

THÈSE

pour l'obtention du Grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
Faculté des Sciences Fondamentales et Appliquées
Diplôme National - Arrêté du 7 août 2006

École Doctorale ED 521 : Sciences et Ingénierie pour l'Information
Secteur de Recherche : Informatique et Applications

Présentée par :
Mehdi Baba-ali

Systeme de nomination hiérarchique pour les systèmes paramétriques

Directeur de Thèse : M. Yves BERTRAND
Co-directeur de Thèse : M. David MARCHEIX
Co-directeur de Thèse : M. Xavier SKAPIN

Soutenue le 22 Janvier 2010 devant la Commission d'Examen composée de :

M. Marc Neveu, Professeur des Universités, Université de Bourgogne, LE2IPrésident du Jury
Mme Dominique Bechmann, Professeur des Universités, Université de Strasbourg, LSIT Rapportrice
M. Christophe Schlick, Professeur des Universités, Université de Bordeaux, LABRIRapporteur
M. Guy Pierra, Professeur des Universités, ENSMA, LISIExamineur
M. David Marcheix, Maître de Conférences, Université de Poitiers, LISI Co-directeur de Thèse
M. Xavier Skapin, Maître de Conférences, Université de Poitiers, XLIM-SIC Co-directeur de Thèse

Remerciements

Mes remerciements vont, tout d'abord, à mes directeurs de thèse, David Marcheix et Xavier Skapin qui m'ont dirigé tout au long de cette thèse. Leurs disponibilités, leurs réactivités, leurs enthousiasmes et leurs conseils judicieux sont pour beaucoup dans le résultat final de ce travail. Merci également à Yves Bertrand pour son soutien, sa confiance et ses conseils avisés.

Je remercie les membres de mon jury de thèse, Dominique Bechmann, Christophe Schlick, Marc Neveu et Guy Pierra pour l'intérêt qu'ils ont porté à mon travail et pour l'avoir enrichi de toutes leurs remarques.

Un merci particulier et chaleureux à Guy Pierra, Yamine Aït-Ameur et Pascal Lienhardt pour leur accueil à Poitiers, leur soutien tant sur le plan moral que sur le plan administratif et grâce à quoi, j'ai pu poursuivre en thèse.

Merci à Nicolas Guiard pour sa contribution à ma thèse. Je remercie aussi particulièrement Guillaume Damiand pour sa collaboration dans la définition formelle d'une partie des opérations de base définies dans le cadre des cartes généralisées. Enfin, merci à Daniel Menevaux pour sa vivacité, la collaboration qu'on a eu ensemble dans la recherche d'un contrat industriel même si celui n'a pas vu le jour.

Parmi les membres du laboratoire SIC, je tiens tout spécialement à remercier Françoise Perrain et Françoise Guilbard pour leur enthousiasme et leur aide dans les problèmes administratifs ou autre. Merci aussi à Aldo, Wassim et Ghadir pour avoir partagé avec moi leur bureau durant ma thèse. Merci également à Alexandre, Carine, Gycelin, Martin, Marc et Olivier pour nos discussions sincères, parfois animées et souvent drôles. Sans oublier, Chaker, Olivier avec qui j'ai partagé des discussions forts intéressantes.

Parmi les membres du laboratoire LISI, je tiens à remercier Idir, Ahmed, Sybille, Chimène, Omar, Nabil, Loé et Christian avec qui je garde pour chacun un souvenir particulier. Merci pour Claudine pour sa sympathie et son aide dans diverses tâches administratives. Je tiens également à remercier Stéphane, Manu et Laurant pour leur gentillesse et leur disponibilité.

Je remercie Christophe qui a eu la gentillesse de lire ce rapport. Ses corrections éclairées ont été un plus dans la qualité de ce manuscrit.

Je remercie ma très chère mère d'être toujours là pour moi, quoi qu'il arrive et sans condition. Je remercie également ma famille et en particulier mon frère et mes deux sœurs ainsi que mes tantes et mes cousins pour leurs présences, leur soutien et le bol d'air qui m'ont procuré surtout pendant les périodes difficiles que j'ai traversé dans mon travail. Merci également à mon meilleur ami Rafik qui tiens une place importante dans mon cœur et qui a été là dans les périodes les plus difficiles. Merci de m'avoir supporté et je te souhaite la même chose pour la suite de ta thèse.

Table des matières

1	Introduction générale	1
1.1	Motivations	1
1.1.1	Contexte	1
1.1.2	Problématique	2
1.1.3	Objectifs	3
1.2	Trame du mémoire	3
1.2.1	État de l’art	4
1.2.2	Suivi des événements dans un modèle topologique	4
1.2.3	Système hiérarchique de nomination persistante	5
2	État de l’art	7
2.1	Vers un modèle paramétrique	7
2.1.1	Modélisation paramétrique	9
2.1.1.1	Modèle paramétrique équationnel (ou variationnel)	10
2.1.1.2	Modèle paramétrique fonctionnel	11
2.1.1.3	Modèle paramétrique hybride	12
2.1.2	Modélisation géométrique	12
2.1.2.1	Les modèles classiques	13
2.1.2.2	Les modèles topologiques	16
2.1.2.3	Les courbes et surfaces paramétriques	21
2.1.2.4	Les surfaces implicites	23
2.1.3	Modélisation par caractéristiques	24
2.1.3.1	Concept de caractéristique	25
2.1.3.2	Techniques de création	26
2.1.3.3	Limites	28
2.2	Problème de la nomination persistante	30
2.2.1	Nécessité d’un nom (1 ^{er} problème)	31
2.2.2	Nécessité d’un appariement entre noms (2 ^{ème} problème)	32
2.2.3	Capture des intentions de conception (3 ^{ème} problème)	32
2.3	Solutions proposées	33
2.3.1	Concepts communs	33
2.3.1.1	Distinction entre les entités invariantes et les entités contingentes	33
2.3.1.2	Distinction entre les entités avec et sans source	34
2.3.1.3	Nomination des entités invariantes issues d’extrusion ou de révolution	34
2.3.1.4	Nomination basée sur les entités invariantes	35

2.3.1.5	Utilisation d'un historique de noms	36
2.3.1.6	Nomination hétérogène des entités	38
2.3.1.7	Structure des plates-formes mises en œuvre	38
2.3.2	Points différents	38
2.3.2.1	Solutions de nomination avec un appariement global ou local . .	39
2.3.2.2	Solutions appropriées aux cas polyédriques et celles spécifiques aux cas non-planaires	39
2.3.2.3	Nomination des entités de dimension supérieure à 2	48
2.3.3	Analyse et limites des solutions existantes	50
2.4	Conclusion	52
3	Suivi des événements dans un modèle topologique	55
3.1	Introduction	55
3.2	Évolution de la topologie au sein du noyau géométrique	58
3.2.1	Mécanismes de suivi d'événements	59
3.2.1.1	Mécanisme <i>sci_élém</i>	61
3.2.1.2	Mécanisme <i>créa_élém</i>	62
3.2.1.3	Mécanisme <i>fus_élém</i>	63
3.2.1.4	Mécanisme <i>sup_élém</i>	64
3.2.1.5	Mécanisme <i>modif_élém</i>	65
3.2.2	Génération du journal de bord	66
3.3	Les opérations de base	69
3.3.1	Quelques notions supplémentaires autour des cartes généralisées	69
3.3.2	Suppression et contraction	71
3.3.2.1	Suppression	72
3.3.2.2	Contraction	76
3.3.2.3	Généralisation	77
3.3.3	Insertion et éclatement	79
3.3.3.1	Insertion	80
3.3.3.2	Éclatement	84
3.3.3.3	Généralisation	85
3.4	Implantation des mécanismes de suivi d'événements	87
3.4.1	Solution apportée par la structure de <i>tag</i>	87
3.4.2	Opérations élémentaires et mécanismes de suivi d'événements intégrés . .	89
3.4.2.1	Insertion d'une arête pendante dans une face	90
3.4.2.2	Scission d'une arête	93
3.4.2.3	Scission d'une face	95
3.4.2.4	Identification de faces	97
3.5	Conclusion	101
4	Système hiérarchique de nomination persistante	105
4.1	Introduction	105
4.2	Nomination persistante d'arêtes	106
4.2.1	Trois point-clés pour la caractérisation des arêtes	107
4.2.1.1	Utilisation d'entités invariantes	108

TABLE DES MATIÈRES

4.2.1.2	Exploitation des informations issues de différents niveaux de granularité	108
4.2.1.3	Utilisation d'informations géométriques	108
4.2.2	Caractérisation des arêtes	110
4.2.2.1	Caractérisation des arêtes invariantes	110
4.2.2.2	Caractérisation des arêtes contingentes	111
4.2.3	Appariement des arêtes	118
4.2.3.1	Appariement des arêtes invariantes	118
4.2.3.2	Appariement des arêtes contingentes	119
4.3	Système de nomination hiérarchique	125
4.3.1	Notion d'agrégat	125
4.3.2	Caractérisation des agrégats	127
4.3.2.1	Caractérisation des agrégats invariants	127
4.3.2.2	Caractérisation des agrégats contingents	128
4.3.3	Appariement des agrégats	131
4.3.3.1	Appariement d'agrégats invariants d'entités	131
4.3.3.2	Appariement d'agrégats contingents d'entités	131
4.4	Conclusion	133
5	Conclusion générale	135
5.1	Travail réalisé	136
5.1.1	Apports et résultats	137
5.1.2	Domaine de validité des résultats	138
5.2	Perspectives	138
5.3	Synthèse	139

Chapitre 1

Introduction générale

Au début des années 60, la modélisation géométrique est apparue. Depuis, elle n'a eu de cesse de se développer, aidée en cela par les capacités de calcul de plus en plus grandes. Aujourd'hui, la représentation sous forme informatique et la manipulation virtuelle de formes géométriques d'objets sont présentes dans des domaines aussi divers que l'architecture, la mécanique, les jeux vidéo, etc. Parallèlement, de nouveaux besoins sont apparus dans le domaine de l'industrie, notamment des besoins en terme de gestion du cycle de vie d'un produit ou de réduction des coûts de production. Les systèmes classiques de modélisation géométrique ne suffisent plus. En effet, lorsqu'il y a besoin de faire varier les modèles à base de caractéristiques pour rectifier des détails dans la conception, les modèles précédents restent impuissants. En raison de cela, il y a eu émergence des systèmes paramétriques (Pro/ENGINEER, I-DEAS, CATIA, TopSolid, etc.). Dans ce type de système la conception repose sur un modèle qui conserve à la fois la représentation explicite de l'objet conçu, mais également l'ensemble des opérations de modélisation utilisées pour le concevoir.

1.1 Motivations

1.1.1 Contexte

Dans les processus de conception actuels, faire collaborer plusieurs domaines d'expertise est une approche établie. Le principe est d'utiliser différents types d'informations, notamment physiques (par exemple, le type de matériaux utilisés dans la fabrication de pièces mécaniques) ou fonctionnelles (par exemple, des informations ayant trait aux aspects aérodynamiques d'un véhicule) pour obtenir des objets conformes aux cahiers des charges. L'objectif escompté est de s'abstraire des limites liées à l'utilisation des modèles géométriques sous-jacents dans la perspective de simulations, visualisations, reconstructions, etc. Avant d'aboutir à ce type de conception, la modélisation que l'on pourrait qualifier de *purement* géométrique d'objets ou de scènes réalistes est passée par trois étapes :

1. l'apparition et le développement des *modèles géométriques* pour répondre aux besoins de représentation et de manipulation virtuelles des objets réels ;
2. l'introduction des *modèles à base de caractéristiques* qui permettent de représenter des informations sémantiques autres que géométriques (perpendicularité, courbure de surfaces, incidence, adjacence, etc.). Ces types de modèles sont essentiellement apparus dans le domaine de la conception assistée par ordinateur pour la gestion du cycle de vie d'un produit.

3. la mise en œuvre de *modèles paramétriques* afin d'*automatiser* les processus qui permettent la variation des modèles géométriques en modifiant les paramètres des opérations de modélisation.

Prendre en compte différents types d'informations permet d'intégrer des spécificités inhérentes à des domaines aussi variés que la géologie, l'architecture ou même l'archéologie. D'autre part, le fait de pouvoir générer plusieurs variantes du même objet ou de la même scène modélisée renforce l'intérêt grandissant des systèmes paramétriques.

Nous nous intéressons essentiellement à la modélisation paramétrique, et, en particulier à la fonctionnalité qu'offre ce type de modèle pour faire varier la topologie des objets ou des scènes modélisées.

1.1.2 Problématique

Dans un modèle paramétrique à base d'historique, la construction de représentations géométriques est un processus piloté par un ensemble d'opérations de modélisation. Ce processus s'effectue par étapes, et une opération de modélisation, à une étape donnée, a besoin de l'instance courante de la géométrie résultant des étapes antérieures. Par conséquent, un système paramétrique conserve non seulement la géométrie explicite d'un objet (représentation concrète appelée *noyau géométrique*) mais également l'ensemble des opérations de modélisation utilisées pour le concevoir (représentation abstraite) appelée *spécification paramétrique*. Plusieurs auteurs ont étudié la structure interne des modèles de données paramétriques. Leurs contributions ont mené à l'apparition de différentes représentations [SV95, PAAB⁺96] de ces modèles et mis en exergue leurs structures mathématiques sous-jacentes [RS98, WN05]. De plus, certains d'entre eux ont présenté et discuté divers problèmes liés soit à la sémantique des opérations de modélisation [Che95, Mar06], soit à la gestion des contraintes de modélisation [BFH⁺95].

Les liens entre le noyau géométrique et la spécification paramétrique sont assurés par des références de la spécification paramétrique (paramètres d'entrée des opérations de modélisation, cibles de contraintes, etc.) et qui désignent des entités de la représentation géométrique. Plus précisément, lorsqu'une entité de la représentation géométrique est référencée par une opération de modélisation, cette référence doit désigner le « même » objet même si le modèle est réévalué¹. Le problème est donc d'assurer la validité des liens entre la spécification paramétrique et le noyau géométrique, c'est-à-dire que les références utilisées restent les « mêmes » ou du moins qu'elles désignent, pour le concepteur, des entités équivalentes. Ce problème est plus connu sous le terme de *nomination persistante* ou de *nomination topologique*.

Pour assurer la validité des références employées dans la spécification paramétrique, il est d'abord nécessaire de caractériser de façon unique et non ambiguë les entités topologiques de la représentation géométrique. Ensuite, il est impératif de mettre en œuvre un mécanisme qui fait correspondre, ou « apparie » les entités des modèles initial et réévalué, puisque ces modèles sont des variantes représentant les mêmes objets. De plus, la prise en compte des différentes sémantiques associées aux opérations de modélisation nécessite de considérer non seulement les entités topologiques atomiques (sommets, arêtes, faces, etc.) mais également les « agrégats » de ces entités en toute dimension.

1. Par réévaluation d'un modèle paramétrique nous entendons la modification d'un certain nombre de paramètres puis la reconstruction de l'objet modélisé. Lorsqu'il y a modification des opérations de modélisation, c'est-à-dire modification de la spécification paramétrique, nous parlons d'*édition* du modèle paramétrique.

Plusieurs solutions ont été proposées pour résoudre le problème de la nomination persistante. La plupart d'entre elles se déploient dans un espace $3D$ et reposent sur un certain nombre de concepts communs :

- La structure des noms attribués aux entités topologiques atomiques (sommets, arêtes et faces) : cette structure peut être décomposée en deux parties. La première est essentiellement utilisée dans le cas des objets planaires et utilise le voisinage topologique de chaque entité [Che95, Kri95, AMP00]. La seconde intervient dans le cas des objets non-planaires et utilise des informations géométriques pour lever l'ambiguïté apparaissant lorsque plusieurs entités possèdent le même voisinage [WZZZ01, BNB05, WN05].
- L'utilisation exclusive des faces dans les processus de caractérisation et d'appariement : ce type d'entités est considéré plus stable en $3D$ que les arêtes et les sommets. Les faces sont alors employées pour renforcer la désignation d'entités moins stables car subissant des transformations topologiques non prévisibles.
- L'enrichissement de la structure traditionnelle des systèmes paramétriques en rajoutant une couche qui permet la gestion des noms (attribution, comparaison et appariement).

Toutefois, ces mêmes solutions présentent un certain nombre de limites. En effet, la structure du voisinage topologique d'un nom est différente selon la dimension de l'entité considérée. D'où, le caractère hétérogène des mécanismes de nomination déployés. La structure du nom et le mécanisme d'appariement sont différents, qu'il s'agisse des faces, des arêtes ou des sommets. Cette hétérogénéité limite l'extension des mécanismes de nomination aux entités et aux agrégats d'entités de toute dimension. Le peu de solutions développées pour appréhender d'avantage la sémantique des opérations de modélisation [Mar06] en témoignent puisque les seuls agrégats traités sont les coques (agrégats de faces).

1.1.3 Objectifs

Afin de repousser les limites des solutions proposées pour la résolution du problème de la nomination persistante, nous proposons un système hiérarchique de nomination persistante. Le but est de caractériser puis apparier les entités et les agrégats d'entités de toute dimension, y compris celles de dimension 4 (hyper-volume), voire au-delà. Or, les systèmes actuels de nomination basée sur les faces ne permettent pas cette généralisation. En effet, le fait de définir tout, en utilisant exclusivement des faces, rend spécifique les processus de caractérisation et d'appariement à chaque dimension d'entité. Pour y remédier, il est nécessaire d'homogénéiser ces processus en se basant sur les arêtes au lieu des faces et bâtir sur la base des arêtes un système hiérarchique de nomination persistante. Notre méthode consiste à :

- d'abord, caractériser les arêtes puis les apparier de façon robuste en exploitant les données issues de leurs voisinages et les propriétés géométriques qui leur sont associées (planaires ou non-planaires, orientation, etc.) ;
- ensuite, caractériser et apparier récursivement les entités et agrégats d'entités, de dimension supérieure, en s'appuyant sur les résultats obtenus sur les entités déjà traités.

1.2 Trame du mémoire

Le fil directeur de ce mémoire est la modélisation paramétrique autour de laquelle vont se nouer divers aspects tels que la géométrie, la modélisation à base de caractéristiques, les

contraintes, le problème de la nomination persistante, les solutions proposées pour résoudre ce problème, leurs spécificités et insuffisances, les aspects algorithmiques, etc. Pour dissenter sur tous ces aspects, nous avons organisés ce mémoire autour de trois chapitres.

1.2.1 État de l'art

Le deuxième chapitre décrit l'évolution qui a conduit à la définition des systèmes paramétrique. Il présente l'un des problèmes majeurs associés à la modélisation paramétrique, à savoir le problème de la nomination persistante, et dresse les limites des solutions, qui sont proposées pour le résoudre.

Dans un premier temps, nous présentons les critères qui caractérisent les différents types de systèmes paramétriques. Puis, nous décrivons la genèse de tels systèmes en passant en revue les modèles géométriques et les modèles à base de caractéristiques.

Dans un deuxième temps, nous abordons le problème de la nomination persistante qui apparaît lorsqu'un utilisateur souhaite générer plusieurs variantes du même modèle par un changement de paramètres. Nous illustrons notamment la nécessité de disposer d'une part d'un nom muni d'une structure suffisamment riche pour caractériser les entités topologiques de façon unique et non ambiguë capable d'appréhender les intentions de conception ; et, d'autre part, d'un mécanisme d'appariement suffisamment robuste pour mettre en correspondance les entités du modèle initial et celles du modèle réévalué.

La dernière partie est consacrée aux solutions proposées pour résoudre le problème de la nomination persistante. Nous n'abordons pas la question par une simple énumération de ces solutions, mais nous les analysons sur la base de critères déterminés. Ainsi, les concepts communs et les différences des solutions autour desquelles s'articule la recherche sur le problème de la nomination persistante sont présentés. À partir de cette analyse, nous mettons en exergue les limites constatées qui constituent le cœur de notre problème, c'est-à-dire, la nomination de façon homogène les entités et les agrégats d'entités de toute dimension.

1.2.2 Suivi des événements dans un modèle topologique

Pour déterminer le nom *adéquat* de chaque entité sur lequel va se baser un mécanisme d'appariement, il est impératif de connaître l'évolution des entités dans un noyau géométrique d'un système paramétrique.

Le troisième chapitre est consacré à cette question en décrivant, dans un premier temps, des mécanismes de suivi des événements (création, suppression, scission, fusion et modification) survenus sur les entités topologiques du noyau géométrique, lors des opérations de modélisation.

Dans un deuxième temps, nous présentons le procédé qui permet de regrouper ces événements au sein d'une structure que nous appelons « journal de bord ». L'implantation des mécanismes de suivi nécessite d'utiliser un modèle topologique robuste et capable d'intégrer les caractéristiques de ces mécanismes. Notre choix s'est porté sur le modèle des cartes généralisées [Lie94]. Notre étude montre que quatre opérations topologiques élémentaires sont nécessaires pour représenter ces mécanismes : la suppression et la contraction de cellules ([DL03]), et l'insertion et l'éclatement de cellules que nous avons définies durant nos travaux.

Finalement, nous validons l'intégration des mécanismes de suivi dans les cartes généralisées à travers l'étude des opérations booléennes, caractéristiques d'opérations de modélisation géométriques complexes.

1.2.3 Système hiérarchique de nomination persistante

Le quatrième chapitre répond aux objectifs fixés en décrivant un système hiérarchique de nomination persistante. Ce système met en œuvre la caractérisation et l'appariement des entités et des agrégats d'entités de toute dimension. Le niveau le plus bas dans la structure hiérarchique sous-jacente à notre système de nomination est le niveau des arêtes. Dès lors, la caractérisation et l'appariement des entités de plus hautes dimensions se base sur celles des arêtes. Par conséquent, le niveau des arêtes revêt une importance certaine dans notre solution.

Dans un premier temps, nous présentons une vue d'ensemble du modèle mettant en évidence les liens entre les différents niveaux de la structure hiérarchique.

Ensuite, nous abordons le niveau des arêtes de notre système hiérarchique de nomination. Nous définissons alors la structure de nom d'une arête en se basant sur son voisinage topologique et sur des informations géométriques permettant de lever les ambiguïtés qui peuvent apparaître. Le principe de base consiste d'une part à utiliser toutes les entités et les agrégats d'entités invariants appartenant au voisinage topologique et d'autre part, à considérer la forme des objets manipulés pour en extraire les informations géométriques. Pour l'appariement, nous définissons un mécanisme qui (1) détermine dans l'ensemble des arêtes, celles qui doivent être appariées entre elles, (2) évalue le degré d'inclusion des arêtes du modèle initial dans celles du modèle réévalué, et inversement, et (3) recouper les informations de l'étape précédente pour en déduire les appariements effectifs entre arêtes. Par ailleurs, dans le cas où les arêtes présentent le même voisinage topologique, le mécanisme d'appariement que nous avons mis en œuvre compare les informations géométriques qui leurs sont associées.

Dans la dernière partie, nous exploitons les noms utilisés pour caractériser les arêtes et les résultats obtenus par le mécanisme d'appariement précédent pour caractériser et appairer les entités et les agrégats d'entités de dimension supérieure.

Finalement, nous présentons la conclusion générale et les perspectives envisagées de ce travail dans le cinquième chapitre.

Chapitre 2

État de l'art

Sommaire

2.1	Vers un modèle paramétrique	7
2.1.1	Modélisation paramétrique	9
2.1.2	Modélisation géométrique	12
2.1.3	Modélisation par caractéristiques	24
2.2	Problème de la nomination persistante	30
2.2.1	Nécessité d'un nom (1 ^{er} problème)	31
2.2.2	Nécessité d'un appariement entre noms (2 ^{ème} problème)	32
2.2.3	Capture des intentions de conception (3 ^{ème} problème)	32
2.3	Solutions proposées	33
2.3.1	Concepts communs	33
2.3.2	Points différents	38
2.3.3	Analyse et limites des solutions existantes	50
2.4	Conclusion	52

2.1 Vers un modèle paramétrique

Bien souvent, les avancées technologiques sont guidées par des besoins exprimés dans divers domaines (par exemple, la conception numérique des pièces mécaniques dans le domaine de l'industrie des machines électroniques) pour atteindre des objectifs spécifiques (dans l'exemple précédant, le but est de contrôler des machines à commandes numériques). L'informatique graphique n'échappe pas à la règle. En effet, la nécessité de manipuler graphiquement des objets a rapidement posé le problème de leur représentation en machine. Aussi, pour voir le rôle, l'intérêt et les problèmes liés aux systèmes paramétriques dans le processus de modélisation, nous allons dresser un bref aperçu de l'évolution historique en informatique graphique. Puis, nous allons détailler chaque type de modélisation apparaissant dans cette évolution, qui a mené à l'apparition des systèmes paramétriques.

Dans ce contexte, Sutherland [Sut63] fut le premier à proposer une solution avec son système 2D de Dessin Assisté par Ordinateur (DAO) qu'il a nommé *sketchpad*. Depuis lors, en l'espace de 25 ans, le domaine de la modélisation géométrique a connu un large essor, caractérisé par un foisonnement hétéroclite d'approches et de modèles. Les principaux modèles utilisés de nos jours sont :

1. les modèles dits « classiques » [BP07] qui, historiquement, avaient pour objectif la représentation de solides. On peut citer deux approches qui ont connu une très large diffusion : (a) la représentation de solides par frontières Brep¹ dont le premier modèle à base de fil de fer a été développé par Braid à l'université de Cambridge [Bra75] et, (b) le modèle de construction géométrique des solides CSG², basé sur un ensemble d'opérations booléennes et qui a été développé par Requicha et Voelcker à l'université de Rochester [Req77] ;
2. les modèles à base topologique [DL89, Lie91, Bri93, Lie94, FMP03, SDL06] qui ont peu à peu remplacé les modèles classiques, permettent de représenter toutes sortes d'objets, bien au-delà des seuls solides ;
3. les courbes et surfaces paramétriques (Bézier [GR74, Béz86], B-splines [dB78], NURBS³ [PT97], etc.) qui sont, aujourd'hui encore, les modèles surfaciques les plus utilisés en Conception Assistée par Ordinateur (CAO) ;
4. les surfaces implicites [WMW86, BW90, TO99, CBC⁺01] qui offrent une représentation volumique des objets et qui assurent que leur bord est une surface fermée sans auto-intersection.

D'un point de vue chronologique, les travaux se sont déroulés de façon non coordonnée en abordant en même temps les modèles de représentation d'objets bidimensionnels ($2D$), tridimensionnels ($3D$) voir n -dimensionnels (nD). Chacun de ces modèles comporte des avantages et des limites propres. Tout dépend des traitements et des objets qu'on souhaite construire, visualiser et/ou modifier.

Cependant, aucun d'entre eux ne permet l'intégration explicite de caractéristique de forme (rainure, trou, arrondi, etc.) et/ou de conception (les détails de fonctionnement d'une pièce mécanique, les propriétés matérielles, la tolérance d'un objet modélisé, etc.), encore moins des contraintes⁴ géométriques à faire respecter. De plus, les modèles géométriques n'utilisent pas un langage ou des opérations inspirées de l'expérience métier d'un utilisateur (absence de sémantique). Les modélisations par contraintes [But79] et par caractéristique [Kyp80, Woo82] sont donc apparues à la fin des années 70 et au début des années 80, pour combler ce manque.

Par la suite, la recherche est devenue plus prolifique pour ces deux types de modélisation. Les systèmes de modélisation par contraintes [Sch02, WSMF06] sont plus adaptés à la modélisation géométrique en 2 dimensions. En effet, si définir un système bien contraint en $2D$ n'est déjà pas chose aisée, en $3D$, cela s'avère être encore plus complexe. D'autre part, les solveurs (outils permettant l'analyse du système de contraintes pour détecter de possibles erreurs, et pour *planifier*⁵ la résolution quand le système est correct) rencontrent des problèmes de résolution plus fréquent en $3D$ qu'en $2D$. Les systèmes à base de caractéristiques de forme développés, à la fois dans les laboratoires universitaires de recherche [Min85, PW87, CD88, SR88a] et dans le monde industriel (CasCade, CATIA, Pro/ENGINEER) sont, quant à eux, pour la plupart destinés à la CAO (Conception Assistée par Ordinateurs), dans laquelle la normalisation des

1. Brep : Boundary representation

2. CSG : Constructive Solid Geometry

3. NURBS : B-spline rationnelle non-uniforme

4. Selon Shah et Mäntylä [SM95], une contrainte est une propriété déclarée, que certaines entités doivent vérifier.

5. La planification (aussi appelée décomposition) consiste à déterminer un ordre pour la résolution des composants du modèle : le système est décomposé en sous-systèmes qui doivent être résolus dans un certain ordre, puis les solutions des sous-systèmes sont combinées pour produire une (ou plusieurs) solution(s) du système entier.

2.1. VERS UN MODÈLE PARAMÉTRIQUE

caractéristiques de forme joue un rôle important (par exemple, la norme ISO⁶ [ISO WD 10303-48 :92] s'intéresse aux caractéristiques de forme répétitives, c'est-à-dire, l'arrangement de caractéristiques de forme simples). Une caractéristique est porteuse de sémantique et cette dernière est par nature spécifique à un point de vue technique particulier. L'utilisation de caractéristiques de formes et leurs nomination est un aspect important et novateur dans le domaine de la modélisation géométrique, car il permet de formaliser et de prendre en compte une partie de l'intuition du concepteur. Toutefois, ce genre de système ne se suffit pas à lui même s'il ne garde pas trace de l'historique des opérations de modélisation.

En effet, il est intéressant de pouvoir réévaluer le processus de modélisation lorsque le concepteur s'aperçoit qu'il doit modifier un paramètre au niveau d'une caractéristique. Pour ce faire, un système *paramétrique*⁷ sauvegarde l'historique de modélisation à travers un ensemble d'opérations paramétrées. Chaque opération porte une sémantique particulière (comme c'est le cas pour les caractéristiques précédemment citées) et référence les entités contenues dans un noyau géométrique.

La description que nous allons effectuer des différents types de modélisation en informatique graphique, n'est pas une simple énumération. Chaque famille de modèles (modélisation géométrique, modélisation par caractéristiques, modélisation à base de contraintes) est décrite de façon plus ou moins détaillée. Au travers d'un ou plusieurs éléments de chaque famille, nous allons préciser dans quelle mesure ces derniers sont des modèles paramétriques. Enfin, nous allons mettre en évidence les difficultés à maintenir les relations entre les différents éléments d'un système paramétrique.

2.1.1 Modélisation paramétrique

Lorsqu'on passe en revue la bibliographie autour des systèmes paramétriques [Rol91, AM95, SM95, BS01, HK01], on trouve des termes aussi divers et variés que la modélisation générative, la modélisation déclarative, les modèles à base de caractéristiques de forme, les modèles à base de contraintes, les modèles *history based*, la modélisation variationnelle, la modélisation procédurale, etc. Dans leur ensemble, ces modèles désignent toutes les propriétés que l'ont pourrait attendre d'un système paramétrique idéal. Cependant, lorsqu'ils sont pris un par un, ce n'est plus tout à fait le cas. Une clarification de ce qu'est réellement un système paramétrique s'impose donc.

Un modèle paramétrique est une structure à double couche. La première qui s'appelle couche géométrique ou noyau géométrique, contient la *représentation explicite* des objets modélisés, en d'autres termes, la géométrie que nous allons largement évoquée dans la section 2.1.2. La seconde, qui s'appelle *spécification paramétrique*, historique de construction, *représentation implicite* ou tout simplement *spécification*, contient un ensemble de paramètres et un processus permettant de recalculer la représentation explicite lorsque les paramètres sont modifiés. Par analogie avec la programmation, il est possible de considérer le paramétrique, comme une structure duale avec d'un côté le programme et de l'autre les données.

