'incluent pratiquement pas les objets non linéaires. Ce qui pousse à espérer dans un proche avenir d'avoir l'occasion de s'intéresser à l'étude d'un système de nomination persistante qui sera capable de supporter de tels objets.

Références

[Naray2008a] Narayan, K. Lalit : *Computer Aided Design and Manufacturing*. New Delhi: Prentice Hall of India, ISBN 812033342X, p. 3-4, 2008 .

[FarinG] Farin, G.: *A History of Curves and Surfaces in CAGD*, Handbook of Computer Aided Design. Ed. North Holland, 2002.

[Pottmann] Pottmann, H.; Brell-Cokcan, S. and Wallner, J. *Discret surface for architecture design*, pp. 213–234 in *Curve and Surface Design*, Patrick Chenin, Tom Lyche and Larry L. Schumaker (eds.), Nashboro Press, ISBN 0-0-9728482-7-5, 2007.

[Babali06] Baba-ali Mehdi : *Système de nomination hiérarchique pour les systèmes paramétriques*, thèse de doctorat, Poitiers, France 2006.

[Agbod99] Agbodan,D, Marcheix,D, Pierra,G : *A Data Model Architecture For Parametrics* in *Journal for Geometry and Graphics*, Vol.3, N°.1, pp.17-38,1999.

[Agbod02] Agbodan,D, Marcheix,D, Pierra,G : *A Data Model Architecture For Parametrics* in *Journal for générique d'entités topologiques*, thèse de doctorat, Futuroscope Chasseneuil, France, 2002.

[Requicha 80] A.A.G. Requicha, "Representation of solid objects — Theory, methods, and systems", ACM Computing Survey, Vol. 12, N° 4, pages 437-464, December 1980.

[Sutherland 63] I.E. Sutherland, "A Man-Machine Graphical Communication System", AFIPS Spring Joint Computer Conference 23, pages 329-346, 1963.

[Kripa95a] Kripac,J : *A mechanism for persistently naming topological entities in history-based parametric solid models* (Topological ID System) in *Proceedings of Solid Modeling '95*, Salt Lake City, Utha USA, pp.21-30, 1995.

[Capoy96a] Capolylas,V, Chen,X, Hoffman,CM : *Generic naming in generative, constraint-based design Computer-Aided Design* Vol. 28 pp. 17-26.

[Shah95] Shah,J.J., Mäntylä,M : *Parametric and feature-based CAD/CAM* : Concepts, Techniques, Applications, John Wiley and Sons Inc., july 1995

[HJ92] C.M. Hoffmann and R. Juan. Erep : *An editable, high-level representation for geometric design and analysis*. In Selected and Expanded Papers from the IFIP TC5/WG5.2 Working Conference on Geometric Modeling for Product Realization, pages 129–164, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co.

[SV95] V. Shapiro and D.L. Vossler. *What is a parametric family of solids* ? In SMA' 95 :Proceedings of the third ACM symposium on Solid modeling and applications, pages 43–54, New York, NY, USA, 1995. ACM.

[PAAB+96] G. Pierra, Y. Ait-Ameur, F. Besnard, P. Girard, and J.C. Potier. *A general Framework for parametric product model within step and part library*. In European Conference Product Data Technology, pages 18–19, 1996.

[RS98] S. Raghothama, , and S. Shapiro. *Boundary representation deformation in parametric solid modeling*. ACM Transactions on Graphics, 17(4) :259–286, 1998.

[WN05] Y. Wang and B.O. Nnaji. *Geometry-based semantic id for persistent and Interoperable reference in feature-based parametric modeling*. Computer Aided design, 37 :1081–1093, 2005.

[Che95] X. Chen. *Representation, Evaluation and Editing of Feature-Based and Constraint-Based design*. PhD thesis, Purdue University, West Lafayette, Indiana, 1995.

[Mar06] D. Marcheix. *A persistent naming of shells*. International Journal of CAD/CAM, 6(1) :121–133, 2006.

[BFH+95] W. Bouma, I. Fudos, C.M. Hoffmann, J. Cai, and R. Paige. *Geometric constraint solver*. Computer-Aided Design, 27(6) :487–501, 1995.