Les modèles paramétriques peuvent être classés en deux grandes catégories, suivant leurs structures mathématiques sous-jacentes et les différentes problématiques auxquelles ils se réfèrent [PPG94]. Aussi, nous pouvons distinguer l'approche fonctionnelle et l'approche équationnelle (ou variationnelle).

6. ISO est le sigle de *International Organization for Standardization*

7. dit history-based

2.1.1.1 Modèle paramétrique équationnel (ou variationnel)

Un tel modèle met en relation les entités topologiques par un système de contraintes (distance, tangence, parallélisme, etc.). Puis, un solveur permet de résoudre ces contraintes en construisant un système d'équations. L'ensemble de ces équations est alors résolu de manière simultanée en utilisant des méthodes regroupées en trois classes : les méthodes géométriques [But79, AM88, DMS98], les méthodes combinatoires [Owe91, FH97, JaSRMvVp04] et les méthodes numériques probabilistes [FMJ05, MF06].

Les modèles basés sur les contraintes [Sch02, JTNM06, WSMF06] permettent une conception flexible d'objets géométriques en variant les valeurs associées aux contraintes. Les contraintes algébriques, telles que les équations algébriques et les contraintes géométriques (perpendicularité, alignement, etc.) sont communément utilisées pour exprimer l'intention de conception. Un système typique de modélisation par contraintes permet à un utilisateur d'esquisser une géométrie approximative, de spécifier un ensemble de contraintes puis de déduire à partir de ces contraintes les valeurs appropriées à la géométrie souhaitée.

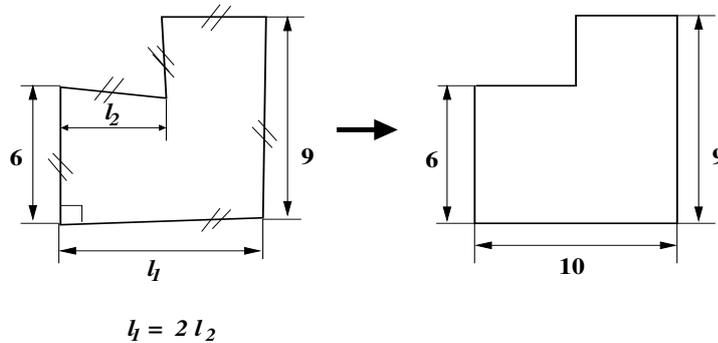


Figure 2.1: Un exemple d'une modélisation paramétrique équationnelle.

La figure 2.1 montre un exemple 2D simple de modélisation par contraintes. Un utilisateur esquisse le modèle à gauche de la figure, puis ajoute des contraintes à cette esquisse, et enfin donne les valeurs nécessaires pour quelques contraintes. Notons que les segments marqués par // sont destinés à être parallèles. La résolution de ces contraintes produit alors le modèle de droite de la figure 2.1.

D'un point de vue mathématique [PAAB⁺96], un modèle paramétrique équationnel se traduit par la relation suivante : soit \mathbf{p} l'ensemble des paramètres définis sur un domaine $D \subset P$ (souvent, P est égal à \mathbb{R}^n) et soit \mathbf{v} l'ensemble des variables (points, courbes, surfaces, valeurs numériques, etc.) qui doivent être évaluées pour produire une instance explicite. \mathbf{v} appartient à V , l'ensemble de toutes les formes possibles. Ainsi, un modèle équationnel est une équation de la forme : $E(\mathbf{p}, \mathbf{v}) = 0$; $\mathbf{p} \in D$ et $\mathbf{v} \in V$ où E est un opérateur qui a comme paramètres les éléments \mathbf{p} et \mathbf{v} . E représente également un système d'équations (issues des contraintes) et qui ne sont en général ni linéaires, ni convexes.

Cette approche présente deux intérêts majeurs :

- la possibilité d'utiliser et de traiter des contraintes cycliques ;
- la simplicité de mise en œuvre en 2D, en particulier pour les esquisses (*sketch*).

En revanche, cette approche présente plusieurs limites intrinsèques :

- les méthodes de résolution de contraintes augmentent en complexité lorsqu'on passe en

2.1. VERS UN MODÈLE PARAMÉTRIQUE

3D. Les raisons sont multiples. Parmi celles-ci, on peut citer le fait que la recherche de toutes les solutions devient souvent impossible dès que le degré de certaines équations excède quatre. De même, les méthodes de résolution de contraintes sont très difficilement généralisables de la 2D vers la 3D.

- fréquemment, le système d'équations possède plusieurs solutions. Dans ce cas, vers laquelle converge-t-il ?
- les modèles paramétriques équationnels sont limités dans leurs capacités à traiter les modèles spécifiés de façon incomplète (système sous-contraint) ;
- l'emploi de façon exclusive de contraintes non orientées⁸ interdit l'utilisation d'opérations telles que l'extrusion, les opérations booléennes, etc. En effet, dans un contexte variationnel, les entités sont d'abord construites (comme le sont les arêtes dans la figure 2.1) puis contraintes, ce qui n'est pas envisageable pour une extrusion par exemple.

2.1.1.2 Modèle paramétrique fonctionnel

Dans cette approche, la définition de la forme finale de l'objet qu'on souhaite modéliser s'effectue en plusieurs étapes. Chacune d'entre elles fait appel à une fonction de modélisation qui, à partir des paramètres et des entités déjà générées, permet d'engendrer une nouvelle instance de la géométrie.

La figure 2.2 montre un exemple de construction d'un cercle C centré en P_3 et tangent à une droite D passant par les points P_1 et P_2 . Un utilisateur esquisse le modèle de la figure en faisant appel, de façon séquentielle, aux fonctions $D = \text{Droite}(P_1, P_2)$ et $C = \text{Cercle}(P_3, D)$. La modification du point P_1 entraîne d'abord la réévaluation de D, puis celle de C. Une modification de P_3 n'entraîne quant à elle que la réévaluation de C.

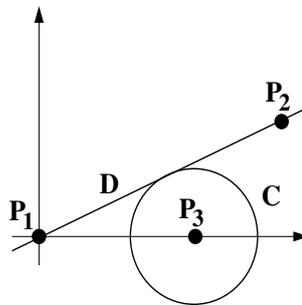


Figure 2.2: Un exemple de modélisation paramétrique fonctionnelle.

Mathématiquement [PAAB⁺96], un modèle paramétrique fonctionnel est une fonction :

$$F : \begin{cases} D \rightarrow V \\ P \rightarrow V = F(P) \end{cases}$$

8. Une contrainte orientée s'exprime sous la forme d'une fonction ou d'une procédure. Elle ne fixe pas de règles à ne pas transgresser mais définit les entités topologiques (sommets, arêtes, surfaces, etc.) de façon ordonnée (d'où le terme orientée) les unes par rapport aux autres. Comme exemple, prenons le cas d'un rectangle de longueur Long et de largeur Larg. Si la valeur de Larg est déterminée dans une étape préalable. Une contrainte orientée peut s'exprimer sous la forme : $\text{Long} = 2 \times \text{Larg}$. Tandis qu'une contrainte non orientée dans la même configuration peut s'exprimer sous la forme d'une équation algébrique : $\text{Long} - (2 \times \text{Larg}) = 0$. Dans le premier cas, la valeur de la largeur est déterminée avant celle de la longueur alors que dans le second les deux valeurs sont déterminées indifféremment l'une par rapport à l'autre.

où F est la fonction qui définit l'instance finale de la géométrie à partir des valeurs de paramètres. La fonction F est toujours exprimée comme une composition de fonctions : $F = f_n \circ f_{n-1} \circ \dots \circ f_1$. Bien que le domaine D ne soit en général pas spécifié, le modèle paramétrique définit une et une seule instance et ce, quelles que soient les valeurs des paramètres d'entrée, ce qui le rend déterministe. À l'inverse des systèmes paramétriques équationnels, les systèmes fonctionnels connaissent un très large essor, notamment pour les raisons suivantes :

- ces systèmes sont totalement déterministe et rendent donc, par définition, un résultat unique ;
- ces derniers peuvent être utilisés en $2D$, en $3D$, voire en nD ;
- les opérations de modélisation dans ce type de systèmes sont porteuses de sémantiques comme celles véhiculées par les caractéristiques de forme.

2.1.1.3 Modèle paramétrique hybride

Chaque type de modélisation paramétrique offre des avantages et des inconvénients propres (voir la table 2.1 pour plus de détails). À partir de ce constat, il est fréquent de trouver dans les systèmes paramétriques de CAO une utilisation conjointe des approches fonctionnelle et variationnelle. L'objectif est de tirer avantage de leurs complémentarités (privilégier l'approche fonctionnelle en $3D$ et l'équationnelle en $2D$).

	Modèle équationnel	Modèle fonctionnel
Nombre de solutions	0 : n	0 : 1
Contraintes orientées	-	+
Contraintes couplées	+	-
Structures répétitives	-	+
Esquisse (à main levée)	+	-
Mise en œuvre $3D$	-	+

Table 2.1: Avantages et inconvénients des modèles paramétriques équationnels et fonctionnels.

2.1.2 Modélisation géométrique

Le terme « modèle géométrique » est entendu au sens d'un ensemble d'outils mathématiques (en particulier la géométrie analytique) permettant de définir (en termes de plongement et de topologie), dans la machine, la forme de ce que sera un objet ou un ensemble d'objets. Un objet peut être plus ou moins complexe (pièce mécanique, couche géologique, organe humain, etc.) et modélisé de manière plus ou moins fine (modélisation multi-échelle). Dans cette optique, la modélisation géométrique ne prend pas en compte les aspects fonctionnels des objets modélisés. Au fil des années (un quart de siècle environ), cette technologie a su se faire une place dans des domaines aussi divers et variés que la CAO, l'architecture, la médecine, l'animation, la géologie, la biologie et bien d'autres domaines.

2.1.2.1 Les modèles classiques

CSG : la construction géométrique des solides

La méthode dite CSG (Constructive Solid Geometry) permet de créer des modèles géométriques en fournissant un ensemble de primitives élémentaires (cube, bloc, cône, cylindre, tore et parfois des demi-espaces), un ensemble d'opérations booléennes *régularisées*⁹ (union, intersection, différence) et des transformations rigides (translation ou rotation). Un solide est représenté par l'intermédiaire d'un arbre (voir la figure 2.3) où les feuilles sont les primitives et les nœuds internes représentent des opérations booléennes ou des transformations rigides.

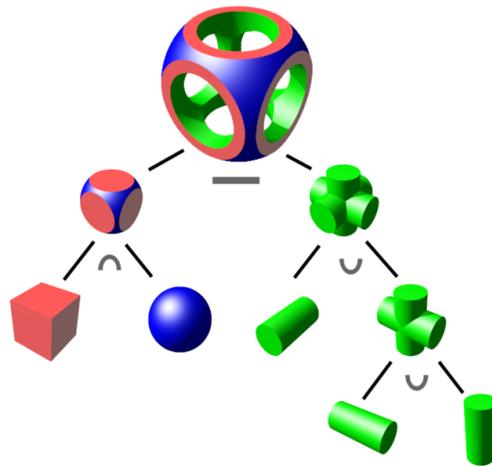


Figure 2.3: Un arbre CSG et l'objet qu'il modélise. On commence par joindre (union) le cube et la sphère, puis on joint en deux étapes les 3 cylindres de droite et enfin on creuse (différence) un trou ayant la forme du solide de droite sur le solide de gauche.

Puisqu'un arbre CSG conserve tout l'historique de construction d'un objet, cet arbre permet de générer des variantes dudit objet. Il suffit pour cela de modifier les dimensions et/ou les positions des primitives. C'est en cela que ce modèle est paramétrique. Toutefois, les limites intrinsèques des modèles CSG quant à l'application de certains algorithmes géométriques ont grandement nuit à leurs succès (par exemple, la non unicité de la représentation CSG ne permet pas une comparaison exacte entre solides, la nature implicite de la représentation CSG limite les interactions entre un utilisateur et un logiciel de modélisation géométrique : on ne peut pas désigner, par exemple, une arête ou une face puisque ces dernières ne sont pas explicitement représentées au niveau de l'arbre CSG, etc.).

Brep : la représentation par les bords

Comme son nom l'indique, il s'agit d'une famille de modèles géométriques qui décrivent un

9. Les modèles CSG permettent de représenter un ensemble de points quelconques. Cette définition pose problème lorsque l'objet modélisé n'est pas un solide. C'est le cas, en particulier, des faces et des arêtes *pendantes*, obtenues par intersections de solides n'ayant qu'une face ou qu'une arête en commun. Ces dernières définissent des portions d'objets sans épaisseur et donc ne représentent pas un solide. Pour contourner ce problème, Requicha [Req80] a proposé de compléter les opérations booléennes classiques par l'opération de régularisation.

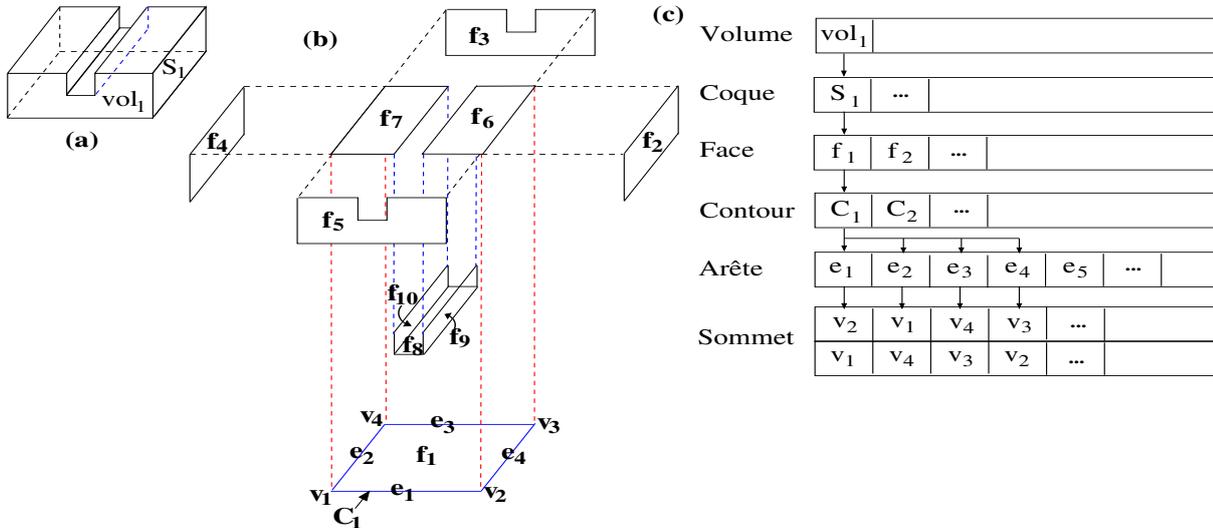


Figure 2.4: (a) Exemple d'une rainure sur un bloc avec (b) une vue éclatée partielle des entités topologiques qui le composent et (c) une représentation d'une partie de sa structure de données Brep.

solide par la donnée de son bord : la description de la surface fermée et orientable qui enveloppe l'intérieur de ce solide. Cette enveloppe est décrite hiérarchiquement, suivant les modèles, par un ensemble de coques (ensemble des faces connexes constituant le bord d'un volume), de faces, de contours (ensemble d'arêtes connexes constituant le bord d'une face), d'arêtes et de sommets (comme illustré sur la figure 2.4). Différentes structures de données ont été proposées et utilisées dans les systèmes de modélisation de solides : les arêtes ailées [Bau75], les demi-arêtes [CK88, Män88], etc.

En ce qui concerne l'intégrité des solides modélisés en Brep, la principale contrainte à vérifier est que les faces du solide forment une surface fermée, sans auto-intersection. En d'autres termes, chaque arête orientée appartient à exactement deux contours en $3D$, les sommets sont formés de cycles de faces adjacentes et le graphe des sommets est connexe.

Les modèles Brep contiennent la description explicite des solides modélisés. Contrairement à la modélisation CSG, la construction des modèles Brep est découplée des opérations de modélisation utilisées, qu'elles soient de haut niveau (opérations booléennes, génération par balayage, etc.) ou de bas niveau (opérations d'Euler [Bau75, BHS80, MS82]). Les modèles Brep ne peuvent donc pas être considérés comme des modèles paramétriques. Ils constituent par contre une représentation très puissante du modèle géométrique lui-même.

Autres modèles géométriques classiques

D'autres modèles géométriques, permettant la représentation de solides, ont été définis. Il s'agit des modèles de partitionnement spatial et du modèle d'instanciation de primitives. Dans le premier cas, le solide est subdivisé en un ensemble d'éléments, fini, fermé et sans intersection. La représentation la plus générale dans la famille des modèles de partitionnement spatial est la modélisation par *décomposition cellulaire*. Historiquement, le modèle par *énumération spatiale* a été le premier à être développé. Le principe est assez simple : approximer (approcher) le solide à modéliser en utilisant une grille fixe et régulière de cellules carrées en $2D$ (*pixels*) et cubiques

2.1. VERS UN MODÈLE PARAMÉTRIQUE

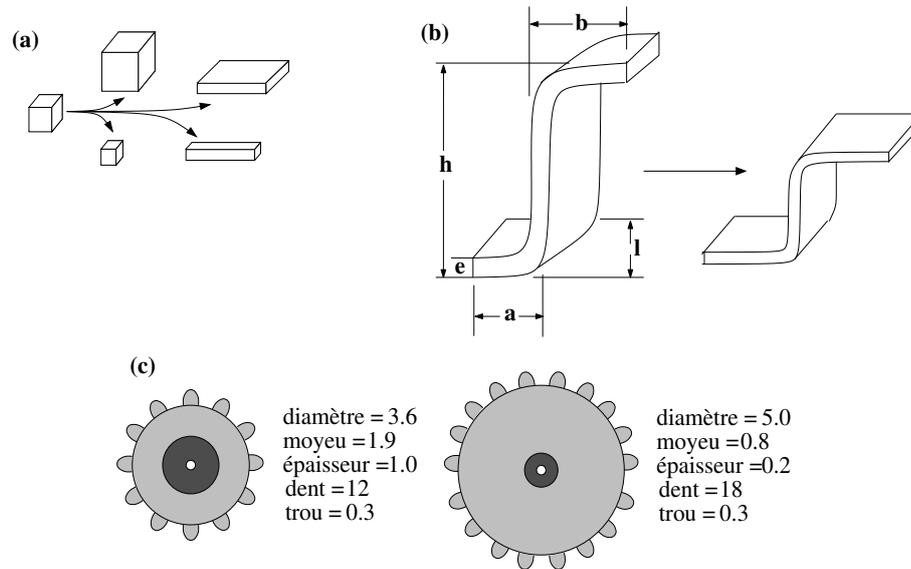


Figure 2.5: Exemples de primitives paramétrées. (a) Primitive cubique et ses variantes. (b) Primitive de section en Z définie par les paramètres a, b, h, l et e . (c) Deux engrenages issus de la même primitive. Figures extraites de [Mor97] et [FDF⁺95].

en 3D (*voxels*). Très vite, le problème causé par la saturation des capacités de stockage mémoire s'est posé. Pour y remédier, le modèle de représentation par arbre octal (*octree*) [JT80] est mis en œuvre. La construction de l'arbre s'effectue en décomposant le solide initial en octants. Lorsqu'un octant n'est que partiellement rempli, il est, à son tour, décomposé, de façon récursive, en octants de plus petite taille. Il est à noter que dans ce type de modèles, les paramètres utilisés correspondent aux positions et dimensions des différentes cellules élémentaires utilisées dans la représentation finale du solide. Ayant une portée locale, ces paramètres ne portent en eux aucune sémantique globale. Donc, cela nous permet de conclure que les modèles de partitionnement spatial ne sont pas paramétriques.

Dans le cas d'un modèle d'instanciation de primitives, l'approche est toute autre. Le système de modélisation définit, *a priori*, un certain nombre de formes 3D primitives qui ont un sens dans un domaine d'application particulier. Ces primitives comprennent non seulement des paramètres de dimensionnement mais aussi de fonctionnement. Elles peuvent être à géométrie simple (figure 2.5(a)) ou complexe (figures 2.5(b) et (c)), à topologie fixe (figures 2.5(a) et (b)) ou variable (figure 2.5(c)). Dès lors, nous considérons une primitive paramétrée comme une famille (ou une classe) de solides dont les membres (instances) sont engendrés suivant la variation de ces paramètres. Les systèmes d'instanciation de primitives sont alors, par nature, concis et dédiés. Ils permettent la construction d'une hiérarchie ou d'un assemblage d'instances de primitives et l'enrichissement du système (ajout de nouvelles formes) passe inévitablement par une édition du code.

Les primitives expriment une certaine sémantique et la description de l'assemblage précédemment cité est en général conservée. Ces deux éléments nous permettent de considérer ces systèmes de modélisation basés sur l'instanciation de primitives comme faisant partie de la famille des systèmes paramétriques.

2.1.2.2 Les modèles topologiques

Les modèles topologiques (ou modèles géométriques à base topologique¹⁰) permettent de représenter des objets subdivisés, c'est-à-dire partitionnés en cellules de différentes dimensions : sommets, arêtes, faces, volumes, etc. Ces modèles permettent de représenter la *structure* d'objets subdivisés. Plus précisément, les modèles topologiques représentent des cellules comme des objets *abstrait*s et l'assemblage de ces cellules comme des *relations* entre cellules. Par exemple, les graphes d'incidences [Wei86, Bri93] représentent les cellules par des nœuds, et les relations d'incidence entre les cellules par les arêtes du graphe. Il est donc impératif d'associer au modèle topologique un modèle de plongement afin d'obtenir un modèle géométrique. Le modèle de plongement permet, dès lors, de définir la forme des cellules et donc leur donner une représentation concrète [BD94].

De très nombreux modèles à base topologique (et opérations associées) ont été proposés, et ce dans différents domaines : algorithmique géométrique, traitement et analyse d'images, mécanique des solides, géosciences, etc. Ces modèles peuvent être répartis selon les critères suivant (classement extraits du livre [BP07] dans le chapitre « modèles topologiques ») :

1. *type de cellules* : dans ce contexte, deux sous-catégories de modèles permettent de représenter des assemblages de :
 - cellules *régulières* : simplexes (triangles, tétraèdres, etc.), simplexes cubiques (carrés, cubes, etc.), simplexes, qui sont des produits cartésiens de simplexes, etc. ;
 - cellules *quelconques* : le graphe d'incidence, les modèles ordonnés (cartes [BS85, Lie94], cartes généralisées [Lie94], etc.), les complexes simpliciaux abstraits [PBCF94, CFM⁺94, FMP03] permettent de manipuler une plus grande diversité de cellules comparé aux modèles précédents. Les cellules vérifient des propriétés topologiques assurées par la définition même du modèle.
2. *type d'assemblage* : on peut distinguer des modèles topologiques qui permettent, ou non, la représentation de cellules plusieurs fois incidentes entre elles (notion de *multi-incidence*). Par exemple, une boucle peut être représentée par une arête dont les deux sommets extrémités sont confondus en un seul. Deux classes d'objets émergent suivant ce critère :
 - les quasi-variétés (ou *manifold* en terminologie anglo-saxonne) : une quasi-variété cellulaire de dimension n peut être définie comme un ensemble de cellules de dimension n , collées entre elles le long de cellules de dimension $n - 1$ de leurs bords, de telle manière qu'une cellule de dimension $n - 1$ est incidente à, au plus, deux cellules de dimension n . Les cartes généralisées ainsi que les cartes combinatoires sont des exemples de quasi-variétés.
 - les assemblages complexes ou *non-manifold* : dans ce cas, la multi-incidence est prise en charge (en dimension 2, une arête peut être incidente à plus de deux faces et deux faces peuvent avoir un seul sommet en commun [Mar96]). Les chaînes de cartes [EL93,

10. Par cette terminologie, Françon [FB00] a souhaité faire la distinction entre ces modèles et les modèles Brep. La confusion provient du fait, qu'à l'origine, les modèles Brep ont été mis en œuvre puis développés pour la représentation et la manipulation de solides à travers la subdivision de la surface du bord. Mais avec l'essor de la modélisation 3D, où les objets modélisés peuvent être constitués de plusieurs volumes, la question suivante s'est posée : que cherche-t-on réellement à représenter ? Un assemblage de faces séparant des volumes, ou un assemblage de volumes ? Le premier cas correspond à la modélisation Brep et le second à la modélisation de subdivisions. Dans ce cadre, les modèles topologiques se placent clairement dans le second cas.

EL94] et les ensembles semi-simpliciaux [LL95, LL96] sont des modèles qui permettent la représentation d'assemblages complexes de cellules.

Le choix d'un modèle est dicté par un certain nombre d'impératifs, à savoir le type d'objets à modéliser (solides, quasi-variétés cellulaires orientées ou non, quasi-variétés simpliciales, etc.), les opérations qu'on souhaite appliquer sur les objets d'un modèle (par exemple, les opérations booléennes ne sont pas adaptées aux objets simpliciaux [LL96, FMP03]), le compromis entre l'espace utilisé par les structures de données propre au modèle, le temps imparti à l'application des opérations de modélisation, et la complexité logicielle à développer certaines opérations.

Nous avons fait le choix des cartes généralisées comme noyau géométrique du système paramétrique que nous allons utiliser dans le cadre de notre travail. Pour quelles raisons? Tout d'abord, nous souhaitons manipuler des quasi-variétés cellulaires sans restriction au niveau de l'orientabilité des objets modélisés. Ensuite, de par la définition homogène du modèle, les traitements effectués sur les cellules peuvent être intuitivement généralisables en toute dimension. Enfin, la distinction entre topologie et plongement permet une meilleure flexibilité dans le développement des opérations de modélisation. En effet, plusieurs opérations de modélisation portent soit sur la structure, soit sur la géométrie de l'objet. Dans ce cas, lorsqu'on modifie la topologie de l'objet, on ne se soucie guère de son plongement, et vice-versa¹¹. Aussi, la composition d'opérations de bas niveau (qui ont, en général, une incidence locale) permet de mettre en œuvre des opérations plus complexes ajoutant une sémantique particulière à ces dernières.

Les cartes généralisées (ou G-cartes)

Une carte généralisée est un modèle à base topologique représentant une quasi-variété cellulaire de dimension n quelconque. Intuitivement, une carte généralisée peut être obtenue en éclatant successivement les cellules composant l'objet qu'elle représente. Comme nous pouvons le voir sur la figure 2.6, l'objet $2D$ est constitué de deux faces F_1 et F_2 collées le long d'une arête g . En éclatant cette arête en deux (figure 2.6(b)), on obtient deux faces distinctes. Les deux nouvelles arêtes correspondent à l'arête g vue de F_1 et à l'arête g vue de F_2 et les deux faces résultant de l'éclatement sont, par la suite, liées par une relation de dimension 2 afin de mémoriser l'adjacence initiale de F_1 avec F_2 (figure 2.6(c)).

Lorsque le même procédé est appliqué sur les arêtes a, b, c, d, e et f , cela n'engendre pas de nouvelles arêtes, car elles font toutes partie du bord de cette subdivision : pour s'en souvenir, on définit, pour chaque arête, une relation de dimension 2 qui relie l'arête en question à elle-même.

Le bord de chaque face (F_1 ou F_2) est constitué d'arêtes collées via des sommets. On peut donc réappliquer le même procédé et éclater chaque sommet incident à deux arêtes en deux nouveaux sommets. Dans cette configuration, chaque relation, de dimension 2, associée à chaque arête est propagée aux nouveaux sommets constituant son bord (figure 2.6(d)). En plus, de la même façon que précédemment, on garde la trace que la paire de sommets issus de l'éclatement ne faisait à l'origine qu'un seul sommet, en liant ces derniers par une relation de dimension 1 (figure 2.6(e)).

Le bord de chaque arête est maintenant défini par deux sommets résultant des éclatements successifs (figure 2.6(f)). Entre ces sommets, on définit une relation de dimension 0 qui signifie que ces derniers composent les extrémités d'une même arête (figure 2.6(g)).

11. Pour le reste des opérations (les opérations qui portent sur la géométrie et la topologie en même temps), la distinction est aussi utile lors du débogage des programmes implémentant ces opérations.

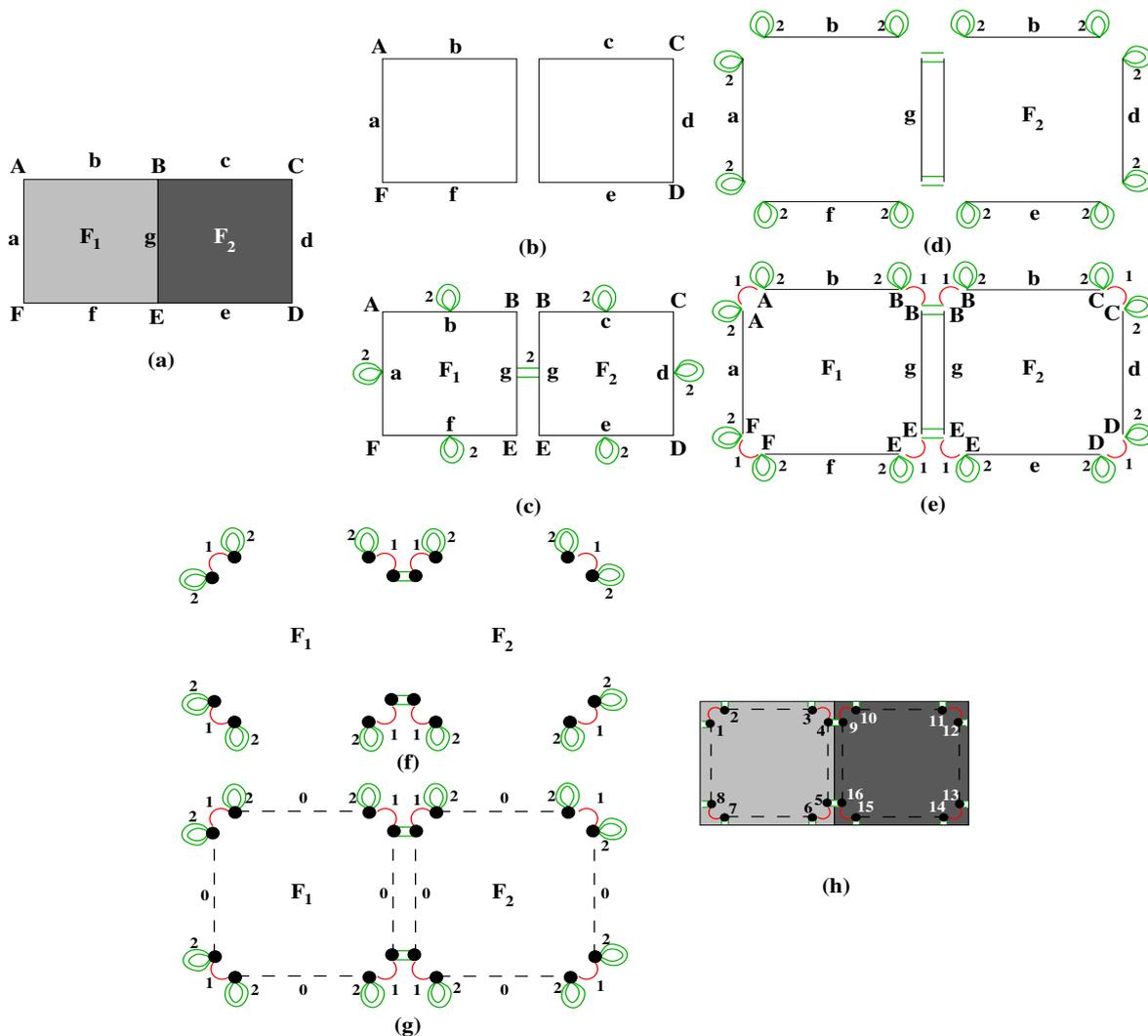


Figure 2.6: Genèse des G-cartes par processus de décomposition (application sur un objet $2D$). (a) Un objet $2D$. (b) Éclatement en faces. (c) Ajout des relations de dimension 2. (d) Éclatement en arêtes. (e) Ajout des relations de dimension 1. (f) Éclatement en sommets. (g) Ajout des relations de dimension 0. (h) 2-G-carte correspondante.

2.1. VERS UN MODÈLE PARAMÉTRIQUE

Les sommets résultants en fin de processus de décomposition représentent les éléments de base des G -cartes. Ils sont appelés *brins*. De plus, les relations de dimension 0, 1 et 2 qui appariement les brins par paires sont des applications. Ce qui conduit à la définition suivante extraite de [Lie94] :

Définition 2.1 (*Carte généralisée*)

Soit $n \geq 0$. Une carte généralisée de dimension n (ou n - G -carte) est un $n+2$ -uplet $(B, \alpha_0, \alpha_1, \dots, \alpha_n)$ tel que :

1. B est un ensemble de brins fini et non vide ;
2. $\forall i, 0 \leq i \leq n, \alpha_i$ est une involution sur B ;
3. $\forall i, j, 0 \leq i < i+2 \leq j \leq n, \alpha_j \circ \alpha_i$ est une involution sur B (condition de la quasi-variété) ;

Les cartes généralisées ont l'avantage d'être définies de façon homogène en toute dimension. Elles permettent de représenter la topologie des quasi-variétés orientables ou non, avec ou sans bord.

Dans un espace nD , une scène est décrite au moyen d'une subdivision cellulaire. Les cellules de toute dimension (sommets, arêtes, faces, volumes, hyper-volumes, etc.) sont définies en faisant appel à la notion d'orbite (figure 2.7).

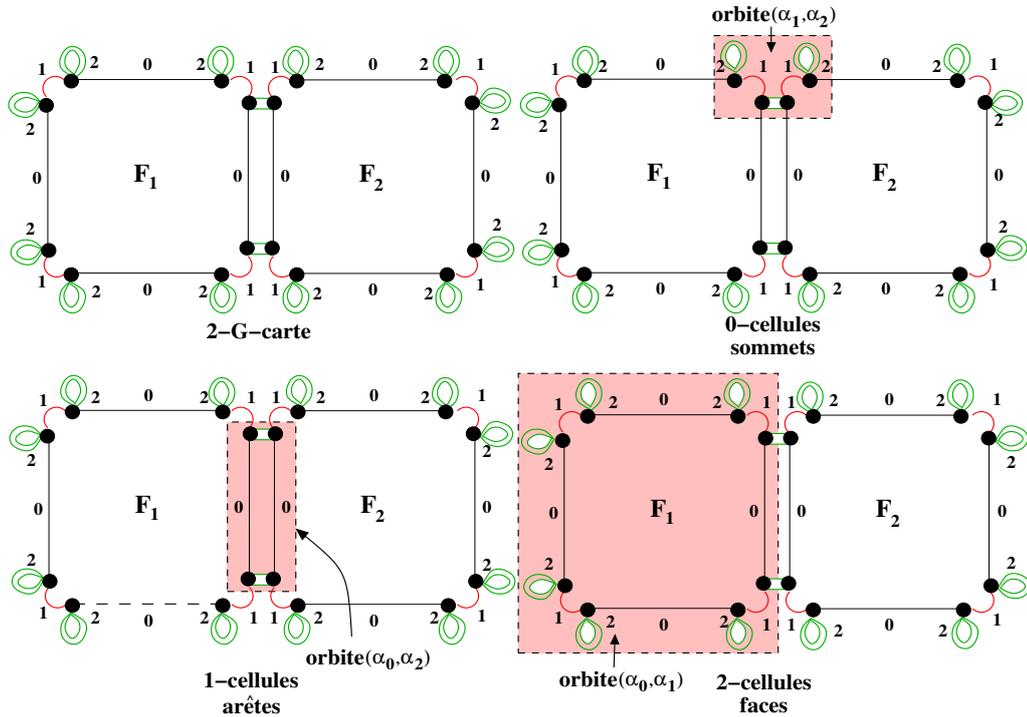


Figure 2.7: Cellules topologiques d'une 2- G -carte. Les brins sont symbolisés par des points et les involutions α_0 (resp. α_1, α_2) sont représentées par des traits (resp. des arcs de cercles en rouge et des traits doubles en vert). Pour obtenir les 0-cellules (resp. 1-cellules et 2-cellules), on regroupe les brins des orbites (α_1, α_2) (resp. (α_0, α_2) et (α_0, α_1)).

Définition 2.2 (Orbite)

Soient $G = (B, \alpha_0, \alpha_1, \dots, \alpha_n)$ une n - G -carte, b un brin quelconque de B et ϕ un ensemble de m permutations définies sur B . On appelle orbite de b par rapport à ϕ l'ensemble, noté $\langle \phi \rangle (b)$ et défini par :

$$\langle \phi \rangle (b) = \{b' \in B \mid 1 \leq k \leq m, \exists \phi_1 \in \phi, \dots, \exists \phi_k \in \phi, b' = \phi_k(\dots(\phi_1(b)))\}.$$

De façon informelle, une orbite $\langle \phi \rangle (b)$ correspond à l'ensemble des brins de B que l'on peut atteindre à partir de b par une composition quelconque des permutations de ϕ et de leurs inverses.

Définition 2.3 (Cellule de dimension i , ou i -cellule)

Soient $N = \{0, \dots, n\}, I = \{i_1, \dots, i_p\} \subseteq N$ et $b \in B$; on note $\langle \rangle_I (b)$ l'orbite de b relativement à l'ensemble des involutions d'indices compris dans I . Une cellule de dimension i (ou i -cellule), $0 \leq i \leq n$, de b correspond à l'orbite $\langle \rangle_{N-\{i\}} (b)$ ou plus simplement $\langle \rangle_i (b)$.

Une autre particularité des G -Cartes est de pouvoir couvrir seulement une partie de l'espace dans lequel elles sont définies ou la totalité de celui-ci. On parle alors respectivement de G -Cartes avec ou sans bord (voir la figure 2.8).

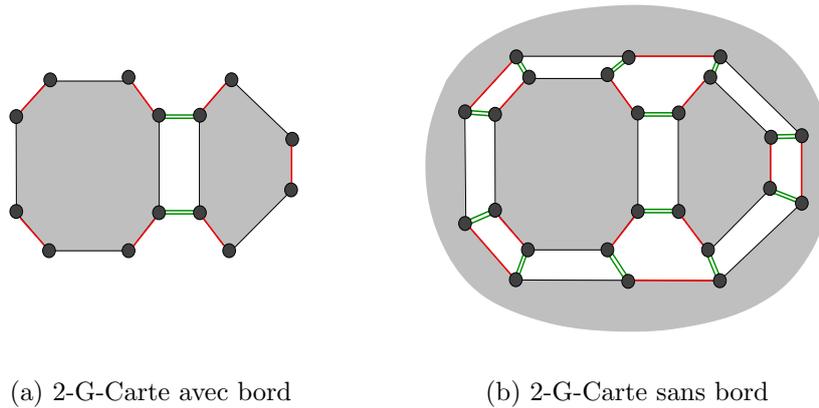


Figure 2.8: Exemple de G -Carte avec et sans bord.

Définition 2.4 (Carte généralisée sans bord)

Soit $G = (B, \alpha_0, \alpha_1, \dots, \alpha_n)$ une n - G -Carte. G est sans bord si et seulement si $\forall i \in [0, n]$ et $\forall b \in B, \alpha_i(b) \neq b$.

Le plongement des G -cartes est défini par des espaces supports. En effet, les cellules d'une carte généralisée peuvent être associées à des objets géométriques de dimension correspondante (point, morceau de courbe, morceau de surface, etc.). Par exemple, le plongement d'une cellule de dimension 2 (face) peut être défini par une surface support, découpée par le bord de la face; chaque arête du bord est plongée sur une courbe support contenue dans la surface support précédente, et délimitée par ses points extrémités. Bien sûr, cette représentation n'est pas la plus simple. En général, pour des raisons de simplicité, des points sont associés aux cellules

2.1. VERS UN MODÈLE PARAMÉTRIQUE

de dimension 0 (sommets) d'une G-carte, puis les cellules de dimension supérieure utilisent ces coordonnées pour avoir un plongement linéaire (les arêtes sont des morceaux de droites, les faces sont des morceaux de plans $2D$, ainsi de suite). Le cas des cellules non planaires (principalement les courbes et les surfaces) est traité dans le cadre des modèles que nous nous proposons d'introduire dans ce qui suit.

2.1.2.3 Les courbes et surfaces paramétriques

La recherche sur les courbes et les surfaces paramétriques a débuté il y a une quarantaine d'années avec les travaux de Bézier, Coons, Ferguson et De Casteljaou [Cas85, Béz86, Coo87]. La définition de tels éléments géométriques s'appuie essentiellement sur la donnée d'un ensemble de points qu'ils contribuent à façonner. Ces points s'appellent les *pôles* ou plus communément, les points de contrôle. Pour comprendre la genèse de cette définition, intéressons-nous à la notion de barycentre d'un ensemble de points. Le barycentre C de $(n + 1)$ points ou pôles P_i affectés des masses φ_i est défini par :

$$\left(\sum_{i=0}^n \varphi_i\right)C = \sum_{i=0}^n \varphi_i P_i$$

Cela signifie que la position du barycentre est contrôlable par les points P_i (d'où le terme points de contrôle). En supposant que les masses φ_i varient en fonction d'un paramètre t et que $\sum_{i=0}^n \varphi_i \equiv 1$, une courbe paramétrique est alors définie par :

$$C(t) = \sum_{i=0}^n \varphi_i(t) P_i$$

Le choix des φ_i , dans la dernière équation, permet de différencier les principales formes des courbes paramétriques : Bézier, *B-splines* [dB78], *B-splines* rationnelle (plus connue sous l'appellation NURBS¹²) [PT97] ou *X-splines* [Bar88]. La figure 2.9 illustre le cas des courbes de Bézier.

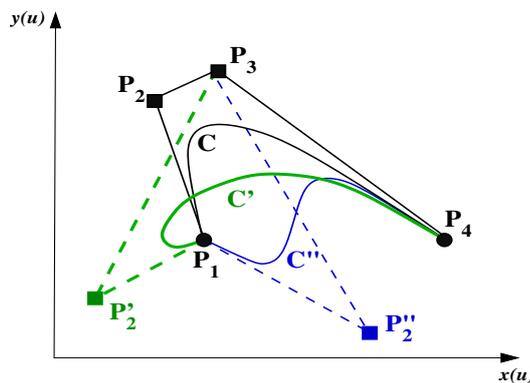


Figure 2.9: Famille de courbes de Bézier (C , C' et C'') et leurs points de contrôle. Seul le point P_2 varie pour chaque courbe. Son influence peut être assimilée à l'action d'un ressort entre le point de contrôle et la courbe.

12. NURBS : Non Uniform Rational B-Splines.

Les surfaces paramétriques se définissent, quant à elles, sous la forme d'un produit tensoriel. Cette définition se comprend à travers l'idée de paramétrer la déformation d'une courbe, comme le montrent les figures 2.10 et 2.11. Soient C_0 une courbe paramétrique définie suivant l'expression précédente et C_1 une autre courbe qui correspond à la déformation et au déplacement de C_0 . Le passage de C_0 à C_1 conduit à la définition des surfaces paramétriques les plus couramment utilisées. Pour ce faire, prenons un point $C_0(u)$ et son analogue $C_1(u)$. Le passage de la position de départ vers celle d'arrivée décrit une courbe. Cette courbe prend en paramètre la variable v et se nomme $S_u(v)$. Ainsi, chaque valeur du paramètre u décrit une courbe S_u . L'ensemble des points ainsi engendré lors de la déformation décrit une surface notée $S(u, v)$. Dans ce cadre, la surface $S(u, v)$ s'exprime comme suit :

$$S(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \varphi_i(u) \psi_j(v) P_{i,j}$$

où les paramètres u et v jouent des rôles symétriques.

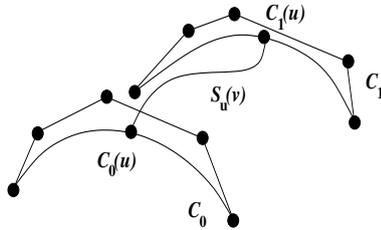


Figure 2.10: Déformation continue d'une courbe.

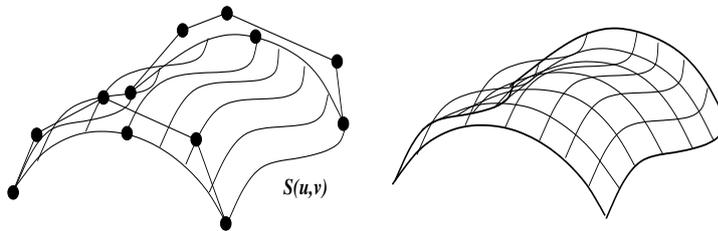


Figure 2.11: Réseaux de courbes définissant une surface.

L'intérêt de définir les courbes et les surfaces paramétriques provient à la fois de la capacité d'obtenir des formes quelconques¹³ (dont l'expression n'est pas totalement contrainte par le repère de travail, par exemple le repère cartésien) ainsi que la bonne maîtrise des formes engendrées et la capacité de modification au niveau des détails. Cette souplesse avérée a pour contrepartie la lourdeur de certains traitements (comme l'étude du positionnement d'un point par rapport à une surface ou l'intersection de deux surfaces).

Le qualificatif « paramétrique » associé à ces courbes et surfaces n'est pas le même que celui des systèmes paramétriques. Nous considérons que, ces entités sont plus « paramétrées » que paramétriques. En fait, une courbe ou une surface paramétrique n'a pour paramètres qu'un

13. En réalité, pas tout à fait quelconques car le caractère régulier des maillages de points de contrôles (*pôles*) limite la forme des courbes et des surfaces paramétriques.

ensemble de points de contrôle. Certes, ces derniers permettent de faire varier leur forme, mais ils ne peuvent exprimer une sémantique particulière car ils n'ont qu'une influence locale sur la forme de l'entité modélisée.

2.1.2.4 Les surfaces implicites

Les modèles de représentation des surfaces paramétriques et implicites sont les modèles surfaciques les plus utilisés. L'affichage de ce type de surfaces se trouve être accéléré par les cartes graphiques puisqu'elles sont converties, à cet effet, en un nombre paramétré de facettes planes. Malgré cela, les surfaces implicites se sont fait une place dans le domaine de la modélisation géométrique pour pallier à un certain nombre d'inconvénients, entre autres, ceux suscités par les surfaces paramétriques :

- Leur définition ne permet pas de définir naturellement si un point de l'espace est situé à l'intérieur ou à l'extérieur d'un objet (traitement indispensable pour détecter les collisions en animation ou pour le calcul d'intersections en lancer de rayons ou autre) ;
- Elles sont mal adaptées à la représentation d'objets qui se déforment, en particulier ceux qui changent de genre topologique au cours du temps (séparation en plusieurs composants, fusions, métamorphoses).

Une surface implicite est définie comme l'ensemble de points $\mathbf{p} = (x, y, z)$ de l'espace vérifiant une équation de type $f(\mathbf{p}) = c$, où f est une fonction de \mathbb{R}^3 dans \mathbb{R} et est appelée *fonction potentiel*. Cette fonction est continue, différentiable et modélise une surface finie et fermée séparant par la même l'*intérieur* de l'*extérieur* de l'objet modélisé.

Avec l'expression générale de f , il n'est pas facile de contrôler la forme d'une surface implicite en faisant varier les paramètres de l'équation qui la définit. Donc, au mieux, ces surfaces constituent les formes de base (primitives) des méthodes de modélisation constructives comme peut l'être le modèle CSG. Pour remédier à cela plusieurs types de surfaces implicites ont été définis :

- Surfaces implicites à squelettes ponctuels : c'est l'exemple de surface implicite le plus simple à définir et surtout à contrôler. Dans la littérature, on trouve ce type de surfaces sous la dénomination de *blobs*[WMW86], *Metaballs* ou *Soft Objects*. Le contrôle de la forme de la surface implicite s'effectue à travers un ensemble de points c_i appelés *squelettes*. Chaque squelette influe sur la surface à travers la fonction $f_i(\mathbf{p})$ définie par : $f_i = g_i \circ d_i$ où g_i est une fonction décroissante, continue et dérivable et d_i est la distance (le plus souvent Euclidienne) entre un point de la surface implicite \mathbf{p} et le squelette ponctuel c_i (c'est-à-dire, $d_i(\mathbf{p}, c_i) = \|\mathbf{p} - c_i\|$). Ainsi, le *blob* est défini par le mélange (la somme) de primitives engendrées par les squelettes ponctuels c_i (figure 2.12) telle que sa fonction potentiel est égale à :

$$f(\mathbf{p}) = \sum_{i=0}^{n-1} f_i(\mathbf{p}) \quad (2.1)$$

L'une des principales limites de ce type de surfaces se situe au niveau de l'influence qu'a chaque c_i sur f . En effet, cette dernière reste la même. Par conséquent, le champ des formes pouvant être modélisées est plus restreint.

- Surfaces implicites à squelettes complexes : dans ce type de surfaces implicites, les squelettes ponctuels sont remplacés par des squelettes quelconques. La modélisation devient ainsi plus souple et la gamme de surfaces implicites modélisées devient plus vaste. Par

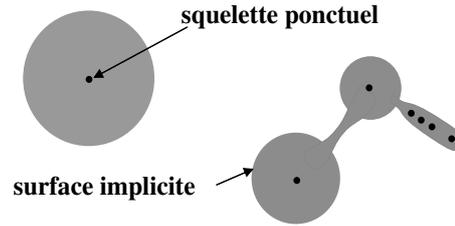


Figure 2.12: *Blob* construit par sommation des contributions de ses squelettes ponctuels.

exemple, l'utilisation de squelettes filaires (ensemble d'arêtes connexes) ou surfaciques simplifie la définition de surfaces implicites tubulaires ou localement planes.

Dans la littérature, il existe deux méthodes pour générer des surfaces implicites à partir de squelettes quelconques. La première méthode définit des surfaces implicites par la distance au squelette considéré [BW90]. Théoriquement, tout objet géométrique peut être utilisé comme un squelette S_i . Dès lors, il suffit juste de savoir calculer la distance $d(\mathbf{p}, S_i)$ entre un point \mathbf{p} de l'espace et le point le plus proche du squelette S_i . Ainsi, en se basant sur le principe précédent pour la définition de la fonction potentiel, celle définie dans l'équation (2.1) devient :

$$f(\mathbf{p}) = \sum_{S_i \in S} f_i(\mathbf{p}) \quad (2.2)$$

La seconde méthode définit les surfaces implicites par convolution. L'idée de base introduite par [BS91] consiste à tenir compte, sur chaque élément d'un squelette, de la contribution de tous les points de cet élément pour calculer le potentiel en un point \mathbf{p} de l'espace. Pour ce faire, une intégrale remplace la sommation dans l'équation (2.2) comme suit :

$$f(\mathbf{p}) = \int_{S_i} f_i(\mathbf{p}, u) du \quad (2.3)$$

- Surfaces implicites à potentiel échantillonné : à l'opposé de ce qui a été fait jusqu'à présent (c'est-à-dire, la définition de la fonction potentiel des surfaces implicites de façon analytique), une alternative consiste à définir le potentiel en tout point \mathbf{p} de l'espace en effectuant une interpolation tri linéaire (ou de plus haut degré) sur un nombre fini de valeurs échantillons. Ces valeurs représentent les valeurs de potentiel aux sommets d'une grille tridimensionnelle.

De telles représentations ont été introduites en modélisation dans le cadre de systèmes de sculpture interactive [GH91]. Toutefois, la simplicité de leurs mise en œuvre par de simples grilles tridimensionnelles ne couvre pas les inconvénients qui leurs sont propre. Aussi, pour pallier à cela, plusieurs travaux ont utilisé des structures de grille multi résolution pour stocker les échantillons des valeurs de potentiel [FCG00, PF01].

2.1.3 Modélisation par caractéristiques

Les modèles géométriques définis dans la section 2.1.2 sont en mesure de générer des objets et des scènes d'une très grande complexité. Cependant, ils ne permettent pas d'exprimer un haut niveau de sémantique. Évidemment, lorsqu'on effectue une scission de faces, on peut déduire que l'on souhaite décomposer une surface en plusieurs morceaux. En revanche, on ne pourra savoir

2.1. VERS UN MODÈLE PARAMÉTRIQUE

que cette opération est appliquée afin de réaliser une rainure sur un bloc (voir l'exemple de la figure 2.13(a)). Aux intentions de conception sont associées des stéréotypes de formes ayant une signification fonctionnelle comme les rainures, les poches, etc (figure 2.13(d)). Ainsi, les limites en termes de capacités d'expressions peuvent être repoussées en ayant recours aux caractéristiques. La figure 2.13 montre un exemple de modélisation par caractéristiques.

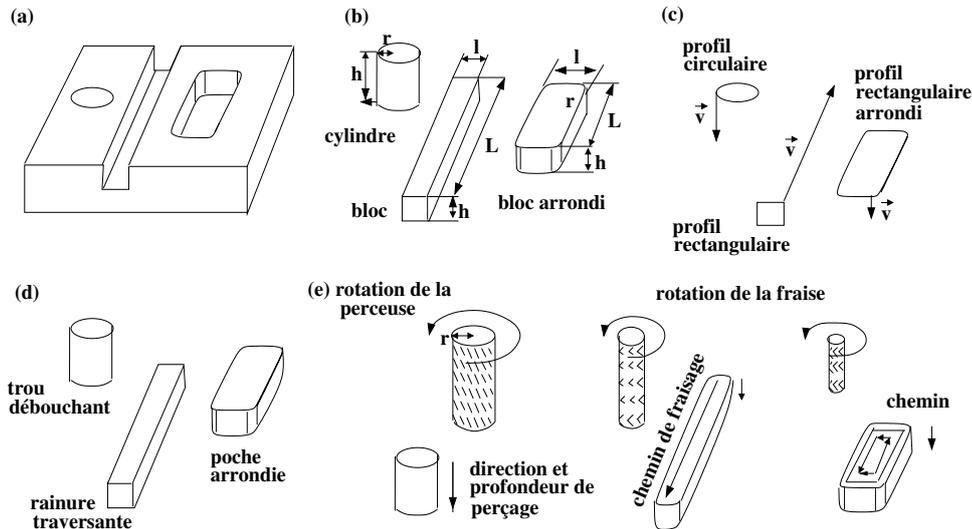


Figure 2.13: Exemple de définition de caractéristiques (adapté de [HR98]). (a) La pièce modélisée. (b) Une définition CSG (primitive élémentaire + dimensions). (c) Une définition Brep par extrusion (profil + vecteur ou chemin d'extrusion). (d) Une définition par caractéristiques de forme. (e) Une définition par caractéristiques d'usinage (outil + chemin + profondeur).

Cette modélisation par caractéristiques devient paramétrique lorsqu'elle conserve tout le processus de modélisation, et lorsqu'elle permet la réévaluation de la géométrie en modifiant les valeurs des paramètres associés à chaque caractéristique. C'est dans cette perspective que nous nous intéressons à la modélisation par caractéristiques et plus particulièrement à la modélisation par caractéristiques de forme.

2.1.3.1 Concept de caractéristique

Selon Shah et Mäntylä [SM95], une caractéristique (ou *feature*) peut être définie comme « une entité représentant les intentions de conception ou le sens de la géométrie d'une pièce ou d'un assemblage »¹⁴. La conception ou modélisation par caractéristique [SR88b, Ros90, Gar91, Rol91, RR92] est donc la création de modèles géométriques à travers un certain nombre de propriétés qui les caractérisent. Ainsi, il est possible de leur adjoindre des informations liées au matériau (résistance, ductilité¹⁵, etc.), à la métrologie (rugosité, dispersion, etc.), à la maintenance, etc. Le but affiché est de faciliter le travail exécuté pendant la conception et en aval de celle-ci. Il est également possible d'exprimer une caractéristique comme :

1. un constituant ou une propriété physique d'une pièce ;

14. « ... a feature represents the engineering meaning or significance of the geometry of a part or assembly. »

15. Ductilité : propriété de certains matériaux (essentiellement des métaux) de pouvoir être étirés et allongés sans se rompre.

2. une forme géométrique générique ;
3. un moyen pour véhiculer une signification en ingénierie ;
4. un élément ayant des propriétés prévisibles [SSR⁺88].

Il existe plusieurs types de caractéristiques que nous énumérons de façon non exhaustive :

- les caractéristiques de forme : elles décrivent des formes récurrentes ou stéréotypées, souvent propres à un domaine d'application donné. Elles permettent de représenter l'intention du concepteur ou la sémantique d'une géométrie associée à une partie d'un assemblage. Les caractéristiques de forme peuvent être considérées comme des parties de construction permettant la définition d'un produit ou la création d'un raisonnement géométrique [SM95]. Cette catégorie de caractéristiques se décompose en sous-catégories. On peut citer, en exemple, les caractéristiques de fabrication qui regroupent des informations sur les procédés de fabrication (usinage, moulage, soudure, etc.), sur les outils (type, chemin d'accès, etc.), etc ;
- les caractéristiques de tolérance : elles capturent la variation admissible de la forme, la taille, la position initiale des objets modélisés ;
- les caractéristiques d'assemblage : elles définissent des groupements de plusieurs types de caractéristiques telles que les conditions de liaison, la position et l'orientation d'une pièce par rapport à une autre, les contraintes cinématiques, etc ;
- les caractéristiques fonctionnelles : elles correspondent à l'ensemble des caractéristiques relatives à des fonctions spécifiques pouvant contenir des intentions de conception, des attributs non géométriques, des paramètres de performance, etc ;
- les caractéristiques de matériau : elles recensent la composition des matériaux, leurs traitements, etc.

Dans le cadre de notre travail, nous nous intéressons plus particulièrement aux caractéristiques de forme.

2.1.3.2 Techniques de création

Un modèle à base de caractéristiques pilote la construction d'objets à travers un ensemble d'opérations de haut niveau (rainure, trou, bosse, etc.). Le résultat est une géométrie représentée par les modèles géométriques et qui se traduit en termes de topologie par des volumes, des faces, des arêtes, etc. Par conséquent, un modèle à base de caractéristiques peut être représenté par une structure à deux composantes : le modèle géométrique (ou encore le noyau géométrique par analogie avec la première composante du système paramétrique défini dans la section 2.1.1) et le modèle des caractéristiques contenant les définitions de la forme, les attributs de dimension et de position, les contraintes géométriques, les attributs non géométriques (par analogie avec les systèmes paramétriques, cette seconde composante correspond à la couche de spécification paramétrique, avec la sauvegarde de l'historique de modélisation en moins), etc. Selon que l'on déduit un modèle de l'autre, [SM95, GM93] classent la conception à base de caractéristiques en deux catégories :

1. la *reconnaissance de caractéristiques* : des informations (caractéristiques) sont calculées à partir d'un modèle géométrique et lui sont associées par la suite. Les méthodes de calculs peuvent être interactives ou automatiques ;
2. la *conception par caractéristiques* : la géométrie est créée à partir de la donnée de caractéristiques.

2.1. VERS UN MODÈLE PARAMÉTRIQUE

La figure 2.14 donne un aperçu des différentes techniques de création de caractéristiques.

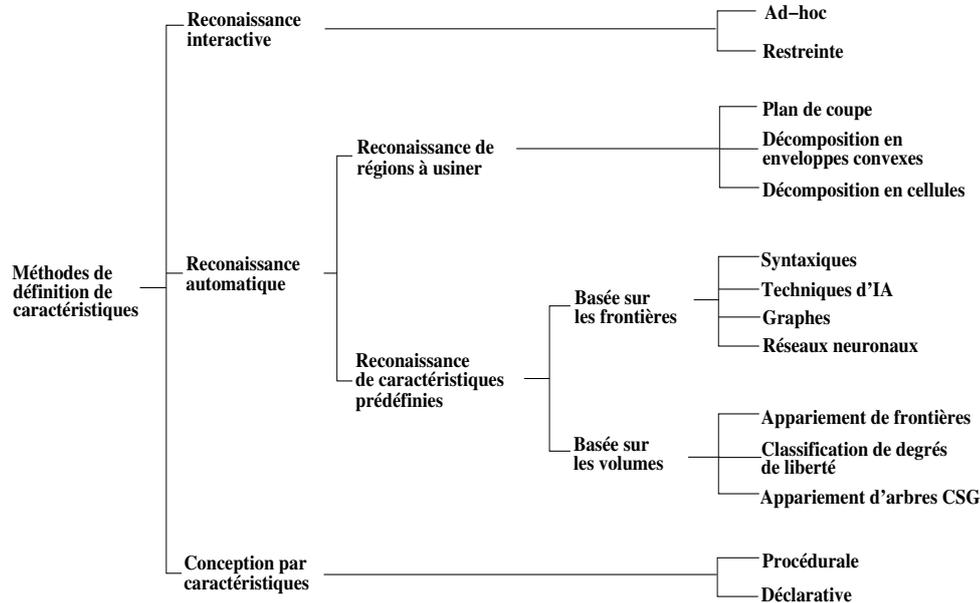


Figure 2.14: Les techniques de création de caractéristiques (extrait de [SM95]).

Les techniques de création de caractéristiques sont diverses et nombreuses [SHk93, AM95, Reg95, SM95, HR98]. Nous nous intéressons principalement aux techniques de conception par caractéristiques. Néanmoins, dans un objectif de complétude, nous allons, dans la partie qui suit, exposer brièvement le principe des techniques de reconnaissance de caractéristiques. Pour plus de détails, nous recommandons la lecture des documents [Flo87, LF87, JC88, WK93, SSS94, TK94, PRS94, SW97].

Reconnaissance de caractéristiques

Historiquement, le souhait exprimé par les industriels était d'associer des informations relatant le cycle de vie d'un produit durant sa fabrication aux éléments du modèle géométrique qui le représente.

Ainsi sont nées les techniques de reconnaissance de caractéristiques qui transforment un modèle géométrique (issu des bureaux de conception) en un modèle à base de caractéristiques (principalement les caractéristiques d'usinage) et assurent, par la même occasion, l'intégration de la conception à la production. Initialement *interactive* (ajout de caractéristiques après l'expertise d'un utilisateur humain), la reconnaissance de caractéristiques est devenue, dans les années 80, plus ou moins *automatique*.

Conception par caractéristiques

Avec cette technique, le modèle géométrique est spécifié au moyens de caractéristiques diverses et variées. Pour cela, le système de modélisation utilise une bibliothèque de caractéristiques prédéfinies. Dès lors, il suffit d'instancier, au besoin, des éléments de cette bibliothèque par une

spécification de contrainte, de paramètres de dimension et de position, etc. Le schéma de la figure 2.15 montre cette architecture.

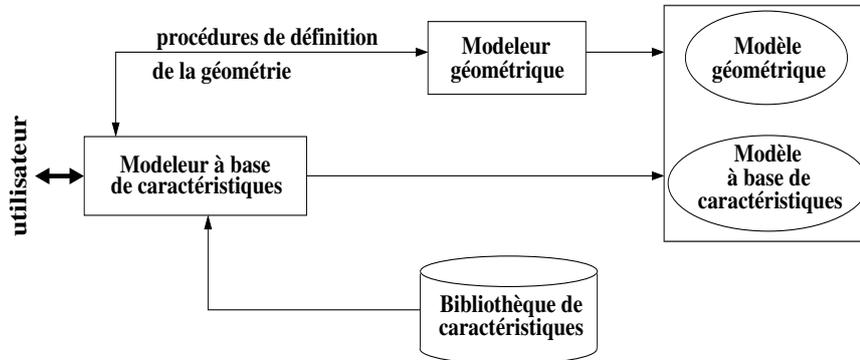


Figure 2.15: Conception par caractéristiques (extrait de [SM95]).

Les premiers systèmes à base de caractéristiques ont été mis en œuvre pour répondre aux besoins d'usinage exprimés par l'industrie (voir l'exemple de la figure 2.16). Les objets étaient créés en soustrayant d'un brut initial, les caractéristiques qui correspondent à l'enlèvement de matière par opération d'usinage, d'où la nature dite « destructive » de ce type de système. De même, il existe plusieurs prototypes développés dans cette perspective. On peut citer :

- le prototype élaboré par Arbab durant sa thèse [Arb82] ;
- le système NEXT-cut défini par Cutkosky et ces collègues [CT92] ;
- Le système QTC (*Quick Turnaround Cell*) de l'université de Purdue [CAM88] ;
- le système HutCAPP [MOP89], etc.

parallèlement, d'autres besoins ont été exprimés et de nouveaux systèmes par caractéristiques sont apparus. Ces derniers ont adopté une approche moins dépendante de la fabrication. En d'autres termes, les caractéristiques peuvent être aussi bien additives que soustractives. Ils portent une sémantique qui va au-delà des pratiques utilisées dans la fabrication d'outils industriels. Les systèmes qui entrent dans cette catégorie sont, entre autres, issus des travaux universitaires [Min85, LDS86], mais aussi du monde industriel tels que Pro/ENGINEER, CATIA, TopSolid, etc.

Comme pour les systèmes paramétriques, les contraintes utilisées dans les systèmes à base de caractéristiques sont orientées ou non. Dans le premiers cas, on parle de systèmes par caractéristiques avec une approche *fonctionnelle*, alors que dans le second, on parle plutôt de systèmes par caractéristiques avec une approche *équationnelle*.

2.1.3.3 Limites

Les techniques susmentionnées montrent la diversité des approches existant dans le domaine de la modélisation par caractéristiques. Chaque approche répond à un objectif spécifique et présente des avantages et des inconvénients propres. Ainsi, la reconnaissance des caractéristiques est difficilement généralisable aux cas réels. La conception par caractéristiques d'usinage est conçue pour être directement exploitable lors de la fabrication, mais n'est pas nécessairement naturelle pour un concepteur. Dès lors, l'utilisation de caractéristiques génériques est devenue l'approche qui se généralise dans tous les modeleurs.

2.1. VERS UN MODÈLE PARAMÉTRIQUE

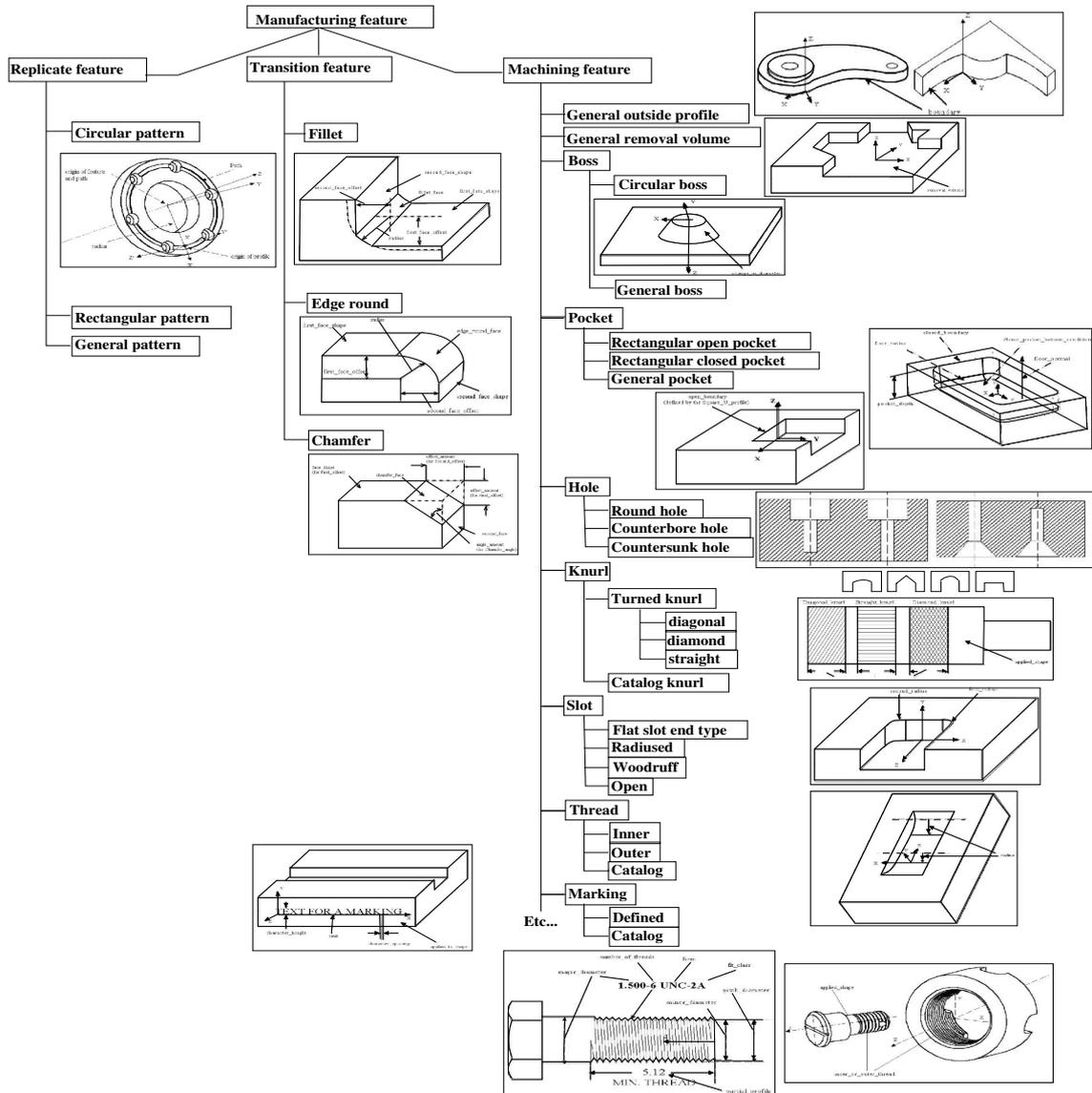


Figure 2.16: Vue partielle des caractéristiques de fabrication définies dans le fascicule 224 de la norme STEP.

2.2 Problème de la nomination persistante

De part leurs structures, les systèmes paramétriques connaissent bon nombre de problèmes [BB00] : des difficultés dans l'échange de spécifications (historiques de modélisation) entre deux systèmes paramétriques différents [SR04, MH05], une réduction de la flexibilité des systèmes paramétriques due aux dépendances fonctionnelles entre opérations de modélisation et le problème de la *nomination persistante* [MP02] (ou *persistent naming* en anglais).

Dans le cadre de cette thèse, nous nous intéressons au problème de la nomination persistante. Plus précisément, la difficulté qu'a un système paramétrique à *identifier* les entités géométriques et topologiques référencées par la spécification paramétrique, à maintenir ces références valides lors de la réévaluation et être assez « souple » pour permettre à l'utilisateur d'exprimer ses intentions de conception. Identifier les entités consiste à leurs attribuer un *nom* lors de leurs conception afin d'une part, de pouvoir les utiliser comme paramètres d'autres opérations de modélisation durant la phase de modélisation initiale, et, d'autre part de pouvoir les retrouver dans la phase de réévaluation¹⁶. Il s'agit de trouver une correspondance (un *appariement* ou *matching* en terminologie anglophone) entre les entités du modèle initial et celles du modèle réévalué. Pour illustrer ce problème, prenons l'exemple de la figure 2.17.

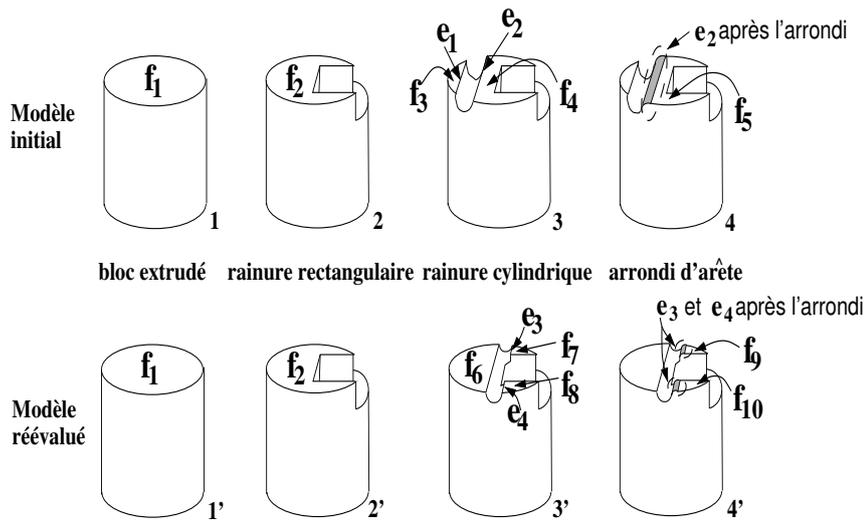


Figure 2.17: Problèmes d'attribution de noms et d'appariement.

Dans cet exemple, le modèle initial est conçu au moyen d'une spécification paramétrique comprenant quatre opérations de modélisation successives. La dernière consiste à arrondir l'arête e_2 résultant de l'intersection de la face du haut du bloc extrudé avec la face latérale de la rainure cylindrique. Dès lors, un des paramètres de l'opération d'arrondi, contenu dans la spécification paramétrique, n'est rien d'autre que la référence à l'arête e_2 . Après la quatrième étape, l'instance courante de la géométrie ne contient plus l'arête e_2 : elle est supprimée. Aussi, comment

16. La réévaluation que nous considérons consiste à rejouer le processus de modélisation (puisque les opérations sont sauvegardées au niveau de la spécification paramétrique) lorsqu'un certain nombre de paramètres des opérations de modélisation sont modifiés. Il ne s'agit en aucun cas de la modification de la spécification paramétrique (changement de l'ordre des opérations ou le remplacement des opérations par d'autres) car dans ce cas on parle plutôt d'*édition* du modèle paramétrique.

stocker dans la spécification paramétrique une référence à l'arête e_2 pour l'opération d'arrondi, cette arête n'existant plus? Supposons maintenant que nous nous placions dans un contexte de réévaluation. À l'étape 3', l'arête initiale e_2 est cette fois scindée en deux arêtes e_3 et e_4 . À l'étape 4', le système ne sait plus quelle(s) est (sont) l'arête (les arêtes) qui doit (doivent) être arrondie(s). Le problème est donc d'identifier toutes ces arêtes, puis apparier l'arête e_2 avec les arêtes e_3 et e_4 (pour appliquer sur chacune l'opération d'arrondi) en dépit des changements topologiques. Le mécanisme de nomination doit être assez puissant pour effectuer un appariement robuste lors de la réévaluation.

Dans les sections suivantes, nous montrerons, la nécessité de disposer d'une part d'un nom muni d'une structure suffisamment riche pour caractériser les entités topologiques de façon unique et non ambiguë; et, d'autre part, d'un mécanisme d'appariement suffisamment robuste pour mettre en correspondance les entités du modèle initial et celles du modèle réévalué (ces entités pouvant ne pas avoir la même structure topologique). Nous verrons aussi que la résolution de ces problèmes induit directement la capacité d'un système paramétrique à exprimer et capturer les intentions de conception.

2.2.1 Nécessité d'un nom (1^{er} problème)

Comme nous avons pu le constater dans l'exemple précédent, le premier impératif de la nomination, dans tout système paramétrique, est la nécessité d'identification des entités. En effet, les entités topologiques doivent être caractérisées de façon unique afin qu'elles puissent être désignées de façon non ambiguë à partir de la spécification paramétrique. Intuitivement, la première idée qui vient à l'esprit est d'utiliser un numéro, un pointeur ou même le nom de la fonction utilisée pour générer le nom de ces entités. Cependant, toutes ces possibilités ne sont guère robustes. Dans les deux premiers cas, il suffit que l'entité considérée disparaisse au cours du processus de modélisation pour que ce lien devienne obsolète. Alors que dans le troisième cas, la structure topologique incidente à l'entité considérée peut changer et donc la fonction utilisée ne peut plus être appliquée. Cela est d'autant plus vrai qu'un pointeur devient invalide lorsque l'entité qu'il désigne disparaît comme c'est le cas de l'arête e_2 à l'étape 4 de la figure 2.17.

De plus, comme évoqué dans la section 2.1.2, une entité, dans un modèle topologique ou Brep, possède un voisinage topologique et le modèle qui la représente organise de façon directe (les graphes d'incidences) ou implicite (les cartes généralisées) la topologie d'un objet en subdivisant son bord : volume > coque > face > agrégat d'arêtes > arête > sommet. Chaque opération de modélisation génère un nombre non prédictible d'entités (conséquence intrinsèque de la définition des modèles topologiques et Brep). Étant donné ce caractère aléatoire et instable de la modélisation, il faut que le nom de chaque entité contienne des informations suffisamment stables pour permettre la résolution du problème de la nomination persistante. Comme nous allons le voir, ces informations proviennent principalement des propriétés géométriques et topologiques des objets modélisés. De plus, le nom d'une entité doit avoir une structure, à même de représenter ces informations. Par exemple, dans la figure 2.17, une nomination aussi triviale qu'une numérotation des arêtes e_2 , e_3 et e_4 ne peut être envisagée. En effet, les numéros attribués à ces dernières ne permettent pas de savoir que l'arête e_2 a été scindée en deux arêtes e_3 et e_4 .

2.2.2 Nécessité d'un appariement entre noms (2^{ème} problème)

Lorsque les paramètres des opérations de modélisation sont modifiés, la structure topologique des objets modélisés peut légèrement ou fortement varier. Il est alors nécessaire de déterminer dans quelle mesure un modèle initial, à une étape particulière, ressemble à un autre réévalué à la même étape. En d'autres termes, il est nécessaire de savoir si une entité du modèle initial correspond à une, plusieurs ou aucune entité du modèle réévalué.

Dans la figure 2.17, et plus particulièrement à l'étape 3', la modification de la position de la rainure cylindrique engendre un changement au niveau de la topologie. Cette modification pose, comme on l'a vu dans la section 2.2, un problème au niveau de l'étape 4'. Pour le résoudre, il faut savoir que l'arête e_2 du modèle initial correspond aux arêtes e_3 et e_4 dans le modèle réévalué.

Ce résultat particulier ne peut être donné que par un mécanisme général d'appariement qui prend en compte toutes les cas de correspondance possible. Dans l'exemple précédant, les cas en question sont : (1) $e_2 \leftrightarrow e_3$ et e_4 ; (2) $e_2 \leftrightarrow e_3$; (3) $e_2 \leftrightarrow e_4$; ou enfin (4) $e_2 \leftrightarrow \emptyset$. Ce mécanisme repose sur le processus de caractérisation pour établir les correspondances recherchées.

2.2.3 Capture des intentions de conception (3^{ème} problème)

Le jeu automatique d'un processus de modélisation dans un système paramétrique répond à un besoin certain, exprimé par l'utilisateur. Pour mener à bien ce travail, l'idéal consiste à capturer l'intention de l'utilisateur pour que le système sache ce qu'il a à faire au moment de la réévaluation. Ces intentions peuvent être simples (par exemple l'arrondi d'une arête) mais aussi complexes. Pour illustrer le deuxième cas, prenons l'exemple de la figure 2.18. Dans cet exemple, la spécification paramétrique contient trois opérations de modélisation : construction d'un bloc par extrusion de la face f , ajout d'une protrusion cylindrique et arrondi de l'arête située entre une coque et la protrusion. En fonction de l'arête à arrondir (celle qui se situe entre la face $f_{1.1}$ et la protrusion, ou bien celle qui se situe entre la coque latérale du bloc (montrée en gris sur la figure 2.18) et la protrusion), le modèle réévalué est différent. Dans le premier cas, on obtient le modèle de la figure 2.18(b) où l'arrondi de l'arête disparaît car il n'y a plus d'intersection entre la face $f_{1.1}$ et la protrusion. Dans le second cas, on obtient le modèle de la figure 2.18(c) où l'arrondi de l'arête est appliqué. La raison est que l'intersection entre la coque latérale du bloc et la protrusion subsiste toujours même lorsque cette protrusion est déplacée sur la face $f_{1.2}$.

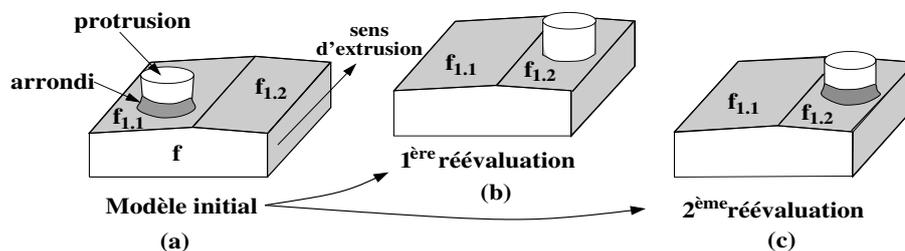


Figure 2.18: Différence au niveau des intentions de conception. (a) Le modèle initial. (b) le modèle réévalué sans arrondi. (c) Le modèle réévalué avec arrondi. La coque latérale du bloc extrudé est constituée de l'ensemble des faces représentées en gris clair sur la figure.

Pour capturer ce type d'intention, les solutions apportées pour résoudre le problème de la nomination persistante, doivent gérer la caractérisation et l'appariement d'entités de plus haute

dimension telles que les coques (agrégats de faces) et les volumes.

2.3 Solutions proposées

Bien que la modélisation paramétrique se soit développée et étendue pendant plus de deux décennies, aussi bien dans le monde industriel que dans le monde de la recherche scientifique, peu de publications ont été proposées pour la résolution du problème de la nomination persistante. Un travail précurseur et important dans ce domaine est celui de Hoffmann et Juan [HJ92]. À la même période, plusieurs auteurs ont étudié la structure interne du modèle de données paramétriques. Leurs contributions ont permis l'apparition de différentes représentations [SV95, PAAB⁺96] et de mettre en exergue leurs structures mathématiques sous-jacentes [RS98, WN05]. Aussi, certains d'entre eux 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].

Nous ne souhaitons pas faire une simple énumération des différentes contributions mais les étudier puis les analyser au travers d'un certain nombre de critères afin d'identifier les problèmes qui restent ouverts. Ainsi, dans les sous-sections suivantes, nous allons passer en revue les concepts communs aux différentes contributions proposées dans le cadre du problème de la nomination persistante, les points distinctifs à certaines solutions et qui ont permis d'étendre le champ des objets étudiés et enfin une analyse de ces solutions pour mettre en exergue les problèmes qui restent ouverts.

2.3.1 Concepts communs

Au vu des travaux publiés et malgré le fait qu'ils ne sont pas tous explicitement indiqués, il existe un certain nombre de concepts, présents dans la majorité des solutions proposées pour la résolution du problème de la nomination persistante.

2.3.1.1 Distinction entre les entités invariantes et les entités contingentes

Le premier concept fondamental que l'on peut trouver dans la plupart des solutions existantes pour la résolution du problème de la nomination persistante est la distinction entre les entités invariantes et contingentes. Une définition précise de cette notion se trouve dans [AMP99].

Définition 2.5 (*entité invariante*)

Une entité invariante est une entité géométrique et topologique qui peut être, complètement et sans ambiguïté, caractérisée par la structure d'une opération de modélisation et ses paramètres d'entrée, indépendamment des valeurs impliquées.

Dans la figure 2.17, les entités invariantes comprennent la face de début et de fin ainsi que la face latérale du bloc cylindrique extrudé, les coques latérales des rainures rectangulaire et cylindrique, la face résultant de l'opération d'arrondi (en gris sur la figure), etc. Pour caractériser, c'est-à-dire nommer de telles entités, il est nécessaire de définir un mécanisme de nomination qui associe ces entités aux opérations de modélisation et à leurs paramètres d'entrée.

Définition 2.6 (entité contingente)

Une entité contingente est une entité topologique ou géométrique qui résulte de l'interaction entre le modèle géométrique préexistant et les entités invariantes issues de la dernière opération de modélisation.

Par exemple, dans la figure 2.17, le nombre de faces du haut du cylindre extrudé à l'étape 3 diffère de celle à l'étape 3'. Ce nombre dépend de la position de la rainure cylindrique qui intervient aux étapes 2 et 2'. Il se trouve que la position de la rainure ne peut être connue à l'avance d'où le caractère imprévisible des entités contingentes. Donc, un mécanisme de nomination est d'autant plus nécessaire pour ce type d'entités.

2.3.1.2 Distinction entre les entités avec et sans source

Le deuxième concept commun à la plupart des mécanismes de nomination est la distinction entre les entités avec et sans source.

Définition 2.7 (entité avec source)

Une entité avec source est une entité topologique résultant d'une autre entité topologique de même dimension, et ayant une partie du plongement en commun avec cette dernière, suite à l'application d'une opération de modélisation.

Cette notion est largement utilisée dans les mécanismes de nomination proposés dans la bibliographie mais n'est définie de façon explicite que dans le travail de Wu [WZZZ01]. En effet, Wu énonce qu'une entité topologique E , résultant d'une opération booléenne entre deux objets O_1 et O_2 , possède une source s'il existe une autre entité topologique E' appartenant à O_1 ou O_2 tel que le plongement (la géométrie) de E est une partie du plongement de E' . En d'autres termes, si l'on note ens_1 et ens_2 les deux ensembles de points constituant les plongements respectifs de E et E' alors $ens_1 \subseteq ens_2$. E est alors une entité avec source et E' est la source de E . Par exemple, dans la figure 2.17, les faces f_3, f_4 et f_5 ont les sources respectives suivantes : f_2, f_2 et f_4 . Mais les arêtes e_1, e_2, e_3 et e_4 sont des entités sans source. D'une part, les arêtes, au même titre que les sommets, peuvent être dépourvus de source comme nous venons de le voir. D'autre part, les faces en 3D ne peuvent être que le résultat de la création, la scission ou la fusion d'autres faces. C'est pourquoi, on ne peut trouver de faces sans sources alors que pour les arêtes et les sommets ce n'est pas le cas. Pour cette raison, les systèmes paramétriques actuels et passés ont développé des mécanismes de nomination sur la base des entités faces qui semblent plus stables.

2.3.1.3 Nomination des entités invariantes issues d'extrusion ou de révolution

Le troisième concept commun porte sur les mécanismes de nomination utilisés pour caractériser les entités invariantes résultant des opérations de modélisation qui impliquent des structures topologiques 2D comme paramètres d'entrée. Par exemple, un profil et une trajectoire dans le cas d'un objet extrudé. Les mécanismes de nomination supposent que chaque élément de la structure topologique 2D soit, au préalable, nommé de façon robuste (c'est-à-dire qui ne reste pas tributaire de la forte variation de la topologie). Ensuite, ces mécanismes nomment chaque entité invariante en faisant intervenir le(s) nom(s) des entité(s) à partir de laquelle (desquelles) celle-ci résulte. Les travaux de [Che95, WZZZ01, AMPT03] élaborent des

2.3. SOLUTIONS PROPOSÉES

variantes pour implémenter ce concept mais le principe reste le même. Pour un profil extrudé, chaque entité topologique (sommet, arête et face) appartenant au résultat final de l'opération d'extrusion est nommée en fonction (1) des noms des entités composant le profil, (2) celles constituant la trajectoire sans oublier (3) l'identifiant de l'opération d'extrusion (en général, cet identifiant est le numéro de l'extrusion dans la liste des opérations de modélisation constituant la spécification paramétrique.) Dans ce contexte, la face latérale de l'objet extrudé, correspond au produit cartésien d'une arête du profil avec une arête de la trajectoire. Donc, la face ainsi considérée est caractérisée par le numéro de l'extrusion et le nom de ces deux arêtes.

Pour illustrer ces propos, étudions, par exemple, la contribution de Wu [WZZZ01]. Dans son travail, Wu propose une caractérisation de chaque entité extrudée (ou ayant subi une révolution) par le quintuplé suivant :

$$(ID, ID_p, ID_{\text{élément}}, ID_{\text{chemin}}, ID_{\text{trajectoire}})$$

où ID est l'identifiant de l'opération d'extrusion et/ou de révolution, ID_p est l'identifiant du profil $2D$ à extruder le long du chemin identifié par ID_{chemin} et $ID_{\text{élément}}$ puis $ID_{\text{trajectoire}}$ sont respectivement les identifiants de deux arêtes faisant partie du profil et du chemin précédemment considérés. Dans le cas où il s'agit de l'extrusion d'un profil le long de la normale à ce profil, les identifiants ID_{chemin} et $ID_{\text{trajectoire}}$ sont initialisés à 0. Pour la figure 2.19, si l'on considère l'identifiant de l'opération d'extrusion égal à 1, l'identifiant de chaque arête du profil et du chemin ainsi que l'identifiant du profil P à extruder le long du chemin C sont contenus dans le tableau de la figure 2.19, alors le nom de la face $F_{e_1 e'_3}$ est donné par le quintuplé $(1, 4, 3, 3, 1)$: où, 4 et 3 sont respectivement les identifiants du profil P et du chemin C utilisés et 3 et 1 correspondent respectivement aux identifiants des arêtes e'_3 du profil et e_1 du chemin.

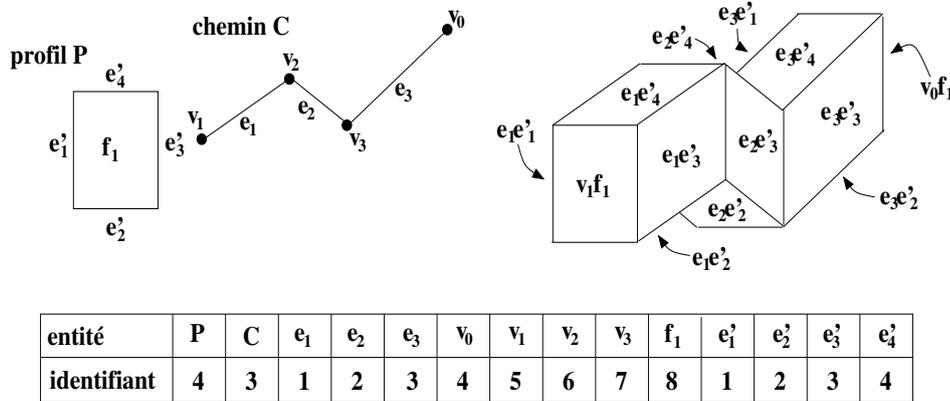


Figure 2.19: Nomination des entités invariantes issues de l'opération d'extrusion.

2.3.1.4 Nomination basée sur les entités invariantes

La difficulté, dans un système paramétrique, se concentre sur la nomination et l'appariement d'entités contingentes. Le caractère imprévisible et instable de ce genre d'entités en est la principale raison. Pour pallier à cela, toutes les solutions proposées dans la littérature définissent des structures de noms à base d'entités invariantes. Dans les faits, les solutions remplacent la

référence à l'entité contingente par son voisinage topologique direct dans l'espoir de remplacer la référence contingente par un ensemble de références invariantes. Cependant, ce voisinage peut lui-même contenir des éléments contingents. Pour éviter de tomber sur une boucle sans fin, les noms utilisés sont ceux des entités ancêtres invariantes à partir desquelles les entités contingentes du voisinage topologique sont historiquement résultantes. Le fait d'utiliser les ancêtres invariants permet d'une part, d'utiliser des références plus robustes pour caractériser une entité, et d'autre part d'éliminer le problème des références croisées qui peuvent apparaître durant l'appariement. Regardons cela de plus près sur la figure 2.20. Dans cet exemple, l'objet modélisé à l'étape 3 est obtenu au moyen d'une opération booléenne entre le cylindre et le bloc. Toutes les faces du cylindre et du cube sont invariantes puisqu'à partir de la primitive de création d'un cylindre ou d'un cube, on sait par avance qu'on va obtenir les faces représentées dans la figure.

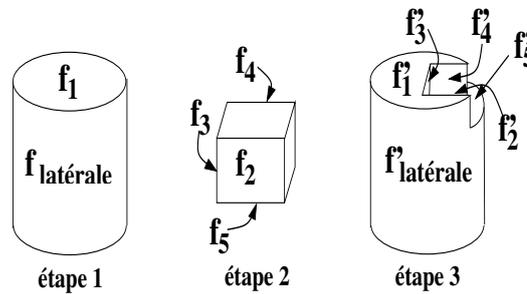


Figure 2.20: Évitemment des références (noms) croisées.

La caractérisation de la face contingente f'_1 est représentée par son voisinage topologique en termes de faces : f'_2, f'_3, f'_4 et $f'_{\text{latérale}}$. Or toutes ces faces sont contingentes. Dès lors, en remontant dans l'historique de construction, il est possible de retrouver l'ancêtre invariant de chaque face contingente appartenant au voisinage. On obtient alors la caractérisation de la face f'_1 par la donnée des faces f_2, f_3, f_4 et $f_{\text{latérale}}$.

2.3.1.5 Utilisation d'un historique de noms

Le cinquième concept commun est celui de l'utilisation d'un historique de noms par les différentes solutions proposées dans la bibliographie. L'utilisation des noms d'ancêtres invariants requiert la capacité de tracer l'historique de chaque entité. Ceci peut être fait soit par l'utilisation de structures dédiées [Kri95, AMP00] ou au moyen d'un mécanisme qui propage des attributs associés à chaque cellule [Che95, WZZZ01]. Par ailleurs, la plupart des approches utilisent uniquement l'historique de faces. Les autres entités (sommets et arêtes) sont nommées par référence aux faces. Il y a une raison principale à cela : en $3D$, les faces sont toujours des entités avec source invariante (toute face provient d'une face invariante). Ce n'est pas toujours le cas pour les arêtes et les sommets. Une arête peut être issu de l'intersection de l'intérieur de deux faces et être donc sans source.

Kripac [Kri95] est le premier à avoir utilisé une structure dédiée, un graphe, pour suivre l'évolution des faces tout au long des processus de construction et de réévaluation. Ce graphe fait partie d'un système d'identification topologique qu'il a mis en œuvre pour résoudre le problème de la nomination persistante. À chaque entité (sommet, arête et face) est associée un identifiant (un nom). Les noms des arêtes et des sommets sont basés sur ceux des faces. Puis, un système

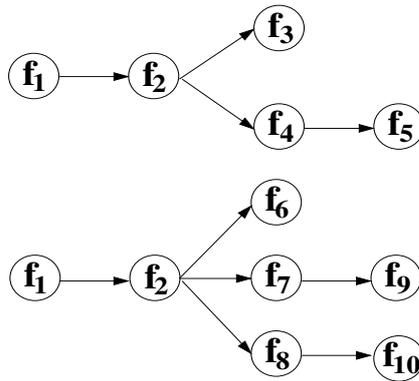


Figure 2.21: Exemple de graphes d'évolution de faces.

d'appariement global calcule les liens de correspondance entre les faces ayant le même ancêtre invariant. De plus, ce dernier compare les noms des arêtes et des sommets du modèle initial avec ceux du modèle réévalué afin d'être appariés. Sur la figure 2.21, nous pouvons voir le graphe qui trace l'évolution de la face f_1 du cylindre de la figure 2.17.

Chen [Che95] a proposé d'utiliser un mécanisme de propagation d'attributs plutôt qu'une structure dédiée de type graphe de faces. Il a donc défini un système paramétrique avec une représentation éditable, appelée EREP¹⁷ [HJ92]. Dans EREP, Chen propose :

1. une structure de noms pour les entités issues des opérations d'extrusion ou de révolution (voir le principe en section 2.3.1.3) ;
2. une caractérisation, basée sur le voisinage topologique (plus ou moins étendu) et l'orientation locale ou celle des caractéristiques de forme, pour les entités contingentes ;
3. un mécanisme de propagation d'attributs (noms des entités invariantes) défini pour associer les faces contingentes à leurs ancêtres invariants. Ces mécanismes de propagation ont été définis dans le cas de la scission, fusion, création et modification d'entités (voir l'exemple d'un mécanisme de propagation lors de scission de faces dans la figure 2.22).

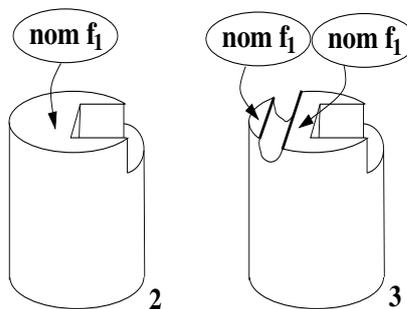


Figure 2.22: Propagation de noms lors de la scission d'une face. Les objets de cette figure sont extraits de la figure 2.17.

17. Au début, EREP a été défini par Hoffmann puis, ce dernier a été repris et enrichi par Chen, son doctorant.

2.3.1.6 Nomination hétérogène des entités

Toutes les solutions mises au point pour une nomination persistante s'accordent à définir une structure de noms utilisant le voisinage topologique [Che95, Kri95, AMP00] et quelque fois des informations géométriques [WZZZ01, BNB05, WN05] (orientation, espace paramétrique, gradient ou autre) lorsque des ambiguïtés apparaissent. Bien que le principe de la représentation des entités de dimension n par l'intermédiaire de leurs bords de dimension $n-1$ soit une constante dans la plupart des modèles topologiques ou Brep, le voisinage topologique de chaque type d'entités n'est pas le même. En effet, le voisinage topologique d'une face comprend un ensemble de faces qui lui sont adjacentes. Une arête est considérée comme l'intersection de deux faces avec, à son bord, un (lorsque l'arête forme une boucle) ou deux ensembles de faces incidents. Alors que le voisinage d'un sommet contient un ensemble de faces. Cette différence induit le fait qu'il y ait autant de mécanismes de nomination et d'appariement que de types d'entités. Cette absence d'homogénéité rend difficile, voir impossible, la manipulation d'entités de plus haute dimension.

2.3.1.7 Structure des plates-formes mises en œuvre

Le septième et dernier concept est l'architecture logicielle globale qui résulte de la plupart des contributions que nous appelons l'architecture à trois couches¹⁸ :

- la couche de spécification paramétrique contient la description des opérations de modélisation (opérations basées sur les caractéristiques, opérations booléennes, opérations de chanfreinage, d'arrondi, d'extrusion, etc.) qui définissent les objets modélisés. Ces opérations se réfèrent à la couche de noms ;
- la couche de noms permet la caractérisation puis l'appariement des entités géométriques et topologiques du modèle paramétrique ;
- la couche géométrique contient les entités géométriques et topologiques qui constituent l'instance courante de la géométrie, c'est-à-dire les objets ou la scène résultant de la dernière opération de modélisation à un instant donné.

2.3.2 Points différents

Ajouté aux concepts précédents, un certain nombre de spécificités liées aux nouvelles exigences dictées actuellement par les systèmes paramétriques sont à étudier. Il s'agit entre autre des contributions qui mettent en œuvre des mécanismes d'appariements locaux ou globaux, celles qui abordent les problèmes liés à l'interaction d'objets courbes ou celles qui apportent une solution partielle à la nomination d'objets hiérarchiques (par exemple, des agrégats de faces).

18. Notons ici qu'il y a une différence entre l'architecture d'un système paramétrique, telle qu'elle a été définie en section 2.1.1, et l'architecture des contributions faites pour résoudre le problème de la nomination persistante. Historiquement, au début, la couche de spécification paramétrique et la couche géométrique ont été définies comme seuls constituants des systèmes paramétriques. Plus tard, les différentes contributions au problème de la nomination persistante ont développé des mécanismes permettant la gestion des noms. Ces mécanismes, qui d'un point de vue théorique, sont regroupés au sein d'une couche à part entière, ont été au fil du temps répartis au sein des couches géométrique et de spécification.

2.3.2.1 Solutions de nomination avec un appariement global ou local

Il est possible de classer les solutions proposées pour résoudre le problème de la nomination persistante, selon le mécanisme d'appariement qui peut être *global* ou *local*. Dans le cas global, l'appariement s'effectue par la comparaison de deux ensembles d'entités (l'un issu du modèle initial et l'autre issu du modèle réévalué) puis la mise en correspondance de toutes ces entités [Kri95, AMPT03]. Le calcul combinatoire qui en découle, est évité dans l'appariement local puisque seule l'entité référencée intervient. En effet, Soit cette dernière est comparée à un ensemble d'entités candidates [Che95], ou alors elle porte en elle les noms des nouvelles entités [WZZZ01]. Les différents mécanismes d'appariement global sont plus couteux en termes de temps de calcul. Cependant, ils permettent d'obtenir des résultats plus fiables.

Pour illustrer le principe sur lequel repose chaque type d'appariement, prenons l'exemple de la figure 2.23. Les modèles représentés dans cette figure résultent de l'application d'une rainure sur un bloc extrudé. Durant la réévaluation, le paramètre concernant l'orientation de la rainure est modifié.

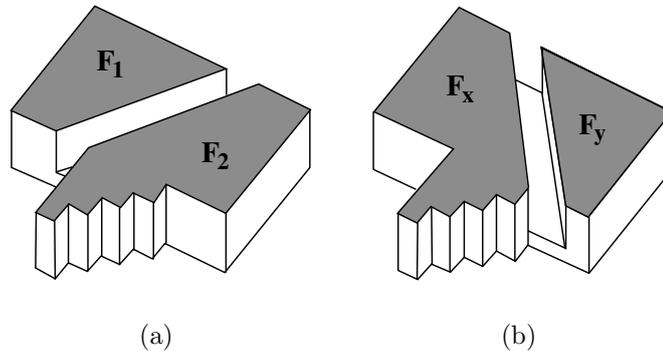


Figure 2.23: (a) Une rainure sur un bloc extrudé (b) Réévaluation du modèle après la modification de la direction de la rainure.

Dans cette configuration, supposons que la face F_2 soit référencée par la prochaine opération de modélisation de la spécification paramétrique. Une méthode d'appariement local cherche, parmi les faces du modèle réévalué F_x et F_y , celle ayant le voisinage topologique le plus proche de F_2 . Il se trouve, par exemple, que le nombre de faces communes entre F_2 et F_x est plus élevé que celui entre F_2 et F_y . La face F_2 sera donc apparié à la face F_x . À l'inverse, une méthode d'appariement global va chercher à apparier les faces du modèle initial (F_1 et F_2) à celles du modèle réévalué (F_x et F_y). Dans ce cas, le bord de la face F_1 est totalement contenu dans celui de F_x . Cela induit l'appariement de F_1 sur F_x et par conséquent celui de F_2 sur F_y .

2.3.2.2 Solutions appropriées aux cas polyédriques et celles spécifiques aux cas non-planaires

Les entités référencées à partir de la couche de spécification paramétrique sont principalement des entités topologiques. Comme énoncé plus haut, de part la définition des entités contingentes, c'est dans ce cas de figure que se concentrent les plus grandes difficultés au niveau

de la caractérisation et de l'appariement. Dans le cas d'objets polyédriques où des faces planes s'intersectent (par exemple le modèle (a) de la figure 2.24), le voisinage topologique suffit pour distinguer les entités les unes par rapport aux autres. Cependant, lorsque les faces d'objets non-planaires s'intersectent (voir les modèles (b) à (k) de la figure 2.24), les informations purement topologiques ne suffisent plus.

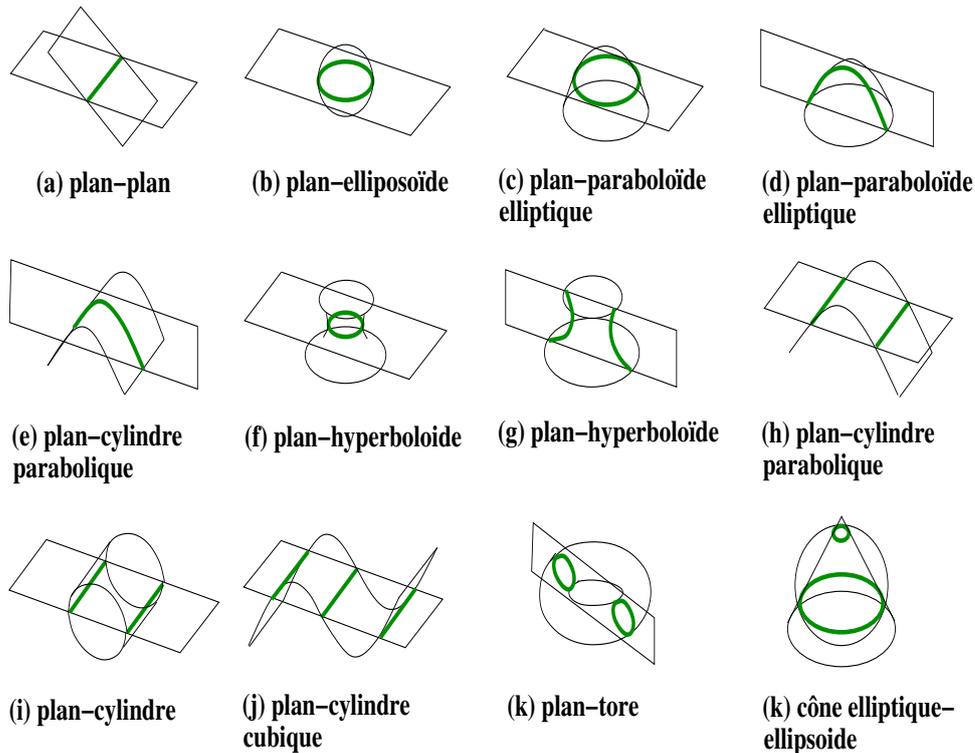


Figure 2.24: Exemples d'arêtes résultant de l'intersection de surfaces. (a) Exemple d'une arête résultant de l'intersection de faces planaires. (b) à (k) Exemples dans lesquels l'intersection de surfaces courbes génère plusieurs arêtes.

C'est alors que des contributions comme celles de Bidarra, Wu ou Wang (dont le détail est donné ci-après) sont apparues pour résoudre le cas où plusieurs entités topologiques ont le même voisinage. Le principe est d'utiliser, en plus de la topologie, des informations géométriques telles que le gradient, les coordonnées paramétriques des points représentant les entités à nommer, etc. Ces solutions ne sont pas complètes dans le sens où elles ne résolvent pas toutes les configurations possible ; néanmoins, leur analyse est intéressante dans le sens où elle nous permet de saisir les problèmes qui restent ouverts dans ce contexte.

Bidarra

Pour Bidarra [BNB05], la solution au problème de la nomination persistante est basée sur les caractéristiques de formes. Il considère un modèle paramétrique uniquement comme un ensemble inter-relié de classes d'entités, à savoir ce qu'il appelle :

- les classes de références : une référence est uniquement une entité auxiliaire nommée avec

2.3. SOLUTIONS PROPOSÉES

une géométrie préalablement prédéfinie. Elle peut être associée à d'autres entités dans une définition paramétrique au moyen de contraintes géométriques. Un exemple simple de ce type d'entités correspond à un plan de référence sur lequel peut s'appuyer la définition d'autres entités du modèle paramétrique. Plus spécifiquement, une entité de référence peut être le plan de la figure 2.25 par rapport auquel vont être positionnés une série de trous. Dès lors, le déplacement de ces trous peut être facilement réalisé, en une étape, par le déplacement (modification) de ce plan référence ;

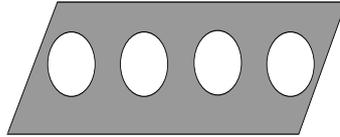


Figure 2.25: Exemple d'une entité référence (plan de référence).

- les classes de caractéristiques déclaratives : il s'agit de classes permettant une description structurée de toutes les propriétés d'un type de caractéristique donné (comme les caractéristiques de forme), définissant alors un *template* (au sens anglophone du terme) pour toutes les instances de la classe. Par exemple, la classe des formes cylindriques dans laquelle chaque instance est structurée avec une face du haut, une face du bas et une face latérale mais qui se décline en différentes dimensions (voir l'exemple de la figure 2.26.)

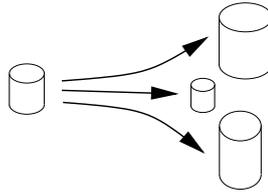


Figure 2.26: Exemple de caractéristique déclarative.

- les classes de caractéristiques procédurales : les caractéristiques procédurales sont les caractéristiques qui prennent en paramètre les arêtes qu'elles référencent. Par exemple, le chanfreinage ou l'arrondi des arêtes. Deux propriétés justifient le fait que ce type de caractéristiques forme une catégorie à part entière. La première est que contrairement aux caractéristiques déclaratives, les caractéristiques procédurales sont liées à l'existence réelle de (ou des) arête(s) référencées. La seconde concerne la forme de ce type de caractéristique qui peut être uniquement déterminé après l'identification et la localisation de l'arête sur laquelle cette caractéristique est appliquée. Dans cette dernière catégorie, les références aux arêtes peuvent présenter des ambiguïtés dans le cas de l'intersection de faces non planaires. C'est en cela qu'elles concentrent l'attention de Bidarra. Dans la figure 2.27, c'est le cas des arêtes e_1 et e_2 . En effet, ces dernières résultent de la face de devant du bloc et la face latérale du cylindre. Étant donnée la courbure de cette face latérale, les arêtes e_1 et e_2 sont incidentes à la même surface. Cela a pour effet de rendre leur distinction impossible à travers les seules informations de voisinage topologique. Pour lever cette ambiguïté, Bidarra a défini une méthode originale de caractérisation. Cette méthode consiste à définir automatiquement, une ou plusieurs paires de demi-espaces si bien qu'aucune paire d'arêtes ne se retrouve dans la même (combinaison de) sous-espaces(s). Pour illustrer le principe

des demi-espaces, prenons l'exemple de la figure 2.27 pour la caractérisation des arêtes e_1, e_2 et e_3 . Sans l'apport du demi-espace représenté dans cette figure, ces trois arêtes ont la même caractérisation : (`<bloc.devant, rainure.latérale>`, `<bloc.haut, bloc.bas>`). Lorsqu'on rajoute l'élément discriminant, qui est le côté où se trouve l'arête considérée (positif ou négatif), l'ambiguïté disparaît et les noms des arêtes deviennent respectivement $e_1 = e_3 = (\text{<bloc.devant, rainure.latérale>}, \text{<bloc.haut, bloc.bas, référence.positive>})$ et $e_2 = (\text{<bloc.devant, rainure.latérale>}, \text{<bloc.haut, bloc.bas, référence.négative>})$.

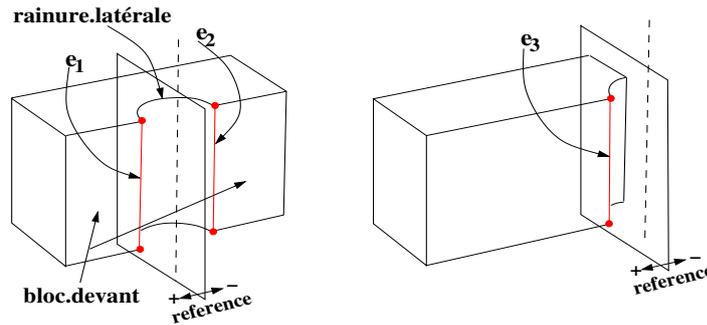


Figure 2.27: Caractérisation d'arêtes avec un partitionnement en demi-espaces.

Cette approche s'avère efficace mais très restrictive puisque la solution reste valable uniquement pour des applications manipulant des surfaces quadratiques : planaires, cylindriques, sphériques et toroïdales.

Wu

Wu [WZZZ01] considère les modèles paramétriques comme une séquence d'opérations booléennes régularisées qui font interagir un objet déjà modélisé (nommé par ses soins *part body* et qu'on note PB) et un objet issu directement d'une caractéristique de forme (nommé *body* et qu'on note B). Lorsque l'objet résultant de l'application d'une caractéristique de forme est créé à chaque face de ce dernier est attribué un nom (appelé *NO* pour *Nom Original*). Ainsi, une opération booléenne est appliquée entre un PB et un B doté de noms associés à chaque face du bord.

Les noms originaux sont sauvegardés tout au long du processus de modélisation paramétrique. Ils sont également propagés, d'une étape à l'autre, aux faces descendantes, dans le cas d'une scission, fusion ou modification afin de lier chaque face contingente à son ascendant invariant. Un nom original est constitué de l'identifiant d'une opération de modélisation (extrusion, révolution, chamfreinage), et d'un ensemble d'identifiants qui permettent de connaître les entités invariantes desquelles est originaire l'entité qui porte ce nom. C'est par exemple le cas des entités issues de l'opération d'extrusion dont le procédé de caractérisation est décrit dans la section 2.3.1.3.

Quand une entité (sommet, arête ou face) est référencée et que les *NO* sont propagés le long des faces descendantes, plusieurs d'entre elles peuvent avoir le même *NO*. Sachant cela, Wu propose d'utiliser ce qu'il appelle un nom réel *NR*. Ce dernier est constitué d'un nom original et des informations additionnelles, appelées *informations de l'espace paramétrique*, permettant d'éliminer l'ambiguïté précédemment décrite. Ainsi, lorsque tout le processus de modélisation

2.3. SOLUTIONS PROPOSÉES

est réévalué, tous les noms réels du PB doivent être appareillés.

Ces informations, purement géométriques, sont extraites de l'espace paramétrique (u, v) de la surface qui porte les entités à caractériser. Un ordonnancement des différents noms, selon les directions u et v de cet espace est effectué. Le critère sur lequel se base cet ordonnancement est la distance entre un point représentatif caractérisant l'entité à nommer et l'origine du système paramétrique. Le but est d'associer un nombre différent à chaque face.

Pour illustrer comment obtenir les informations de l'espace paramétrique d'une entité topologique, prenons l'exemple des figures 2.28(a) et (b) puis calculons l'information de l'espace paramétrique des surfaces F et G . Ce calcul conduit aux résultats de la figure 2.29 dans laquelle la face F (resp. la face G) est modifiée (resp. est scindée) pour donner la face F_1 (resp. les faces G_1 et G_2) du modèle initial et la face F_2 (resp. les faces G_3 et G_4) du modèle réévalué. Par ailleurs, les coordonnées de chaque point caractéristique sont représentées dans le tableau 2.2.

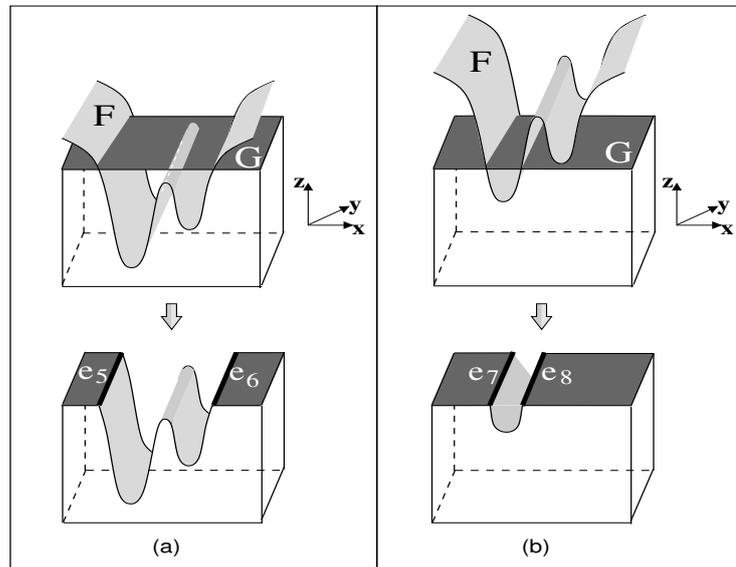


Figure 2.28: (a) Interaction entre le bloc A (à travers sa face du haut G) et la surface courbe F . (b) Réévaluation du modèle résultant en modifiant la position de F par rapport à G .

Aussi, selon le principe de détermination des points caractéristiques de chaque entité topologique, nous pouvons voir que les points A, B, C, D, E et N sont respectivement les points caractéristiques des faces G_1, G_2, G_3, G_4, F_1 et F_2 . Une fois l'ordre des points, déterminé selon la direction de u les noms réels des faces G_1, G_2, G_3, G_4, F_1 et F_2 sont respectivement $NR(G_1) = [NO(G), 1, 2]$, $NR(G_2) = [NO(G), 2, 2]$, $NR(G_3) = [NO(G), 1, 2]$, $NR(G_4) = [NO(G), 2, 2]$, $NR(F_1) = [NO(F), 1, 1]$ et $NR(F_2) = [NO(F), 1, 1]$.

Les autres entités topologiques (sommets et arêtes) sont nommées suivant les noms réels de leurs faces adjacentes et, de façon similaire, une information sur l'espace paramétrique. Comme

Point	A	B	C	D	E	P
(u, v)	(0, 0)	(0.8, 0)	(0, 0)	(0.52, 0)	(0.15, 0.1)	(0.55, 0.1)

Table 2.2: Coordonnées en (u, v) de chaque point.

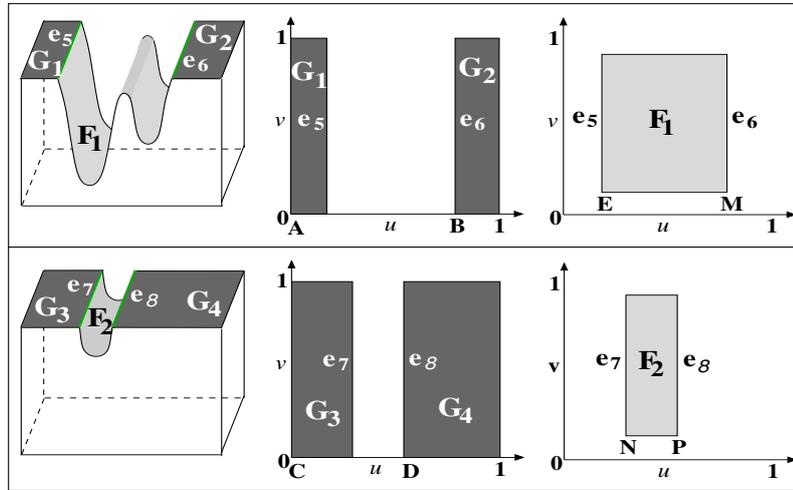


Figure 2.29: Calcul de l'information de l'espace paramétrique.

on peut le voir sur la figure 2.29, les noms réels des arêtes e_5 , e_6 , e_7 et e_8 sont respectivement $[NR(G_1), NR(F_1), 1, 1]$, $[NR(G_2), NR(F_1), 1, 1]$, $[NR(G_3), NR(F_2), 1, 1]$ et $[NR(G_4), NR(F_2), -1, 1]$.

Au final, si une caractéristique de forme référence quelques entités topologiques, ces entités doivent être enregistrées dans la spécification paramétrique sous forme de noms réels. Ainsi, après la réévaluation de l'ensemble des caractéristiques de forme, les références enregistrées doivent être mises en correspondance avec les noms réels référençant les entités du modèle réévalué. Dans l'exemple de la figure 2.29, le modèle de la figure 2.28(a) est réévalué pour donner le modèle de la figure 2.28(b). Ensuite, en utilisant les noms réels des arêtes e_5 et e_6 , on retrouve les arêtes e_7 et e_8 dans le nouveau contexte (les noms originaux des faces G_3 et G_4 sont respectivement les mêmes que ceux des faces G_1 et G_2 puisque ils ont la même face ascendante G). Malheureusement, cette correspondance n'est pas valide car l'arête e_6 ne correspond pas à l'arête e_8 . Manifestement, e_6 résulte de l'intersection du trou droit de la face F avec la face G dans le modèle de la figure 2.28(b) alors que e_8 résulte de l'intersection du trou gauche de la même face avec G dans le modèle de la figure 2.28(c). Les noms réels utilisés par Wu garantissent l'unicité de la référence (toute référence est globalement unique) dans le processus de caractérisation. Cependant, ils ne donnent pas, dans bon nombre de cas, les informations pertinentes nécessaires à un appariement valide.

Wang

Wang [WN05] propose une définition générale d'une famille paramétrique, basée sur le principe de la continuité géométrique introduit par Shapiro [SV95]. Il développe un schéma d'identification sémantique (*semantic id scheme*) basé sur la géométrie au lieu de la topologie. En effet, Wang et ses collègues considèrent la première plus robuste que la seconde. Dans sa contribution, l'identifiant d'une entité se décompose en deux parties : le *segment de surface* et ce que Wang appelle le *sens géométrique* de l'identification (orientation et/ou adaptation). Pour l'identifiant d'une entité, le segment de surface comprend la référence à son plongement non borné

2.3. SOLUTIONS PROPOSÉES

(par exemple, le plan support d'une face) et les identifiants des surfaces adjacentes à l'entité caractérisée. Par exemple, dans la figure 2.30, la face f'_3 est générée par une opération de différence booléenne « b ». Si on suppose que S_3, S_1, S_4, S_5 et S_2 sont respectivement les plans support des faces f'_3, f'_1, f'_4, f'_5 et f'_2 , alors f'_3 est caractérisée par le nom $FACE(DIFF(b) : : SURFACE(S_3) - SURFACE(S_1), SURFACE(S_2), SURFACE(S_5), SURFACE(S_4))$. De même, l'arête e_5 est identifiée par le nom $ARÊTE(DIFF(b) : : COURBE(SURFACE(S_1) + SURFACE(S_3)) - SURFACE(S_2), SURFACE(S_4))$.

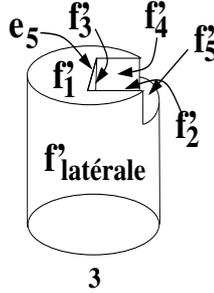


Figure 2.30: Caractérisation à base topologique définie par Wang. Le modèle correspond au résultat de la 3^{ème} opération illustrée dans la figure 2.20.

Cette caractérisation fonctionne bien lorsqu'une seule arête résulte de l'intersection de deux plans (cas planaire). Dans les autres cas (intersection de surfaces courbes), une information supplémentaire (orientation et/ou adaptation) est nécessaire. Pour Wang, l'orientation d'une surface $f(x, y, z) = 0$ au point $p = (x, y, z)$, notée $I_\sigma(f, p)$, correspond à son gradient d'ordre 1 :

$$I_\sigma(f, p) = \nabla f(x, y, z) = \left[\begin{array}{ccc} \frac{\sigma}{\sigma_x} & \frac{\sigma}{\sigma_y} & \frac{\sigma}{\sigma_z} \end{array} \right]^T f(x, y, z).$$

D'un autre côté, l'orientation d'une arête e (droite ou courbe) qui résulte de l'intersection de deux surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$, au point $p = (x, y, z)$, correspond au produit cartésien des deux gradients d'ordre 1 de f et g au point p :

$$I_{11}(f, g, p) = I_\sigma(f, p) \times I_\sigma(g, p).$$

Dans la figure 2.31(a), le plan $G : z = -2$ intersecte la surface $F : z + \frac{1}{15}e^{-(x+5)^2+4} + e^{-0.09x^2+1} = 0$. Les quatre arêtes résultant de l'intersection de G et $F : e_1, e_2, e_3$ et e_4 , bornées par les plans $y + 1 = 0$ et $y - 1 = 0$, ont respectivement les orientations $[0 \ -3.1879 \ 0]^T$, $[0 \ 2.2362 \ 0]^T$, $[0 \ -0.6637 \ 0]^T$ et $[0 \ 0.6647 \ 0]^T$.

Toutefois, lorsque deux arêtes ou plus ont la même orientation, un traitement particulier est appliqué (comme le montre la figure 2.32 avec les arêtes gauche et droite résultant de l'intersection des plans $z = 0$ et $x^3 - x - z = 0$). Dans un cas pareil, des gradients d'ordre supérieur sont calculés, en plus de l'orientation afin de lever l'ambiguïté.

L'adaptation de la surface $f(x, y, z) = 0$ au point $p = (x, y, z)$, $I_\sigma^2(f, p)$, est définie comme le gradient d'ordre 2 :

$$I_\sigma^2(f, p) = \nabla^2 f(x, y, z) = \left[\begin{array}{ccc} \frac{\sigma^2}{\sigma_x^2} & \frac{\sigma^2}{\sigma_y^2} & \frac{\sigma^2}{\sigma_z^2} \end{array} \right]^T f(x, y, z).$$

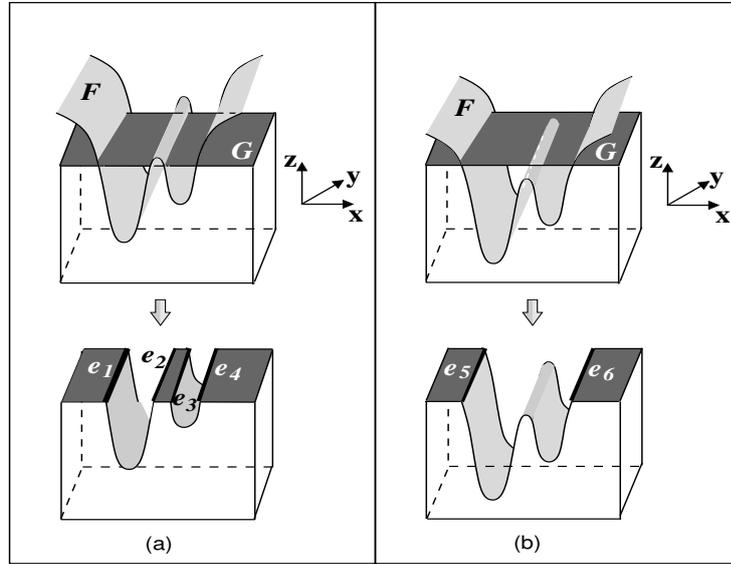


Figure 2.31: Autres réévaluations possibles suite à l'intersection du même bloc et de la même surface que la figure 2.28. Le modèle de la figure 2.31(a) est engendré suite au déplacement de F vers le haut par rapport à sa position initiale dans la figure 2.28(a) ou alors vers le bas par rapport à sa position initiale dans la figure 2.28(b) alors que celui de la figure 2.31(b) est le même que celui de la figure 2.28(a).

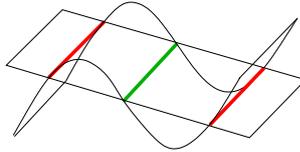


Figure 2.32: Insuffisance de l'orientation pour lever l'ambiguïté dans l'approche de Wang. Cas de l'intersection d'un plan et d'un cylindre cubique.

Dès lors, l'adaptation de l'arête résultant de l'intersection des surfaces f et g peut être définie comme suit

$$I_{12}(f, g, p) = I_{\sigma}(f, p) \times I_{\sigma}^2(g, p).$$

$$I_{21}(f, g, p) = I_{\sigma}^2(f, p) \times I_{\sigma}(g, p).$$

$$I_{22}(f, g, p) = I_{\sigma}^2(f, p) \times I_{\sigma}^2(g, p).$$

Après avoir défini tous ces éléments, lorsque la spécification paramétrique est réévaluée, les identifiants des modèles, initial et réévalué, sont comparés. Ce processus débute par l'appariement des entités caractérisées par les identifiants ayant le même segment de surface. Ensuite, si aucune information additionnelle (orientation / adaptation / gradient d'ordre supérieur) n'est incluse dans les identifiants des entités des modèles initial et réévalué, ou qu'une seule arête résulte de l'intersection de deux surfaces, alors Wang considère qu'il y a une correspondance directe entre ces identifiants. Dans les autres cas, Wang ne décrit pas le procédé qui permet l'appariement de deux faces ayant le même segment de surfaces. En revanche, l'appariement entre arêtes est basé sur la *proximité* des unes par rapport aux autres. La k -proximité entre les arêtes e_1 (résultant

2.3. SOLUTIONS PROPOSÉES

de l'intersection de deux surfaces f_1 et g_1) et e_2 (résultant de l'intersection de deux surfaces f_2 et g_2), aux points p_1 et p_2 appartenant respectivement aux arêtes e_1 et e_2 , est définie par

$$k - \text{proximité}(f_1, g_1, f_2, g_2, p_1, p_2) = |p_1 - p_2| + \sum_{i=1}^k \sum_{j=1}^k |I_{ij}(f_1, g_1, p_1) - I_{ij}(f_2, g_2, p_2)|$$

Donc, pour chaque paire d'arêtes à appairer, Wang calcule la k -proximité¹⁹. Une matrice de proximité R , de dimensions $m \times n$, est générée en listant les arêtes du modèle réévalué comme indices des lignes de la matrice et les arêtes du modèle initial comme indices des colonnes de R . Une fois la matrice de proximité construite, l'appariement des arêtes peut être donné en sélectionnant, dans chaque ligne, la plus petite valeur de k -proximité. Prenons l'exemple de la figure 2.31(a) et supposons que le plan change vers $z = -0.5$, au moment de la réévaluation. Dans ce cas, deux nouvelles arêtes e_5 et e_6 (voir la figure 2.31(b)) apparaissent et doivent être appariées aux arêtes e_1, e_2, e_3 et e_4 . La matrice 1-proximité (2.4) est calculée aux points d'intersection avec le plan $y = 0$.

$$\begin{aligned} 0 - \text{proximité} & \begin{pmatrix} 1.6311 & 2.8719 & 4.8478 & 8.4375 \\ 10.2634 & 8.4786 & 6.3633 & 2.9076 \end{pmatrix} \\ 1 - \text{proximité} & \begin{pmatrix} 3.4742 & 6.4529 & 5.5289 & 10.447 \\ 13.8417 & 10.3244 & 7.4174 & 3.1819 \end{pmatrix} \end{aligned} \quad (2.4)$$

La plus petite valeur de la première ligne (resp. de la seconde ligne) correspond à la 1-proximité entre les arêtes e_5 et e_1 (resp. les arêtes e_6 et e_4). Donc, le processus d'appariement de Wang associe respectivement les arêtes e_5 et e_6 à e_1 et e_4 .

Bien que des résultats satisfaisants soient obtenus, l'approche de Wang présente quelques limitations :

1. Lorsque le processus d'appariement de $m \times n$ entités est effectué ($m > n$), il peut en résulter une correspondance invalide. Dans l'exemple de la figure 2.31, supposons que le modèle de la figure 2.31(b) soit conçu en premier et celui de la figure 2.31(a) en second. La matrice 1-proximité (2.5), dans ce cas, est égale à la transposée de la matrice 1-proximité (2.4).

$$1 - \text{proximité} \begin{pmatrix} 3.4742 & 13.8417 \\ 6.4529 & 10.3244 \\ 5.5289 & 7.4174 \\ 10.447 & 3.1819 \end{pmatrix} \quad (2.5)$$

Il s'avère alors que cette matrice renvoie vers un appariement inexact des arêtes e_1, e_2 et e_3 sur l'arête e_5 (de part les plus petites valeurs des 3 premières lignes). Par ailleurs, considérons les arêtes de la figure 2.33(a) comme le résultat de l'intersection, entre le plan $G : z = -3.5$ et la surface $F : z + \frac{1}{15}e^{-(x+2)^2+4} + e^{-0.09(x-3)^2+1} = 0$, lors de la phase de construction. Si l'on modifie les positions du plan G et de la surface F , de façon à obtenir

19. Généralement, k est l'ordre le plus élevé d'un gradient de surface dans l'identifiant d'une arête.

$z = -2$ pour G et $z + \frac{1}{15}e^{-(x+5)^2+4} + e^{-0.09x^2+1} = 0$ pour F , lors de la phase de réévaluation (voir la figure 2.33(b)), nous obtenons la matrice 1-proximité suivante :

$$1 - \text{proximité} \begin{pmatrix} 4.6894 & 6.7979 & 3.2123 & 7.3957 \\ 9.8238 & 2.9172 & 4.3425 & 5.2495 \end{pmatrix}$$

cela conduit à l'appariement inexact de e_7 sur e_3 et de e_8 sur e_2 .

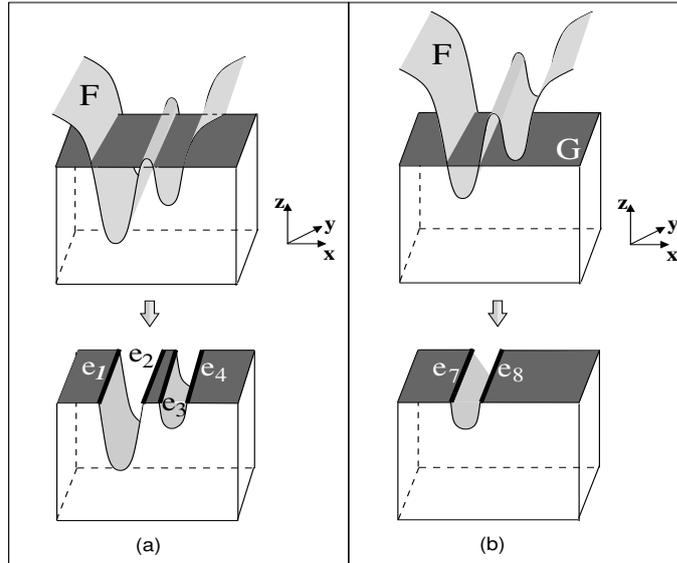


Figure 2.33: Appariement inexact de Wang entre les deux modèles de cette figure.

2. Cette méthode ne prend en compte que les surfaces implicites, définies à l'infini dans un espace $3D$. C'est d'ailleurs parce que cette méthode nécessite de manipuler des surfaces définies à l'infini qu'elle ne traite pas les surfaces paramétriques. Il s'agit là d'une limitation importante car la majorité des systèmes paramétriques utilisent des surfaces paramétriques.

2.3.2.3 Nomination des entités de dimension supérieure à 2

Les solutions que nous avons présentées ou citées jusqu'ici se focalisent sur la nomination persistante d'entités atomiques telles que des sommets, des arêtes ou des faces. Mais peu d'entre elles considèrent la nomination persistante dans le contexte des modèles hiérarchiques. Quelques travaux abordent partiellement cette question en traitant le problème de la nomination de certains agrégats, et plus particulièrement, les agrégats de faces (coques). Le fait de ne pas prendre en compte ce genre d'entité limite le champs des possibilités qui s'offrent à l'utilisateur pour transformer ses intentions en faits réels et ce de façon naturelle (voir la section 2.2.3). Pour cette raison, nous faisons une distinction entre le processus de nomination persistante défini pour les entités atomiques et celui défini pour les coques, voir d'autres types d'agrégats.

Dans la pratique, la plupart des systèmes commerciaux de CAO (par exemple, CATIA ou TopSolid) permettent la nomination de certaines coques. Mais soit le domaine des coques référençables est considérablement restreint à quelques agrégats particuliers (par exemple le bord

2.3. SOLUTIONS PROPOSÉES

complet d'un objet extrudé ou celui issu d'une caractéristique de forme), soit le processus de réévaluation conduit à des problèmes de pertinence et de stabilité du mécanisme de nomination. L'exemple présenté dans la figure 2.34 illustre la difficulté pour les systèmes paramétriques actuels à gérer le problème de la nomination persistante pour des coques.

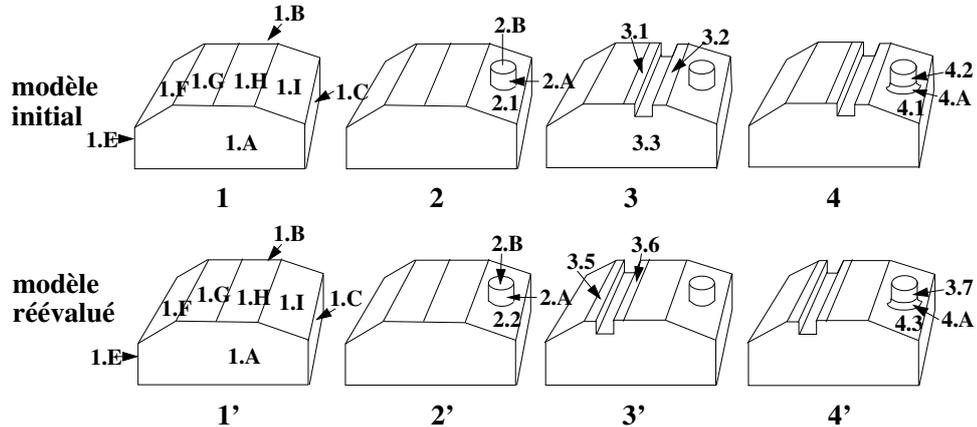


Figure 2.34: Nomination persistante de coques.

Dans cet exemple, le modèle paramétrique contient quatre opérations de modélisation, de gauche à droite (de gauche à droite sur la figure) : la construction d'un bloc en extrudant la face 1.A, l'ajout d'une protrusion cylindrique, la soustraction d'une rainure et l'arrondi de l'arête qui se situe entre une coque et la protrusion. La coque en question, noté S^{ini} , est composée des faces 2.1 et 3.2. Cet agrégat de faces est l'une des deux coques apparaissant à l'étape 3 et qui résultent de la scission de la coque initiale créée à l'étape 1 et composée des faces 1.F, 1.G, 1.H et 1.I. Durant le processus de réévaluation, la position de la rainure est modifiée (voir l'étape 3'). Il en résulte la modification de la topologie de l'objet. Afin de réévaluer la quatrième opération de modélisation (opération d'arrondi), il est nécessaire de calculer dans le nouveau contexte la coque correspondant à S^{ini} (c'est-à-dire trouver la coque correspondant à S^{ini}).

Intuitivement, une coque étant une composition de faces, il semble naturel d'utiliser les appariements réalisés au niveau face pour déterminer celui des coques. La solution définie par Marcheix [Mar06] en est une parfaite illustration. Dans sa contribution, Marcheix définit un modèle de nomination persistante qui reprend un principe fondateur des solutions précédentes : toujours se baser sur les entités invariantes. Pour commencer, le modèle s'appuie sur deux structures de données importantes :

- un graphe de coques : une structure hiérarchique dont le plus bas niveau est constitué de faces et les niveaux supérieurs sont construits par composition de faces et/ou de coques. Tous les nœuds de ce graphe représentent des éléments invariants dans le modèle paramétrique. Par exemple, dans la figure 2.34, le bloc extrudé peut être décrit en termes de coque initiale, de coque latérale et de coque finale. La coque initiale et la coque finale sont respectivement représentées par les faces 1.A et 1.B alors que la coque latérale peut être décomposée en terme de face de droite 1.C, de face du bas 1.D, de face de gauche 1.E et d'une coque représentant le haut du bloc extrudé, composée des faces 1.G, 1.H, 1.I et 1.F ;

– un graphe traçant l'historique de faces (voir la section 2.3.1.5).

Ensuite, l'appariement de deux coques contingentes C_1 et C_2 s'appuie sur les conditions suivantes :

1. C_1 et C_2 ont le même ancêtre invariant : pour vérifier que les deux coques ont le même ancêtre invariant, l'approche utilise le graphe de coques ;
2. C_1 et C_2 ont au moins deux faces appariées. Cela suppose que l'appariement de faces est effectué au préalable.

Ainsi, lorsqu'une coque contingente est sélectionnée dans le modèle initial (par exemple la coque précédente S^{ini}), les liens d'appariements entre les faces de cette coque et les faces du modèle réévalué permettent de trouver les coques contingentes correspondantes à S^{ini} . Ce procédé est effectué en plusieurs étapes :

1. construire un ensemble de faces du modèle réévalué correspondantes aux faces de la coque S^{ini} ;
2. à partir de chaque face de cet ensemble, faire un parcours du modèle géométrique en exploitant les relations d'adjacence entre faces. Les faces retenues dans ce parcours sont celles qui ont le même ancêtre invariant que S^{ini} ;
3. chaque composante connexe constitue une coque correspondante à S^{ini} .

Dans l'exemple de la figure 2.34, la recherche des coques contingentes correspondantes à la coque $S^{ini} = \{2.1, 3.2\}$ passe par les étapes suivantes :

1. les faces correspondantes à 2.1 et 3.2 sont respectivement les faces 2.2 et 1.H ;
2. le parcours topologique à partir des faces 2.2 et 1.H dans le modèle réévalué permet de trouver la face 3.6. En effet, parmi les faces adjacentes à 2.2 et 1.H, seule la face 3.6 a comme ancêtre invariant la coque du haut du bloc extrudé.
3. la coque contingente correspondant à S^{ini} est la coque composée des faces 2.2, 1.H et 3.6.

2.3.3 Analyse et limites des solutions existantes

Comme nous venons de le voir, une nomination persistante repose, en grande partie, sur la donnée d'éléments stables que forment les entités invariantes. Ainsi, lorsque la spécification paramétrique référence une entité contingente, le nom utilisé pour la caractériser est constitué d'informations issues de son voisinage topologique. Ce voisinage ne comprend évidemment pas que des entités invariantes. Par conséquent, les solutions proposées remplacent les entités contingentes par leurs ancêtres invariants en utilisant l'historique de noms présenté dans la section 2.3.1.5.

Dans la liste des entités topologiques constituant le voisinage topologique, les éléments retenus sont les faces. La raison est qu'en $3D$, les faces sont les entités offrant le plus de stabilité et de robustesse à même de résoudre le problème de la nomination persistante, puisqu'il est toujours possible de trouver pour chaque face contingente, un ancêtre invariant remontant dans l'historique de modélisation. Or, ceci n'est pas le cas pour les arêtes et les sommets. De plus, la structure du voisinage topologique est différente selon la dimension de l'entité considérée. D'où, le caractère hétérogène des mécanismes de nomination déployés. La structure du nom et le mécanisme d'appariement sont différents, qu'il s'agisse des faces, des arêtes ou des sommets.

2.3. SOLUTIONS PROPOSÉES

Enfin, pour capturer les intentions de conception, les mécanismes de nomination doivent gérer la caractérisation et l'appariement d'entités de plus haute dimension.

Dans cette optique, Marcheix [Mar06] a proposé un schéma hiérarchique de nomination persistante (figure 2.35). La caractérisation et l'appariement des entités atomiques (sommets, arêtes et faces) s'effectuent selon l'approche d'Agbodan [Agb02]. En effet, le nom de chaque entité comporte le voisinage topologique en termes de faces invariantes adoptant ainsi le principe décrit ci-dessus. Quant à l'appariement, l'approche adoptée est globale. Elle mesure, d'une part, des *ratios de ressemblance* entre les faces ayant le même ancêtre invariant, puis, d'autre part elle utilise les correspondances ainsi calculées pour initier celles entre les sommets et celles entre les arêtes.

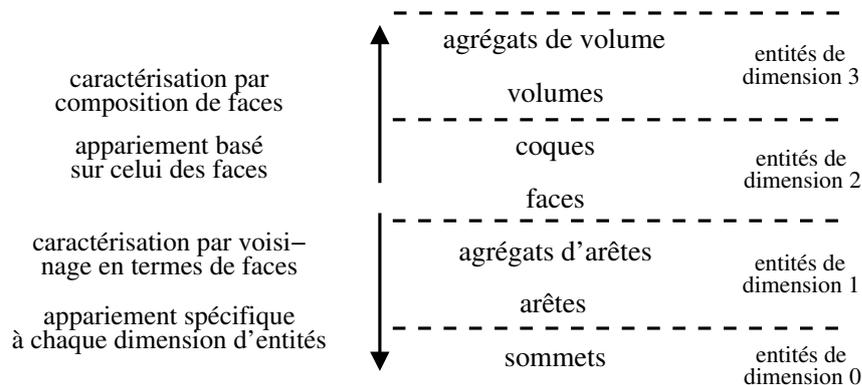


Figure 2.35: Schéma hiérarchique de nomination persistante proposé par [Mar06].

Par ailleurs, la caractérisation et l'appariement des coques s'appuient sur ceux des faces mais de façon différente. En effet, le nom d'une coque comprend, de part la propriété d'agrégation, l'ensemble des faces qui délimitent son intérieur. Dès lors, le processus d'appariement exploite toujours les faces (les entités les plus stables), non pas par la propriété d'incidence, mais par celle d'agrégation. Cet état de fait marque une rupture avec ce qui est a été établi pour les dimensions inférieures (arêtes et sommets). Il en résulte que les mécanismes de nomination définies au niveau des faces sont exploités *de la même façon* (ou de façon homogène) au niveau des dimensions supérieures.

Donc, afin d'homogénéiser les mécanismes de nomination à toutes les entités et les agrégats d'entités en $3D$, nous avons choisi les arêtes comme élément de base pour mettre en œuvre un système hiérarchique de nomination persistante. La nécessité est alors de définir un système de caractérisation et d'appariement assez robuste entre les arêtes. Ensuite, la caractérisation et l'appariement des entités et agrégats d'entités, de toute dimension, s'effectue en exploitant celui des arêtes et en utilisant le fait que ces dernières peuvent se définir comme un agrégat d'arêtes (voir la figure 2.36). La caractérisation et l'appariement des sommets restent quant à eux spécifiques car d'une part, considérer un agrégat de sommets n'apporte pas les informations nécessaires à la description du voisinage topologique d'une entité quelconque, et, d'autre part leurs référencement n'est pas aussi fréquent que celui des arêtes, des faces ou des coques.

Par construction, les arêtes sont considérées comme le résultat de l'intersection de deux faces. Étant donnée le caractère relativement stable des faces en $3D$, cela rend plus robuste la caractérisation des arêtes du moment que leurs noms contiennent les deux faces qui s'in-

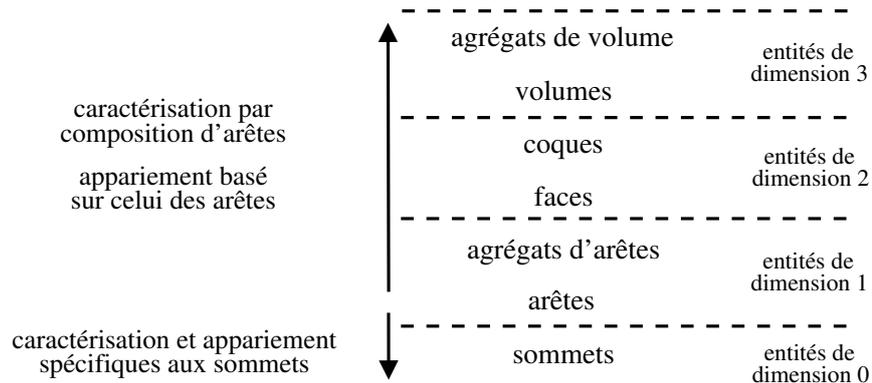


Figure 2.36: Schéma hiérarchique de nomination persistante basé sur les arêtes.

tersectent. Cependant, lorsqu'on envisage la nomination des entités de dimension plus élevée (4 dimensions, voire au-delà), se reposer uniquement sur les faces entraîne des limitations. En effet, comme dans le cas des arêtes en $3D$, l'intersection de deux entités de dimension $n, n \geq 4$ (hyper volumes en $4D$), peut engendrer des faces sans sources. Les faces deviennent alors moins stable et rendent moins robuste la caractérisation des arêtes. La solution est alors d'utiliser pour cette caractérisation, les entités de dimension $n - 1$, en nD , comme l'ont été les faces (entités de dimension 2) en $3D$. Dans ce cas, l'intersection de deux entités de dimension $n - 1$ génère plus d'une arête et la seule donnée des ces deux entités ne suffit pas à lever l'ambiguïté de leurs caractérisations. C'est exactement la même ambiguïté qui apparaît lorsque deux faces s'intersectent dans un cas non-planaire (l'intersection de ces deux faces générant la plupart du temps plusieurs arêtes d'intersection). Nous comptons donc nous inspirer des méthodes de caractérisation et d'appariement dans le cas des faces non planaires pour déterminer des solutions aux problèmes de caractérisation et d'appariement d'entités de dimensions supérieures plongées linéairement.

2.4 Conclusion

Dans ce chapitre, nous avons présenté l'évolution suivie par la modélisation dans le domaine de la représentation virtuelle d'objets réels. Nous avons remarqué que cette évolution a commencé (et se poursuit pour certaines outils) par la modélisation géométrique qui représente uniquement les objets à travers le prisme de leurs propriétés géométriques. Ensuite, elle s'est poursuivie par la modélisation par caractéristiques qui répond au besoin d'introduire des informations fonctionnelles pour une description plus adaptée au cycle de vie d'un produit industriel. Enfin, elle a abouti à la modélisation paramétrique qui est apparue pour faciliter la réévaluation des objets conçus en déclinant sous diverses variations, une même description du processus de modélisation. De part sa structure duale, un système paramétrique pose le problème de la nomination persistante. Ce problème pose à son tour trois impératifs :

- La nécessité d'une structure de noms à même de caractériser, de façon unique et non ambiguë, chaque entité du modèle géométrique manipulé.
- La nécessité d'un appariement d'entités à travers leurs noms lors de la réévaluation.
- La nécessité d'une capture des différentes sémantiques afin d'étendre les références pouvant

2.4. CONCLUSION

être utilisées par les opérations de modélisation, et de permettre au concepteur d'exprimer très précisément ses intentions de conception.

Dans ce chapitre, nous avons passé en revue les différentes solutions proposées dans la littérature pour résoudre le problème de la nomination persistante. Nous avons dressé un ensemble de points communs et de différences entre ces solutions. L'analyse qui en découle a permis de mettre en évidence les difficultés et les lacunes qu'ont ces systèmes à traiter les entités et les agrégats d'entités de dimension supérieure à 2 et celles résultant de l'interaction d'objets non-planaires. Ces insuffisances sont dues à la structure des systèmes hiérarchiques proposés qui elle-même est induite par la prééminence des faces en $3D$.

Pour y remédier, nous proposons un système hiérarchique de nomination persistante basée sur les arêtes. Les mécanismes de nomination sont alors définis au niveau des arêtes, puis généralisés aux dimensions supérieures. Le système paramétrique que nous mettons en œuvre utilise le noyau géométrique des cartes généralisées. Pour ajouter la dimension hiérarchique à ce type de modèle qui manipule des quasi-variétés de dimension quelconques, quatre opérations sont nécessaires. Deux d'entre elles (suppression et contraction) ont été définies par [DL03] et nous définissons les deux autres (insertion et éclatement) dans le chapitre 3. Par ailleurs, la gestion des noms des arêtes se fait principalement au moyen d'un graphe traçant l'évolution de ces dernières au cours du processus de modélisation paramétrique. Donc, à chaque étape de ce processus, le système doit savoir ce qui se passe au niveau du noyau géométrique. Dans notre travail, nous répondons à cet objectif en mettant en œuvre les mécanismes permettant la détection de tous les événements (création, scission, fusion, suppression et modification de cellules) survenus sur les cellules du modèle géométrique. Une fois décelés, ces événements sont regroupés au sein d'une structure que nous appelons *journal de bord*. Les quatre opérations et la structure du journal de bord sont les outils nécessaires à la définition du système de nomination hiérarchique.

Chapitre 3

Suivi des événements dans un modèle topologique

Sommaire

3.1	Introduction	55
3.2	Évolution de la topologie au sein du noyau géométrique	58
3.2.1	Mécanismes de suivi d'événements	59
3.2.2	Génération du journal de bord	66
3.3	Les opérations de base	69
3.3.1	Quelques notions supplémentaires autour des cartes généralisées	69
3.3.2	Suppression et contraction	71
3.3.3	Insertion et éclatement	79
3.4	Implantation des mécanismes de suivi d'événements	87
3.4.1	Solution apportée par la structure de <i>tag</i>	87
3.4.2	Opérations élémentaires et mécanismes de suivi d'événements intégrés	89
3.5	Conclusion	101

3.1 Introduction

Dans le précédent chapitre, nous avons décrit la nécessité d'un système de nomination « hiérarchique », utilisé pour combler certaines lacunes des différentes approches proposées pour résoudre le problème de la nomination persistante. Dans cette perspective, nous avons donc décidé de prendre appui sur les arêtes au lieu des faces afin d'homogénéiser les processus de caractérisation et d'appariement en utilisant les propriétés d'agrégation et de voisinage. Ces propriétés topologiques, ainsi que d'autres, de type géométrique, sont exprimées au sein du noyau géométrique d'un système paramétrique.

D'un point de vue commercial, la mise en œuvre de systèmes paramétriques nécessite un nombre élevé d'heures de développement. C'est le cas, par exemple, des produits leaders sur le marché tels que CATIA, Pro/ENGINEER, CasCade, etc. Pour cette raison, les nouveaux systèmes qui apparaissent sur le marché utilisent des noyaux géométriques *propriétaires* déjà

développés (Parasolid¹, Acis², etc.). Ces derniers présentent l'avantage de fournir un ensemble complet d'opérations de modélisation (opérations booléennes, rainure, chanfreinage, etc.), mais ont l'inconvénient d'empêcher l'accès à leurs différentes fonctionnalités et structures internes. Pour passer outre le caractère propriétaire de tels noyaux, nous avons choisi d'utiliser le modèleur « open source » Moka³ défini sur la base des cartes généralisées.

Structurellement, un modèleur géométrique est constitué de plusieurs composants désignant deux niveaux d'abstraction différents (voir la figure 3.1). Le premier niveau, en interaction directe avec l'utilisateur, correspond à l'interface d'application qui regroupe les différentes opérations de modélisation (rainurage, lissage, assemblage...). Le second, constituant le cœur du modèleur, utilise un modèle géométrique particulier (Brep, CSG, etc.) qui définit les structures de données et les opérations de base pouvant être utilisées par le niveau précédent. Dans le cas d'un modèle topologique, le noyau géométrique manipule des cellules de différentes dimensions : sommets, arêtes, faces, etc.

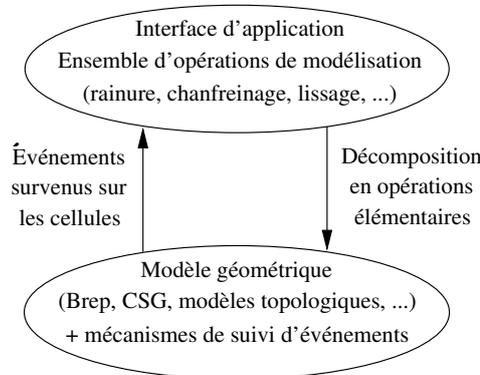


Figure 3.1: Structure d'un modèleur géométrique.

De part leur complexité, les opérations de l'interface d'application peuvent générer des transformations importantes (souvent imprévisibles) sur la structure topologique d'un modèle géométrique. En effet, en fonction du plongement des objets qui interagissent (par exemple leurs positions dans un espace 3D), la topologie résultant d'une opération de modélisation varie. C'est d'ailleurs la raison pour laquelle est apparue le problème de la nomination persistante (voir la section 2.2).

À partir de ce constat, la résolution de ce problème passe par un impératif : celui de savoir ce qui se passe au niveau du noyau géométrique. En effet, le système paramétrique doit savoir qu'un sommet a été *créé*, qu'une arête a été *scindée* ou *supprimée*, qu'une face a été *fusionnée* avec une autre ou que la structure topologique d'un volume a été *modifiée*. Il doit donc répertorier les événements survenus sur les entités topologiques à chaque étape du processus de modélisation paramétrique. Sinon, comment rendre persistant le processus de nomination sans trouver le

1. Parasolid : logiciel phare de la gamme des solutions « PLM Components d'UGS ». C'est le noyau de modélisation volumique d'un grand nombre d'applications mondiales de conception, de fabrication et d'analyse assistées par ordinateur (CAO/FAO/IAO). Pour plus d'informations sur ce produit, visitez <http://www.ugs.com>.

2. ACIS, développé par Spatial Corp., filiale Dassault Systèmes (<http://www.dsdrv.com>), est un standard mondial des environnements CAO 3D. Il est notamment présent dans des logiciels leaders de conception et de simulation tels que les produits Autodesk, Cadkey, Ansys, Fluent, etc.

3. <http://moka-modeller.sourceforge.net/>

3.1. INTRODUCTION

lien entre une entité contingente, apparaissant à une étape du processus de modélisation, et son ancêtre invariant qui apparaît à une étape ultérieure? Comment lier, par exemple, les faces contingentes $f_{3,1}$ et $f_{3,2}$ de la figure 3.2, à leurs ancêtre invariant, la face $f_{1,1}$, sans que le système ai pris connaissance de l'événement suivant : la scission de la face $f_{1,1}$ en deux faces $f_{3,1}$ et $f_{3,2}$ suite à l'application de l'opération booléenne entre les blocs A et B.

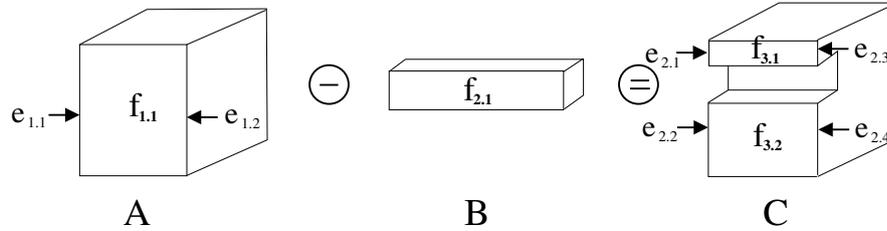


Figure 3.2: Nécessité de connaître l'évolution des entités au sein du noyau géométrique.

Les noyaux géométriques propriétaires fournissent des méthodes permettant de retrouver un certain nombre d'événements. Cependant, du fait de la confidentialité qui entoure ces méthodes, nous savons peu de choses sur leur fonctionnement et sur leurs modèles sous-jacents. Ces noyaux ont leurs propres schémas pour la détection des événements, basés sur leurs propres heuristiques et taxonomies de situations (finies, par définition, et c'est ce qui limite le nombre des cas traités). Il en est de même pour le travail de recherche qu'a mené Armstrong et ses collègues dans le cadre du projet DJINN [ABC⁺00]. Dans ce projet, les fonctions de reconnaissance des transformations survenues sur les entités topologiques ont été formalisées. En effet, Armstrong et al. ont proposé la spécification de plusieurs fonctions dédiées à la gestion des événements survenus sur les entités topologiques. Ce travail de formalisation souligne la nécessité de fournir ces fonctions, mais n'apporte pas de détails sur la façon de générer les événements qui relatent les différentes transformations topologiques survenues. Dès lors, un modèle à part entière est nécessaire pour garantir la fiabilité et l'exhaustivité des événements générés à une étape du processus de modélisation paramétrique.

Dans la troisième section de ce chapitre, nous proposons donc une plate-forme complète qui permet de déceler la création, la scission, la fusion, la modification et la suppression de cellules de toute dimension. Cette plate-forme renvoie systématiquement l'ensemble de ces événements, regroupés au sein d'un « journal de bord », au système hiérarchique de nomination persistante. Nous considérons alors une opération de modélisation comme une « boîte noire ». Cette boîte prend en paramètres des entités de l'instance courante de la géométrie, et restitue d'autres entités ayant subi des transformations topologiques. Le principe est alors de trouver un moyen de lier les entités en entrée avec celles en sortie de la boîte. Les liaisons qui associent une entité à plusieurs décrivent une scission. Celles qui associent plusieurs entités à une seule expriment une fusion. Si une entité à l'entrée de l'opération de modélisation n'est liée à aucune autre à la sortie, il s'agit d'une suppression. La création est décelée dans le cas inverse : une entité en sortie de la boîte noire ne peut être liée avec aucune autre entité à l'entrée de cette dernière. Enfin, lorsque la plate-forme construit une relation entre deux entités caractérisées par des identifiants différents, il est alors question d'une modification topologique entre deux entités différentes.

Pour construire toutes ces associations, nous avons défini cinq mécanismes de suivi d'événements : *créa_élém*, *sup_élém*, *sci_élém*, *fus_élém* et *modif_élém*. Ces derniers sont décrits de façon générique (le fonctionnement de chaque mécanisme ne dépend pas de la dimension des entités)

et indépendante du modèle géométrique utilisé. Il suffit que le modèle permette la définition de la notion de cellule. Or, il se trouve que c'est le cas des modèles géométriques à base topologique dont font parties les cartes généralisées.

L'étude des cartes généralisées a permis de mettre en évidence quatre opérations de modélisation sur la base desquelles peuvent être définies toutes les autres [DL03] : la suppression, la contraction, l'insertion et l'éclatement. Dans la perspective de suivre les événements (transformations topologiques) causés par les opérations de modélisation de ce modèle, nous sommes amenés à étudier de façon formelle la définition de ces opérations de base. Ces opérations ont été introduites pour décrire des structures agrégatives à base de G-cartes que sont les pyramides de cartes généralisées [Sim06]. La description de ces pyramides a nécessité une définition formelle de deux de ces opérations de base : la suppression et la contraction. Les opérations d'insertion et d'éclatement n'ont quand à elle pas été définies formellement. Ces quatre opérations étant à la base des opérations de modélisation et donc du suivi d'évènement qui en découle, nous nous proposons de compléter leurs définitions [BADSM08] dans la deuxième section de ce chapitre.

3.2 Évolution de la topologie au sein du noyau géométrique

Dans un modeleur géométrique, nous distinguons deux niveaux d'opérations que l'on nomme opération globale et opération élémentaire (figure 3.4). Une opération globale (opération booléenne, lissage d'un maillage, triangulation, etc.) est une opération de modélisation géométrique. Disponible au niveau utilisateur et/ou concepteur, elle engendre un ensemble non prévisible d'événements. Sa réalisation peut se faire au niveau interne d'un modeleur (noyau géométrique) par composition d'un ensemble d'opérations élémentaires (opérateurs d'Euler, création de face, scission d'arête, fusion d'arêtes, fusion de faces, identification de faces, etc.). Le choix de ces opérations revient au concepteur en fonction de l'opération globale traitée.

Dans le cas de l'opération globale de différence booléenne $3D$ (figure 3.3), [Gui06] utilise les opérations élémentaires suivantes : scission d'arête, scission de face, insertion d'une arête pendante dans une face⁴ et identification de faces. Les opérations élémentaires ne sont évidemment pas visibles de l'extérieur. Décomposer une opération globale en de telles opérations renforce la maintenabilité (détection et correction aisée des erreurs) et surtout la flexibilité (évolution aisée) du modeleur.

Contrairement à une opération globale, une opération élémentaire engendre un ensemble prévisible d'événements (la figure 3.3 montre le cas de l'opération élémentaire de scission de la face F en F_1 et F_2 , au cours d'une opération globale de différence booléenne). Cet ensemble comprend les événements de création, de suppression, de fusion et de scission de cellules. Pour transcrire ces événements au niveau du journal de bord, nous définissons un ensemble de mécanismes de suivi d'événements : *créa_élém*, *sup_élém*, *fus_élém* et *sci_élém*. Chacun rend compte d'un événement particulier (voir la table 3.1). Donc, en décelant les événements engendrés par une opération élémentaire, nous lui associons les mécanismes de suivi d'événements nécessaires (voir la figure 3.4).

4. Arête accrochée à une face par un côté (un sommet) et libre de l'autre. Une définition formelle de ce type d'entités pour le modèle des cartes généralisées est donnée en section 3.3.1.

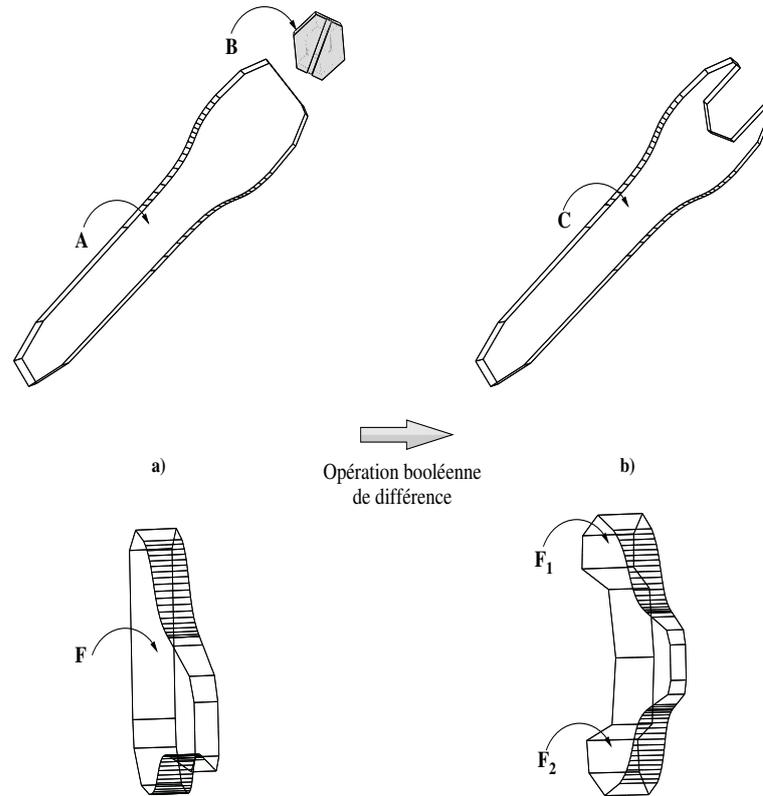


Figure 3.3: Opération globale (différence booléenne) permettant la modélisation d'une clé mécanique. Les figures (a) et (b) présentent les objets vus de devant. Les figures (c) et (d) présentent les objets vus de haut.

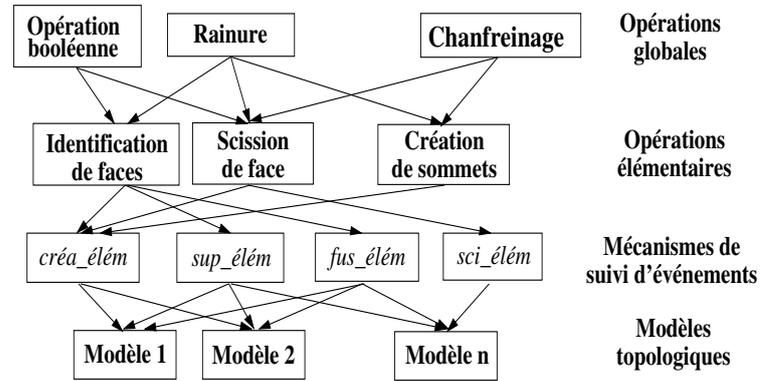


Figure 3.4: Architecture adoptée pour la génération d'une liste d'événements survenus sur les cellules d'un modèle topologique quelconque.

3.2.1 Mécanismes de suivi d'événements

Pour définir les mécanismes de suivi d'événements, nous avons besoin d'introduire les éléments suivantes :

- *ID_Cellule* : un ensemble infini d'identifiants permettant la caractérisation des cellules de

événement	mécanisme de suivi d'événement
scission de cellule	<i>sci_élém</i>
fusion de cellules	<i>fus_élém</i>
création de cellule	<i>créa_élém</i>
suppression de cellule	<i>sup_élém</i>

Table 3.1: Correspondance entre événement et mécanisme de suivi d'événement.

différentes dimensions ;

- *ID_utilisés* : l'ensemble des identifiants utilisés par une opération globale ($ID_utilisés \subset ID_Cellule$).
- *Dim_Max* : la dimension maximale d'une cellule dans un modèle topologique. Lorsque le modèle est en $2D$ (resp. en $3D$), $Dim_Max = 2$ (resp. $Dim_Max = 3$).

Suite à l'application d'une opération globale quelconque O , des événements (induits par le changement de la topologie) apparaissent sur les cellules du modèle topologique considéré. Connaître et répertorier ces événements consiste à déterminer les liens entre les cellules à l'entrée de O et celles en sortie. Pour ce faire, nous procédons en deux étapes :

1. caractériser chaque cellule c par un identifiant id ($id \in ID_Cellule$) ;
2. associer cet identifiant à une paire (ens_id, dim) tel que ens_id ⁵ représente un ensemble d'identifiants référençant toutes les cellules desquelles est originaire c , et dim est la dimension de c . Ainsi, si on suppose que les identifiants id_1, id_2 et id_3 caractérisent respectivement les faces F, F_1 et F_2 de la figure 3.3, alors les paires $(\emptyset, 2), (\{id_1\}, 2)$ et $(\{id_1\}, 2)$ sont associées respectivement aux identifiants id_1, id_2 et id_3 . En effet, F est nouvellement créée et F_1, F_2 résultent de la scission de F .

Les deux associations que nous venons de citer peuvent s'exprimer sous la forme de deux fonctions partielles, *ancêtre* et *dimension*, définies respectivement par

$$ancêtre : \begin{cases} ID_Cellule \rightarrow \mathfrak{P}(ID_Cellule) \\ id \mapsto ancêtre(id) = ens_id \end{cases} \quad \text{et} \quad dimension : \begin{cases} ID_Cellule \rightarrow [0, Dim_Max] \\ id \mapsto dimension(id) = dim \end{cases}$$

($\mathfrak{P}(ID_Cellule)$ symbolise l'ensemble des parties de $ID_Cellule$).

La fonction *ancêtre* lie toute cellule apparaissant durant le processus de modélisation (ou de réévaluation) à un antécédent historique. Afin de ne garder que les événements survenus au cours d'une opération de modélisation (en d'autres termes, à une étape seulement du processus de conception), nous initialisons, au début de l'opération globale, l'ensemble ens_id de la paire (ens_id, dim) avec le singleton $\{id\} : \forall id \in ID_cellule; id \in D_{ancêtre} \Rightarrow ancêtre(id) = \{id\}$ (où $D_{ancêtre}$ est la domaine de définition de la fonction *ancêtre*). Dans le cas contraire, la fonction *ancêtre* relie une cellule avec son antécédent à l'étape de sa création. Ainsi, *ancêtre* peut aller au-delà d'une seule étape du processus de modélisation et ignorer tous les antécédents intermédiaires. L'exemple de la figure 3.5 (qui reprend les trois premières étapes du modèle initial de la figure 2.17) illustre ces deux cas de figures. Les étapes de modélisation de 1 à 3 font apparaître l'évolution de la face f_1 qui est modifiée à l'étape 2 puis scindée à l'étape 3. la face f_2 n'a qu'un seul antécédent, la face f_1 , alors que les faces f_3 et f_4 en ont deux, respectivement f_2 et f_1 . Pour considérer seulement ce qui se produit à partir de l'étape 3, on initialise l'ancêtre

5. ens_id peut être un ensemble vide si c vient d'être créée.

3.2. ÉVOLUTION DE LA TOPOLOGIE AU SEIN DU NOYAU GÉOMÉTRIQUE

de la face f_2 avec son propre identifiant afin que l'historique soit un perpétuel renouvellement à chaque étape.

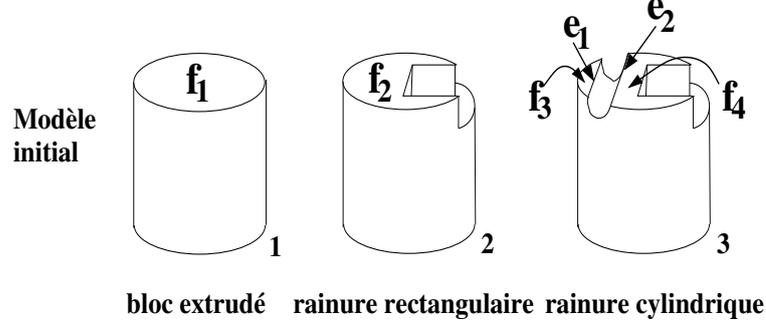


Figure 3.5: Évolution de cellules au cours d'une seule étape du processus de modélisation paramétrique. Les objets de cette figure correspondent aux trois premières étapes du processus de modélisation initiale de la figure 2.17.

Au final, les mécanismes de suivi d'événements sont définis comme l'indiquent les paragraphes ci-dessous :

3.2.1.1 Mécanisme *sci_élém*

Le mécanisme *sci_élém* est associé à toute opération élémentaire qui engendre un événement de scission de cellules. Cet événement rend compte de la décomposition d'une cellule c de dimension dim en plusieurs cellules c_1, c_2, \dots, c_n de même dimension. De façon naturelle, ces nouvelles cellules sont respectivement caractérisées par des identifiants uniques id_1, id_2, \dots, id_n . De plus, pour signifier que :

1. c est l'antécédent historique des cellules c_1, c_2, \dots, c_n à une étape du processus de modélisation ;
2. c_1, c_2, \dots, c_n sont de même dimension que c :

Le mécanisme *sci_élém* associe à chaque id_i la paire (ens_id_i, dim) tel que ens_id_i est un ensemble d'identifiants caractérisant les antécédents historiques (ancêtres) de c_i et vérifiant la propriété suivante :

$$ens_id_1 \subseteq ens_id, \dots, ens_id_n \subseteq ens_id \text{ et } ens_id_1 \cup ens_id_2 \cup \dots \cup ens_id_n = ens_id$$

Prenons l'exemple de la figure 3.6. Dans cet exemple, la face F est scindée en deux nouvelles faces F_1 et F_2 . Avant d'entamer l'opération booléenne, à travers *sci_élém*, nous initialisons l'ancêtre de la face F (c'est-à-dire l'ensemble ens_id_F associé à l'identifiant de la face id_F) avec le singleton $\{id_F\}$. Puis, nous caractérisons les nouvelles faces F_1 et F_2 avec les identifiants id_{F_1} et id_{F_2} , respectivement. Enfin, pour signifier que F est l'ancêtre de F_1 et F_2 (ou que F_1 et F_2 sont originaires de F), nous associons respectivement les paires $(ens_id_{F_1}, dim_{F_1}) = (\{id_F\}, 2)$ et $(ens_id_{F_2}, dim_{F_2}) = (\{id_F\}, 2)$ aux identifiants id_{F_1} et id_{F_2} .

Dès lors, le mécanisme *sci_élém* peut être formalisé par l'algorithme suivant :

Le mécanisme *sci_élém*, décrit par l'algorithme 1, permet d'ajouter de nouvelles relations liant, à travers des identifiants, les cellules résultant de la scission avec leur(s) origine(s) à une étape du processus de modélisation paramétrique (il s'agit de mettre à jour les fonctions *ancêtre*

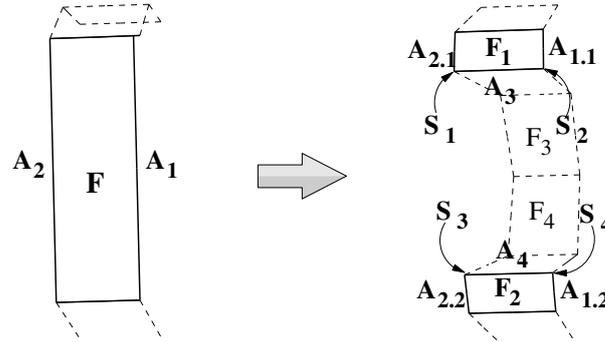


Figure 3.6: Scission de la face F suite à l'application de l'opération booléenne de la figure 3.3 (le modèle illustré représente un « Zoom » local autour de la face F).

et *dimension* définies dans la section 3.2.1). Ce mécanisme prend en paramètres l'identifiant id caractérisant la cellule c (objet de la scission), les identifiants id_1, id_2, \dots et id_n caractérisant les cellules c_1, c_2, \dots et c_n résultant de la scission ainsi que les ensembles ens_id_1, ens_id_2 et ens_id_n qui contiennent les identifiants caractérisant l'origine des cellules c_1, c_2, \dots, c_n . Les points 1 à 4 représentent les conditions préalables au bon déroulement de l'algorithme tandis que les points 5 et 6 en constituent les étapes.

1. c doit obligatoirement exister dans l'instance courante du modèle paramétrique ;
2. id_1, id_2, \dots, id_n sont n identifiants non encore utilisés afin de garantir leurs unicité ;
3. les ancêtres des cellules c_1, c_2, \dots, c_n (caractérisés respectivement par les ensembles $ens_id_1, ens_id_2, \dots, ens_id_n$) constituent une partie de l'ancêtre de c , $ancêtre(id)$ (caractérisé par l'ensemble ens_id) ;
4. les ancêtres de c sont *totalem*ent transmis aux cellules c_1, c_2, \dots et c_n ;
5. ajouter id_1, id_2, \dots, id_n à l'ensemble $ID_utilisés$ car ces derniers viennent d'être employés pour caractériser les cellules c_1, c_2, \dots, c_n ;
6. associer à chaque cellule c_i son ancêtre puis sa dimension.

3.2.1.2 Mécanisme *créa_élém*

Après l'apparition d'une cellule c de dimension dim suite à l'application d'une opération de modélisation, cette dernière doit être caractérisée par un nouvel identifiant id . Celui-ci doit être associé à une dimension dim et à un ensemble d'identifiants $ens_id = \{\}$, signifiant que c n'a pas d'ancêtre. Dans l'exemple de la figure 3.6, les arêtes A_1, A_2, A_3 et A_4 sont nouvellement créées. Le mécanisme fait en sorte de les caractériser respectivement par les identifiants $id_{A_1}, id_{A_2}, id_{A_3}$ et id_{A_4} auxquels ne sont associés aucun ancêtre (cet état de fait est représenté par un ensemble vide $\{\}$) et la dimension 1. Au final, ce traitement peut être décrit par l'algorithme suivant :

Le mécanisme *créa_élém*, décrit par l'algorithme 2, permet d'associer à une cellule c nouvellement créée un identifiant non encore attribué, un entier correspondant à la dimension topologique de c (dans le cas d'une cellule sommet, il s'agit de la dimension 0) et un ancêtre ne contenant aucune référence à une cellule antérieure (puisque c n'apparaît qu'après le début de l'opération globale). Les points 1 et 2 représentent les pré-conditions et 3 et 4 décrivent le fonctionnement de cet algorithme, comme suit :

Algorithme. 1 : mécanisme *sci_élém*

Données : ■ $id, id_1, id_2, \dots, id_n$: n identifiants ;
 ■ $ens_id_1, ens_id_2, \dots, ens_id_n$: n ensembles d'identifiants qui caractérisent l'origine des cellules c_1, c_2, \dots et c_n .

Résultat : ajout des liens *ancêtre* et *dimension* relatifs aux cellules c_1, c_2, \dots, c_n .

début

```

    si  $\exists id, id_1, id_2, \dots, id_n, ens\_id_1, ens\_id_2, \dots, ens\_id_n$ 
    tels que
1   |    $id \in D_{ancêtre}$  et
2   |    $\forall i \in \{1, 2, \dots, n\}, id_i \in ID\_Cellule - ID\_utilisés$  et
   |    $\forall i, j \in \{1, 2, \dots, n\}, id_i \neq id_j$  et
3   |    $\forall i \in ens\_id_i \subseteq ancêtre(id)$  et
4   |    $ancêtre(id) = ens\_id_1 \cup ens\_id_2 \cup \dots \cup ens\_id_n$ 
   |
   |   alors
5   |    $ID\_utilisés \leftarrow ID\_utilisés \cup \{id_1, id_2, \dots, id_n\}$  ;
6   |    $ancêtre \leftarrow ancêtre \cup \{(id_1, ens\_id_1), (id_2, ens\_id_2), \dots, (id_n, ens\_id_n)\}$  ;
   |    $dimension \leftarrow dimension \cup$ 
   |    $\{(id_1, dimension(id)), (id_2, dimension(id)), \dots, (id_n, dimension(id))\}$  ;

```

fin

1. id est un identifiant, non encore utilisé, caractérisant c ;
2. dim est la dimension de c comprise dans l'intervalle $[0, Dim_Max]$;
3. marquer id comme identifiant déjà utilisé ;
4. n'associer aucun ancêtre à c caractérisée par l'identifiant id .

3.2.1.3 Mécanisme *fus_élém*

De façon similaire au mécanisme *sci_élém* pour l'événement de scission, le mécanisme *fus_élém* intervient lorsqu'une fusion de cellules est décelée suite à l'application d'une opération élémentaire. Contrairement à la scission, la fusion se décline par le regroupement d'un ensemble de cellules c_1, c_2, \dots, c_n pour n'en former qu'une, nommée c , et de même dimension. Il convient alors de caractériser c par un nouvel identifiant id et de propager les ancêtres de c_1, c_2, \dots, c_n à c .

Le mécanisme *fus_élém*, décrit par l'algorithme 3, permet d'ajouter deux nouvelles relations liant l'identifiant de c avec ses origines et avec sa dimension. Ce mécanisme prend en paramètres les identifiants id_1, id_2, \dots, id_n des cellules à fusionner, un identifiant unique id de la cellule résultant de la fusion. Les points 1 à 3 représentent les pré-conditions au bon fonctionnement de l'algorithme dont les points 4 à 6 constituent les étapes.

1. Les cellules c_1, c_2, \dots, c_n doivent obligatoirement exister dans l'instance courante du modèle paramétrique.
2. id doit être un identifiant, non encore utilisé dans le modèle topologique considéré, afin de garantir son unicité.

Algorithme. 2 : mécanisme créa_élém

Données : ■ id : un identifiant ;
 ■ dim : dimension de la cellule nouvellement créée.

Résultat : ajout des liens *ancêtre* et *dimension* relatifs à c .

début

 si $\exists id, dim$

tels que

1 | $id \in ID_Cellule - ID_utilisés$ et

2 | $dim \in [0, Dim_Max]$

alors

3 | $ID_utilisés \leftarrow ID_utilisés \cup \{id\}$;

4 | $ancêtre \leftarrow ancêtre \cup \{(id, \{\})\}$;

fin

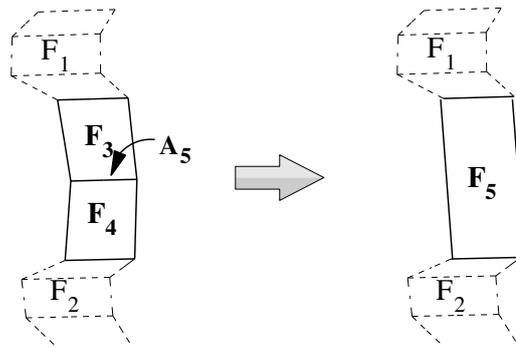


Figure 3.7: Fusion des faces F_3 et F_4 du modèle de la figure 3.3.

3. Les cellules c_1, c_2, \dots, c_n doivent être de même dimension.
4. Adjoindre les ancêtres de c_1, c_2, \dots, c_n ainsi que leurs dimension à c .

3.2.1.4 Mécanisme *sup_élém*

Une suppression de cellule entraîne la perte des informations qui lui sont associées : l'identifiant et la dimension des cellules d'une part, l'ensemble des identifiants caractérisant l'ancêtre des cellules d'autre part. Lorsqu'on souhaite reporter cet événement dans le journal de bord, il faut garder une trace de cet état de fait. Nous distinguons deux types de suppressions de cellules :

1. suppression des cellules présentes au début de l'opération globale : ce type d'événement doit être reporté au niveau du journal de bord car seuls les événements liant les cellules à l'entrée et à la sortie de l'opération globale sont ajoutés ;
2. suppression des cellules apparues au cours de l'opération globale : aucun traitement particulier n'intervient.

3.2. ÉVOLUTION DE LA TOPOLOGIE AU SEIN DU NOYAU GÉOMÉTRIQUE

Algorithme. 3 : mécanisme fus_élé

Données : ■ $id, id_1, id_2, \dots, id_n$: n identifiants ;

■ $ens_id_1, ens_id_2, \dots, ens_id_n$: n ensembles d'identifiants qui caractérisent l'origine des cellules c_1, c_2, \dots et c_n .

Résultat : ajout des liens *ancêtre* et *dimension* relatifs aux cellules c_1, c_2, \dots, c_n .

début

```

  si  $\exists id, id_1, id_2, \dots, id_n$ 
  tels que
1   |  $\forall i \in [1, n], id_i \in D_{ancêtre}$  et
2   |  $id \in ID\_Cellule - ID\_utilisés$  et
3   |  $\forall i, j \in \{1, 2, \dots, n\}, dimension(id_i) = dimension(id_j)$ 
    |
    |
4   | alors
    |    $ID\_utilisés \leftarrow ID\_utilisés \cup \{id\}$ ;
    |    $ancêtre \leftarrow ancêtre \cup \{(id, ancêtre(id_1)) \cup ancêtre(id_2) \cup \dots \cup ancêtre(id_n)\}$ ;
    |    $dimension \leftarrow dimension \cup \{(id, dimension(id_1))\}$ ;
  fin

```

Le mécanisme *sup_élé*, décrit par l'algorithme 4, prend en paramètre l'identifiant id caractérisant c . Il fait en sorte de supprimer la relation qui associe c , à travers id , avec son ancêtre et sa dimension (voir le point 2 de l'algorithme).

Algorithme. 4 : mécanisme sup_élé

Données : ■ id : un identifiant ;

Résultat : suppression de la relation liant la cellule caractérisée par l'identifiant id avec son ancêtre et sa dimension.

début

```

  si  $\exists id$ 
  tel que
1   |  $id \in D_{ancêtre}$ 
    |
    |
2   | alors
3   |    $ancêtre \leftarrow ancêtre - \{(id, ancêtre(id))\}$ ;
    |    $dimension \leftarrow dimension - \{(id, dimension(id))\}$ ;
  fin

```

3.2.1.5 Mécanisme *modif_élé*

La modification de la topologie d'une i -cellule c apparaît lorsque le bord de c (ensemble des cellules de dimension $j < i$) se transforme sans causer la scission ou la fusion de cette dernière. Il convient alors de caractériser c par un nouvel identifiant et de garder tel quel son ancêtre.

Algorithme. 5 : mécanisme *modif_élém*

Données : id, id' : 2 identifiants.

Résultat : ajout des liens *ancêtre* et *dimension* relatif à la cellule dont la topologie est modifiée.

début

si $\exists id, id'$

tels que

1 | $id \in D_{ancêtre}$ **et**

2 | $id' \in ID_{Cellule} - ID_{utilisés}$ **et**

alors

3 | $ID_{utilisés} \leftarrow ID_{utilisés} \cup \{id'\};$

 | $ancêtre \leftarrow ancêtre \cup \{(id', ancêtre(id))\};$

 | $dimension \leftarrow dimension \cup \{(id', dimension(id))\};$

fin

Le mécanisme *modif_élém* décrit par l'algorithme 5, permet de mettre à jour l'identifiant de c avec une nouvelle valeur. Il associe alors l'ancêtre de c avec ce nouvel identifiant. Ce mécanisme prend en paramètres l'identifiant id caractérisant c avant sa modification et l'identifiant id' caractérisant c après cette modification. Les points 1 et 2 représentent les pré-conditions au bon fonctionnement de l'algorithme et le point 3 constitue son corps.

1. La cellule c doit obligatoirement exister dans l'instance courante du modèle paramétrique.
2. id' doit être un identifiant, non encore utilisé dans le modèle topologique considéré, afin de garantir son unicité.
3. Adjoindre l'ancêtre de c avant sa modification ainsi que sa dimension à c après cette modification.

3.2.2 Génération du journal de bord

Après avoir associé à chaque identifiant id (et donc à chaque cellule c du modèle géométrique) un ensemble d'identifiants ens_{id} (resp. une dimension dim) caractérisant l'ancêtre (resp. la dimension) de c , un parcours de toutes les cellules du modèle est entrepris *a posteriori* (c'est-à-dire à la fin d'une *opération globale*). Ce parcours a pour objectif la récupération de l'intégralité des associations reliant, d'une part, chaque cellule à son ancêtre et, d'autre part, chaque cellule à sa dimension topologique dim . Ces relations doivent être interprétées et structurées au sein du journal de bord, de manière à pouvoir distinguer les événements survenus sur les cellules en entrée et en sortie d'une opération globale.

De manière évidente, dans le cas d'une suppression de cellules, la récupération des associations que nous venons de décrire ne peut être réalisée *a posteriori*. En effet, la suppression d'une cellule implique la perte des informations qui lui sont associées. Pour pallier à cette perte, nous proposons, dans un premier temps, d'effectuer un parcours au début de toute opération globale. Durant ce parcours, les identifiants contenus dans les ensembles ens_{d_i} sont regroupés dans un premier ensemble D_1 . À la fin de cette opération, un second parcours est effectué afin de récupérer

3.2. ÉVOLUTION DE LA TOPOLOGIE AU SEIN DU NOYAU GÉOMÉTRIQUE

puis regrouper au sein d'un second ensemble, D_2 , les identifiants caractérisant les ancêtres des cellules restantes (identifiants contenus dans les ensembles ens_id_i associés aux identifiants des cellules restantes). Cela permet la détermination des cellules supprimées en calculant la différence entre ces deux ensembles comme suit $D = D_1 - D_2$ puis en déduisant l'ensemble Z comme suit :

$$Z = \{(\{x\}, \{ \}) | x \in D\}$$

Pour étayer ces propos, considérons l'exemple de la figure 3.7. Dans cet exemple, l'arête A_5 , incidente aux faces F_3 et F_4 , est supprimée suite à l'application de l'opération élémentaire de fusion de faces⁶. De ce fait, les identifiants id_{F_3} , id_{F_4} et id_{A_5} caractérisant respectivement les cellules F_3 , F_4 et A_5 sont regroupés dans D_1 . Ensuite, les identifiants id_{F_3} et id_{F_4} caractérisant les ancêtres de la face F_5 sont regroupés dans D_2 . Le calcul de l'ensemble $D = \{id_{A_5}\}$ permet de déduire l'ensemble $Z = \{(\{id_{A_5}\}, \{ \})\}$ et donc de déceler la suppression de l'arête A_5 .

Dans le cadre du formalisme défini dans la section 3.2.1, le parcours des cellules peut être assimilé au parcours de l'ensemble E défini par :

$$E = \{(id, ancêtre(id)) | id \in D_{ancêtre}\}$$

Certains éléments de E ayant le même ancêtre (en d'autres termes les cellules résultant d'une scission), sont regroupés en n sous-ensembles (n peut être nul). Chaque sous-ensemble, noté S_{E_k} ($k \in [0, n]$), contient les éléments x de E ($x = (id, ens_id)$) ayant le même ens_id (seconde partie du couple égal à x). D'un point de vue formel, S_{E_k} est défini par :

$$S_{E_k} = \{\forall x_i, x_j | (x_i, x_j) \in S_{E_k}^2 \Rightarrow (x_i, x_j) \in E^2 \text{ et } x_i = (id_i, set_id_i), x_j = (id_j, set_id_j), \\ set_id_i = set_id_j \text{ et } set_id_i \neq \emptyset\}$$

Ainsi, dans le cas de la table 3.2 (qui regroupe les éléments de l'ensemble E déduit de la figure 3.6), le calcul des sous-ensembles S_{E_k} donne le résultat suivant : $S_{E_0} = \{(id_{F_1}, \{id_F\}), (id_{F_2}, \{id_F\})\}$, $S_{E_1} = \{(id_{A_{1.1}}, \{id_{A_1}\}), (id_{A_{1.2}}, \{id_{A_1}\})\}$ et $S_{E_2} = \{(id_{A_{2.1}}, \{id_{A_2}\}), (id_{A_{2.2}}, \{id_{A_2}\})\}$.

À partir des sous-ensembles S_{E_k} , nous pouvons extraire les événements de scission en définissant un nouvel ensemble noté S . Chaque élément $(A_k, B_k) \in S$ associe les cellules objets de la scission (A_k) à ceux résultant de cette même scission (B_k) tels que :

$$S = \{x_k = (A_k, B_k) | k \in [0, n] \text{ et } \forall x = (id, ens_id) | x \in S_{E_k} \Rightarrow id \in B_k \text{ et } A_k = ens_id\}$$

Donc, cet ensemble regroupe, sous forme de couples, tous les événements de scission survenus à une étape du processus de modélisation paramétrique. Par exemple, l'ensemble S construit à partir des sous-ensembles S_{E_0} , S_{E_1} et S_{E_2} regroupe les événements de scission $(A_0, B_0) = (\{id_F\}, \{id_{F_1}, id_{F_2}\})$, $(A_1, B_1) = (\{id_{A_1}\}, \{id_{A_{1.1}}, id_{A_{1.2}}\})$ et $(A_2, B_2) = (\{id_{A_2}\}, \{id_{A_{2.1}}, id_{A_{2.2}}\})$. (A_0, B_0) , (A_1, B_1) et (A_2, B_2) traduisent alors respectivement la scission de F , A_1 et A_2 en deux nouvelles faces F_1, F_2 , deux nouvelles arêtes $A_{1.1}, A_{1.2}$ et deux autres arêtes $A_{2.1}, A_{2.2}$.

6. Malgré le fait que l'opération élémentaire de fusion de faces porte le même nom que l'événement de fusion de cellules, ces deux notions sont différentes. En effet, en plus de l'événement de fusion, cette opération génère également un événement de suppression de l'arête incidente aux deux faces à fusionner.

Cellule	Identifiant id_i	Ensemble d'identifiants (set_id_i)	Dimension
F	id_F	$\{id_F\}$	2
A_1	id_{A_1}	$\{id_{A_1}\}$	1
A_2	id_{A_2}	$\{id_{A_2}\}$	1
F_1	id_{F_1}	$\{id_F\}$	2
F_2	id_{F_2}	$\{id_F\}$	2
A_3	id_{A_3}	$\{\}$	1
A_4	id_{A_4}	$\{\}$	1
$A_{1.1}$	$id_{A_{1.1}}$	$\{id_{A_1}\}$	1
$A_{1.2}$	$id_{A_{1.2}}$	$\{id_{A_1}\}$	1
$A_{2.1}$	$id_{A_{2.1}}$	$\{id_{A_2}\}$	1
$A_{2.2}$	$id_{A_{2.2}}$	$\{id_{A_2}\}$	1
S_1	id_{S_1}	$\{\}$	0
S_2	id_{S_2}	$\{\}$	0
S_3	id_{S_3}	$\{\}$	0
S_4	id_{S_4}	$\{\}$	0

Table 3.2: Informations associées à quelques cellules du modèle de la figure 3.6.

Pour tous les autres cas (c'est-à-dire, les événements de création et de fusion), nous définissons l'ensemble A comme suit :

$$A = \{\forall x | x \in E - \bigcup_{k=0}^n S_{E_k} \Rightarrow x = (id, set_id) \text{ et } (set_id, \{id\}) \in A\}$$

Ainsi, pour la figure 3.3, l'ensemble $E - \bigcup_{k=0}^n S_{E_k}$ regroupe les éléments $(id_{A_3}, \{\}), (id_{A_4}, \{\}), (id_{S_1}, \{\}), (id_{S_2}, \{\}), (id_{S_3}, \{\})$ et $(id_{S_4}, \{\})$. Il en résulte la composition suivante de l'ensemble A : $(\{\}, \{id_{A_3}\}), (\{\}, \{id_{A_4}\}), (\{\}, \{id_{S_1}\}), (\{\}, \{id_{S_2}\}), (\{\}, \{id_{S_3}\})$ et $(\{\}, \{id_{S_4}\})$ qui traduit les événements de création des arêtes A_3 et A_4 ainsi que les sommets S_1, S_2, S_3 et S_4 .

Finalement, chaque ligne du journal de bord correspond à un élément de l'ensemble JB défini par :

$$JB = S \cup Z \cup A$$

En effet, pour chaque couple $x = (e_1, e_2)$ appartenant à JB , nous répartissons e_1 (premier élément de x) dans la case des cellules de départ et e_2 (second élément de x) dans la case des cellules d'arrivée de la structure de journal de bord. Donc, à l'issue de l'opération globale de la figure 3.3, les éléments de l'ensemble S sont répartis sur les lignes 1 à 3 du journal de bord représenté par la table 3.3. Ceux de l'ensemble A (du moins pour l'événement de création) sont répartis sur les lignes 4 et 5 et l'ensemble Z ne comprend aucun élément pour cet exemple.

Après avoir réparti les éléments de l'ensemble JB dans le journal de bord, nous passons à la prochaine étape de construction. Cette étape consiste à identifier les événements (création, suppression, scission, fusion, modification) mettant en relation les cellules d'une même ligne dans cette structure.

3.3. LES OPÉRATIONS DE BASE

Entités de départ	Entités d'arrivée	Type d'événement
id_F	$\{id_{F_1}, id_{F_2}\}$	scission
id_{A_1}	$\{id_{A_{1.1}}, id_{A_{1.2}}\}$	scission
id_{A_2}	$\{id_{A_{2.1}}, id_{A_{2.2}}\}$	scission
...
$\{\}$	$\{id_{S_1}\}$	création
$\{\}$	$\{id_{S_2}\}$	création

Table 3.3: Journal de bord généré suite à l'application de l'opération booléenne de la figure 3.3.

Comme nous pouvons le constater, le critère utilisé pour l'interprétation des événements est la cardinalité de l'ensemble des Entités de Départ ($\text{card}(ED)$) et celles d'Arrivée ($\text{card}(EA)$) :

1. si ED est un ensemble vide alors l'événement correspondant est une création (voir la ligne 4 de la table 3.3) ;
2. si EA est un ensemble vide alors l'événement correspondant est une suppression ;
3. si $\text{card}(ED) = 1$ et $\text{card}(EA) > 1$ alors l'événement correspondant est une scission (voir la ligne 1 de la table 3.3) ;
4. si $\text{card}(EA) = 1$ et $\text{card}(ED) > 1$ alors l'événement correspondant est une fusion ;
5. si $\text{card}(EA) = 1$, $\text{card}(ED) = 1$ et $EA \neq ED$ alors l'événement correspondant est une modification.

3.3 Les opérations de base

Pour modifier une G-carte, 4 opérations de base ont été proposé : la suppression, la contraction, l'insertion et l'éclatement. Les deux premières définies par Damiand et al. [DL03] permettent la construction d'une G-carte par diminution du nombre de cellules. Les deux dernières que nous définissons permettent la construction d'une G-carte par ajout de cellules. Nous exploitons pour ce faire le fait que l'insertion et l'éclatement sont l'inverse de la suppression et la contraction. De plus, comme dans le cas de la suppression et de la contraction, l'opération d'éclatement est définie par dualité⁷ à l'opération d'insertion (voir la figure 3.8). Pour chaque opération, nous avons privilégié la généralité en définissant ces opérations pour des cellules quelconques, la seule contrainte étant que l'application de l'opération préserve les contraintes des cartes généralisées. Selon les applications, il est ensuite souvent nécessaire d'ajouter des contraintes supplémentaires afin de préserver des propriétés spécifiques comme par exemple la connexité. Mais ajouter ces contraintes se fait simplement en testant des nouvelles propriétés avant d'appliquer l'opération qui elle reste inchangée. L'avantage de cette approche est de fournir une opération générique pouvant servir dans n'importe quel cadre.

3.3.1 Quelques notions supplémentaires autour des cartes généralisées

En plus de la définition que nous avons donné d'une carte généralisée (voir la section 2.1.2.2, page 17) et celle que nous avons donné d'une cellule (élément principal de la subdivision spatiale

7. De façon informelle, le dual d'une subdivision est une subdivision du même espace, dans laquelle chaque $(n - i)$ -cellule est associée à chaque i -cellule initiale tout en conservant les relations d'incidence.

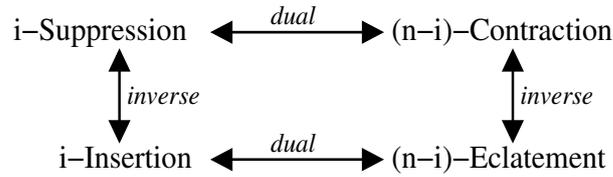


Figure 3.8: Liens entre les 4 opérations de base : suppression/contraction et insertion/éclatement.

qu'elle décrit), nous présentons quelques éléments supplémentaires nécessaires à la définition des opérations de base. Aussi, nous abordons la notion de degré local d'une cellule et la notion de cellule pendante.

Le *degré inférieur* d'une i -cellule correspond au nombre de $(i - 1)$ -cellules distinctes qui lui sont incidentes, et le *degré supérieur* (ou tout simplement *degré*) correspond au nombre de $(i + 1)$ -cellules distinctes qui lui sont incidentes. Ainsi, sur la figure 3.9(a), la face f est de degré inférieur 1 car elle est incidente à une seule arête, et le sommet s est de degré supérieur 2 car il est incident à deux arêtes.

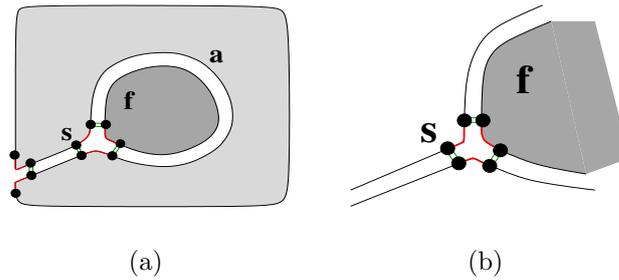


Figure 3.9: Degré et degré local d'une cellule (a) Exemple de G-carte composée de deux faces, trois arêtes et deux sommets. (b) « Zoom » local autour du sommet s . Le degré local supérieur de s est trois.

La notion de degré local permet d'affiner la notion de degré lorsque des cellules sont *multi-incidentes*, *i.e.* incidentes plusieurs fois entre elles. En pratique, pour les cartes généralisées, le degré local (supérieur ou inférieur) d'une cellule C est calculé localement en parcourant les brins de C (voir la figure 3.9(b)), alors que le calcul du degré (supérieur ou inférieur) de C nécessite de parcourir les brins de C et ceux des cellules incidentes, afin de détecter les éventuelles cellules multi-incidentes.

Sur la figure 3.9, l'arête a , deux fois incidente au sommet s , est considérée de manière locale comme deux arêtes distinctes incidentes à s . Ainsi, le degré local supérieur de s est 3, alors que son degré supérieur est 2.

Définition 3.1 (Degré local deux)

Soit $i, 2 \leq i \leq n$, une i -cellule C est de degré local inférieur à deux si

$$\forall b \in C, \quad b\alpha_{i-1}\alpha_{i-2} = b\alpha_{i-2}\alpha_{i-1}$$

De même, soit $i, 0 \leq i \leq n - 2$, une i -cellule C est de degré local supérieur deux (ou tout simplement degré local deux) si

$$\forall b \in C, \quad b\alpha_{i+1}\alpha_{i+2} = b\alpha_{i+2}\alpha_{i+1}$$

Intuitivement, une cellule est pendante si elle est reliée à une autre par une partie de son bord. De façon plus précise, une i -cellule C est pendante si :

- elle est de degré supérieur 1, c'est-à-dire, incidente deux fois à la même $(i + 1)$ -cellule ;
- dans l'ensemble des $(i - 1)$ -cellules incidentes à C , au moins une est de degré supérieur 1 mais pas toutes (voir l'exemple de la figure 3.10) ;

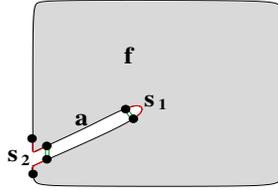


Figure 3.10: Exemple d'une arête pendante a . Le sommet s_1 est de degré supérieur 1 alors que le sommet s_2 est de degré supérieur 2.

- l'ensemble des $(i - 1)$ -cellules incidentes à C et de degré supérieur 1, noté D , est « connexe » (c'est-à-dire qu'entre chaque couple d'éléments (a, b) de D , il existe un chemin de cellules, appartenant exclusivement à D , qui permet de passer de a vers b).

Il en découle la définition suivante d'une cellule pendante.

Définition 3.2 (Cellule pendante)

Pour tout $i, 1 \leq i \leq n$, soient $N = \{0, 1, \dots, n\}$, $C = \langle \rangle_{N-\{i\}} (b)$ une i -cellule et $D = \{C' = \langle \rangle_{N-\{i-1\}} (b') \mid b' \in C \text{ et } \text{degré}(C') = 1\}$ l'ensemble de $(i - 1)$ -cellules incidentes à C et de degré supérieur 1. C est pendante si :

1. C est de degré local supérieur 2 et de degré supérieur 1 ;
2. $D \neq \emptyset$ et $D \neq C$;
3. $\forall a, b \in D$, il existe une suite de cellules $d_1 = a, \dots, d_k = b$ telle que chaque couple de cellules (d_i, d_{i+1}) soit constitué d'éléments adjacents.

3.3.2 Suppression et contraction

Comme son nom l'indique, dans un espace de dimension n , la *suppression* d'une i -cellule se déploie en supprimant une cellule de dimension i , et fusionnant, éventuellement, les deux $(i + 1)$ -cellules qui lui sont incidentes (voir l'exemple des figures 3.11 et 3.12).

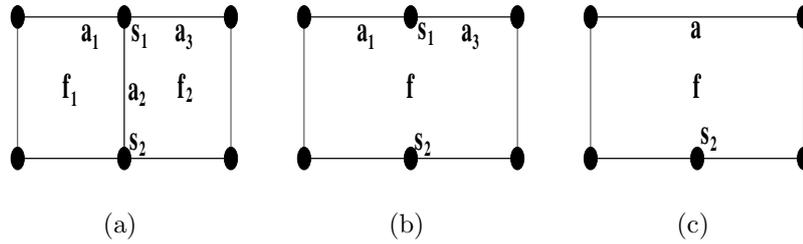


Figure 3.11: Exemples de suppressions en $2D$. (a) Une subdivision $2D$ initiale. (b) Le résultat après la suppression de l'arête a_2 . Les deux faces f_1 et f_2 sont fusionnées en une seule face f . (c) Le résultat après la suppression du sommet s_1 : les deux arêtes a_1 et a_3 sont fusionnées en une seule arête a . La suppression de s_1 n'était pas possible à partir de la subdivision initiale car son degré supérieur était égal à 3.

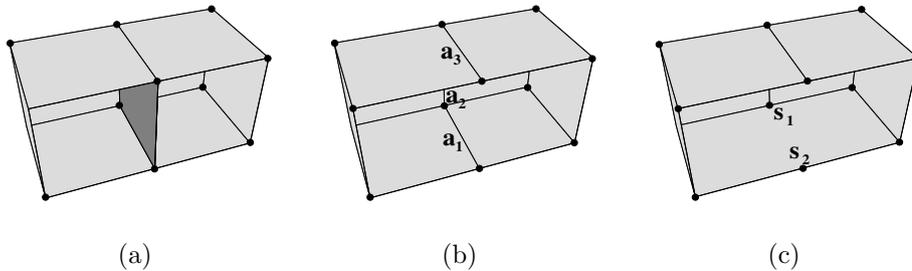


Figure 3.12: Exemples de suppressions en $3D$. (a) Une subdivision $3D$ initiale. (b) Le résultat après la suppression de la face gris foncé. (c) Le résultat après la suppression de l'arête a_1 .

Par ailleurs, la *contraction* d'une i -cellule est l'opération duale de la suppression. Elle consiste à contracter cette cellule pour générer une $(i - 1)$ -cellule (voir la figure 3.13). Dans un espace de dimension n , la suppression est définie pour les cellules de dimensions 0 à $n - 1$, tandis que la contraction est définie pour les cellules de dimensions 1 à n . Dans le cas des cartes généralisées, [DL03] définissent ces deux opérations comme indiqué dans la suite de cette section.

3.3.2.1 Suppression

Pour supprimer une i -cellule c dans une n -G-carte, une précondition doit être vérifiée : il est nécessaire de vérifier que son degré local supérieur soit égal à deux. En effet, il n'est pas envisageable de considérer cette opération autrement, car au-delà de deux, il existe plusieurs façons de recoller les cellules incidentes à c . Partant de là, il suffit de redéfinir les involutions α_i des brins i -voisins de c (notés BV). Cette redéfinition est locale et elle nécessite uniquement un parcours des brins de c .

Lors de la redéfinition des α_i pour les brins de BV , nous utilisons un chemin de brins reposant sur $(\alpha_i \alpha_{i+1})^k \alpha_i$. Ce chemin permet, à partir d'un brin $b' \in BV$, de rentrer dans la cellule supprimée (en utilisant α_i) puis de traverser cette cellule (en utilisant $\alpha_i \alpha_{i+1}$). Il

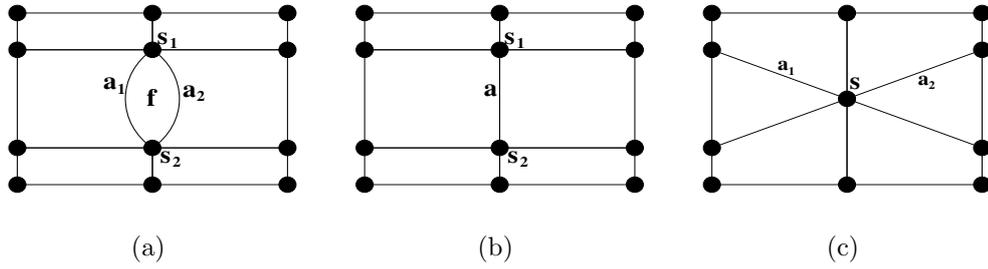


Figure 3.13: Exemples de contractions en $2D$. (a) Une subdivision $2D$ initiale. (b) Le résultat après la contraction de la face f . Les deux arêtes a_1 et a_2 ont fusionné en une seule arête a . (c) Le résultat après la contraction de l'arête a : les deux sommets s_1 et s_2 ont fusionné en un seul sommet s .

faut éventuellement continuer ce parcours (via l'exposant k) afin de traiter le cas où la cellule est repliée sur elle-même (cas des multi-incidences illustrées par l'exemple de la boucle de la figure 3.14).

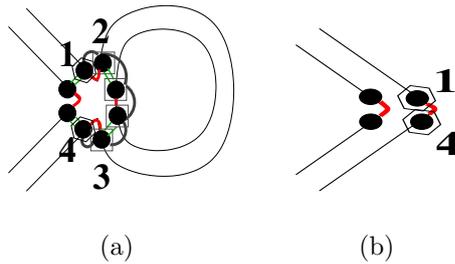


Figure 3.14: 1-suppression en $2D$ dans un cas de multi-incidence : ici une boucle. (a) La 2-G-carte initiale. (b) Le résultat après la 1-suppression. Par exemple, $1\alpha'_1 = 1(\alpha_1\alpha_2)(\alpha_1\alpha_2)\alpha_1 = 4 \in BV$ (car $1(\alpha_1\alpha_2)\alpha_1 \notin BV$, ce brin appartient à C et à $C\alpha_1$).

Remarquons que pour les brins $b' \in B' - BV$, $b'\alpha'_i = b'(\alpha_i\alpha_{i+1})^k\alpha_i$ (k est le plus petit entier tel que $b'\alpha'_i \in BV$ et peut être égal à zéro lorsque $b'\alpha_i \in BV$). Remarquons également que la précondition de l'opération de suppression ne s'applique pas pour les cellules de dimension $i = n - 1$ (car l'involution $\alpha_{i+2} = \alpha_{n+1}$ n'est pas définie). Par ailleurs, une $(n - 1)$ -cellule peut toujours être supprimée dans une n -G-carte. En effet, les cartes généralisées représentent des quasi-variétés cellulaires et une $(n - 1)$ -cellule est toujours de degré local supérieur deux dans une n -G-carte. Dès lors, la définition formelle d'une i -suppression dans une n -G-carte peut être énoncée comme suit :

Définition 3.3 (*i*-suppression)

Soient $G = (B, \alpha_0, \dots, \alpha_n)$ une n - G -carte, $i \in \{0, \dots, n-1\}$, $C = \langle \rangle_{\{1,2,\dots,n\}-\{i\}}$ (b) une i -cellule de G vérifiant que son degré local supérieur est égal à deux (c'est-à-dire, $\forall b' \in C, b'\alpha_{i+1}\alpha_{i+2} = b'\alpha_{i+2}\alpha_{i+1}$) et $BV = C\alpha_i - C$, l'ensemble des brins i -cousus à C n'appartenant pas à C . La n - G -carte obtenue en supprimant C de G est $G' = (B', \alpha'_0, \dots, \alpha'_n)$ définie par :

- $B' = B - C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \alpha'_j = \alpha_j|_{B'}$;⁸
- $\forall b' \in B' - BV, b'\alpha'_i = b'\alpha_i$;
- $\forall b' \in BV, b'\alpha'_i = b'(\alpha_i\alpha_{i+1})^k \alpha_i$, avec k le plus petit entier tel que $b'(\alpha_i\alpha_{i+1})^k \alpha_i \in BV$.

Prenons un exemple d'application de l'opération de suppression et regardons la conséquence du non respect de sa précondition. Pour ce faire, considérons l'exemple de la figure 3.11(a) et analysons la suppression du sommet s_1 (cellule de degré local 3). Afin de mener à bien la suppression de ce sommet, il faut choisir le couple d'arêtes à fusionner (la troisième devenant alors une arête pendante). Ceci pose deux problèmes :

1. le paramètre d'entrée de l'opération : la donnée de la cellule à supprimer ne suffit pas ;
2. le résultat de l'opération qui est peu satisfaisant : l'exemple de la figure 3.11(a) (suppression du sommet s_1), entraîne la création d'une arête pendante et une déconnexion autour du sommet.

La précondition de l'opération de suppression permet de résoudre ces problèmes, sans limiter les possibilités offertes à l'utilisateur. Ainsi, il est toujours possible de transformer le degré local supérieur de la cellule pour qu'il soit égal à deux. Une opération de plus haut niveau intervient alors et modifie la n - G -carte pour atteindre cet objectif.

Nous pouvons voir différents cas de suppressions dans les figures 3.15 à 3.14. Tout d'abord, un exemple de 0-suppression en $1D$ (figure 3.15), puis de 0-suppression en $2D$ (figure 3.16), de 1-suppression en $2D$ (figure 3.17) et enfin un cas de multi-incidence lorsque $k = 2$ (figure 3.14).

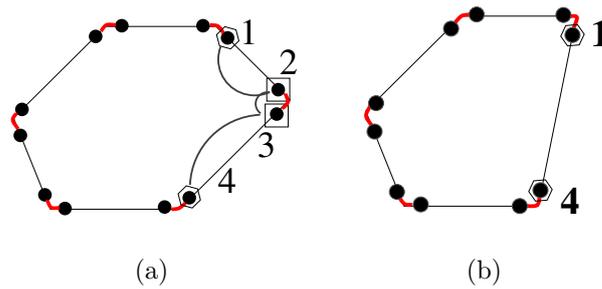


Figure 3.15: 0-suppression en $1D$. (a) La 1- G -carte initiale. (b) Le résultat après la 0-suppression. $C = \langle \alpha_1 \rangle (2) = \{2, 3\}$ (les brins marqués avec des carrés), $C\alpha_0 = \{1, 4\} = BV$ (les brins marqués avec des pentagones). La 0-suppression consiste simplement à définir $1\alpha'_0 = 1(\alpha_0\alpha_1)\alpha_0 = 4 \in BV$ et $4\alpha'_0 = 4(\alpha_0\alpha_1)\alpha_0 = 1 \in BV$.

8. α'_j est égal à α_j restreint à B' , i.e. $\forall b \in B', b\alpha'_j = b\alpha_j$.

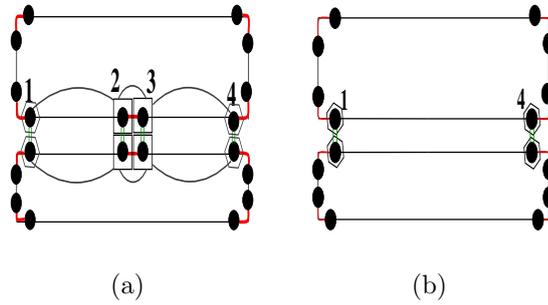


Figure 3.16: 0-suppression en $2D$. (a) La 1-G-carte initiale. (b) Le résultat après la 0-suppression. $C = \langle \alpha_1, \alpha_2 \rangle$ (les brins marqués avec des carrés vides), $C\alpha_0 = BV$ (les brins marqués avec des pentagones). Par exemple, $1\alpha'_0 = 1(\alpha_0\alpha_1)\alpha_0 = 4 \in BV$.

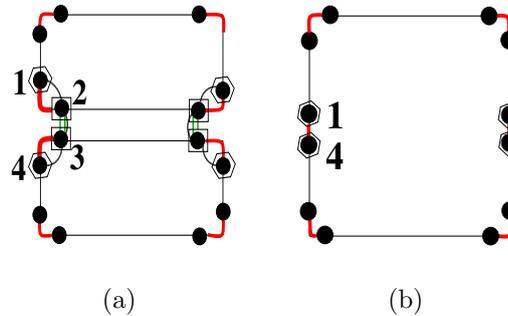


Figure 3.17: Cas général de la 1-suppression en $2D$. Les brins de l'arête à supprimer sont marqués avec des carrés. (a) La 2-G-carte initiale. (b) Le résultat après la 1-suppression.

L'application de l'opération de suppression peut avoir des conséquences sur les propriétés topologiques (par exemple la propriété d'Euler-Poincaré⁹) des objets modélisés. Il est vrai que dans le cas général, lorsque la i -cellule c à supprimer est incidente à deux $(i+1)$ -cellules distinctes c_1 et c_2 , et qu'aucune $(i-1)$ -cellule incidente à c n'est de degré supérieur 1, la suppression entraîne uniquement la disparition de c et la fusion de c_1 et c_2 . De ce fait, la caractéristique d'Euler-Poincaré généralisée reste identique car le nombre de cellules de dimension i et celles de dimension $i-1$ diminue chacun d'une seule unité. En revanche, lorsque c est de degré supérieur 1, plusieurs cas peuvent se produire, conduisant ou non à la modification de la caractéristique d'Euler. La suppression de la face f (de degré supérieur 1) dans le volume de la figure 3.18(a) en est une parfaite illustration. Dans cet exemple, la suppression de f entraîne la création d'un tore (voir la figure 3.18(b)) mais aucune autre disparition de cellule.

Dans tous les cas, même s'il y a modification des propriétés topologiques au niveau des objets modélisés, l'opération de suppression reste valide (le résultat de l'opération est toujours une n -

9. En général, la caractéristique d'Euler peut être définie, en dimension n , comme la somme alternée du nombre de cellules de dimension i , noté k_i , comme suit : $\chi = \sum_{i=0}^n (-1)^i k_i$.

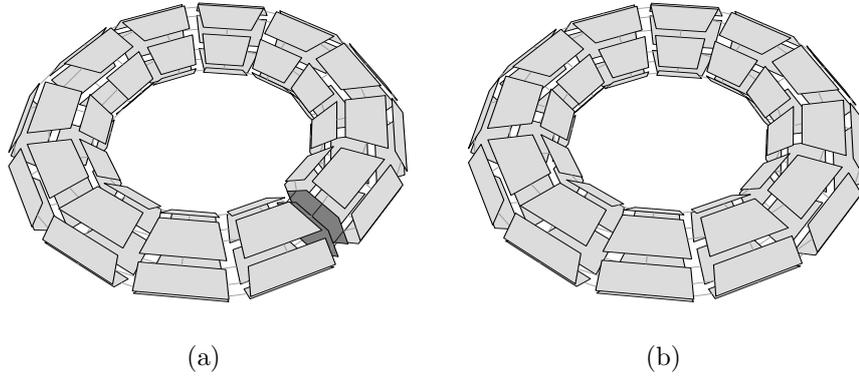


Figure 3.18: Exemple de 2-suppression d'une face incidente deux fois au même volume. (a) La subdivision initiale : la face à supprimer est dessinée en gris foncé. (b) La subdivision obtenue après la suppression qui a entraîné une modification topologique : la création d'un tore.

G-carte). De surcroît, nous pouvons rendre compte de cette modification par le calcul d'un journal de bord. Cependant, si la suppression entraîne une modification topologique majeure (déconnexion en plusieurs composantes connexes des objets traités, ou même la suppression complète de la carte généralisée), il revient aux applications qui prennent en charge ces cas spécifiques, d'ajouter des contraintes pour satisfaire à des besoins particuliers.

3.3.2.2 Contraction

L'opération de contraction est définie de la même façon que l'opération de suppression. Elle consiste à transformer une i -cellule en une $(i-1)$ -cellule. De plus, elle présente la propriété d'être la duale de l'opération de suppression. En d'autres termes, la (i) -contraction est l'opération duale de la $(n-i)$ -suppression. Il suffit alors de remplacer dans la définition de la i -suppression (définition 3.3) les indices i par $(n-i)$, $i+1$ par $(n-i-1)$ et $i+2$ par $(n-i-2)$. Ensuite, pour obtenir la définition de la i -contraction en lieu et place de la $(n-i)$ -contraction, il faut également remplacer $(n-i)$ par i , $(n-i-1)$ par $i-1$ et $(n-i-2)$ par $i-2$.

Définition 3.4 (i -contraction)

Soit $G = (B, \alpha_0, \dots, \alpha_n)$ une n -G-carte, $i \in \{1, \dots, n\}$ et $C = \langle \rangle_{\{1,2,\dots,n\}-\{i\}}$ (b) une i -cellule de G , vérifiant la précondition suivante : $\forall b' \in C, b' \alpha_{i-1} \alpha_{i-2} = b' \alpha_{i-2} \alpha_{i-1}$. Nous notons $BV = C \alpha_i - C$, l'ensemble des brins i -cousus à C n'appartenant pas à C . La n -G-carte obtenue en contractant C de G est $G' = (B', \alpha'_0, \dots, \alpha'_n)$ définie par :

- $B' = B - C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \alpha'_j = \alpha_j|_{B'}$;
- $\forall b' \in B' - BV, b' \alpha'_i = b' \alpha_i$;
- $\forall b' \in BV, b' \alpha'_i = b' (\alpha_i \alpha_{i-1})^k \alpha_i$, avec k le plus petit entier tel que $b' (\alpha_i \alpha_{i-1})^k \alpha_i \in BV$.

Dans les figures 3.19 et 3.20, nous voyons deux exemples de contractions : le premier montre celle d'une arête dans une 1-G-carte et le second d'une autre arête dans une 2-G-carte.

3.3. LES OPÉRATIONS DE BASE

Comme pour la suppression, la contraction d'une i -cellule c peut entraîner des modifications topologiques en fonction des cellules assujetties à cette opération. Elle peut également entraîner la suppression d'autres cellules incidentes à c . La différence avec l'opération de suppression, à proprement parler, est que par dualité, elle concerne les $(i + 1)$ -cellules incidentes à c et non plus les $(i - 1)$ -cellules.

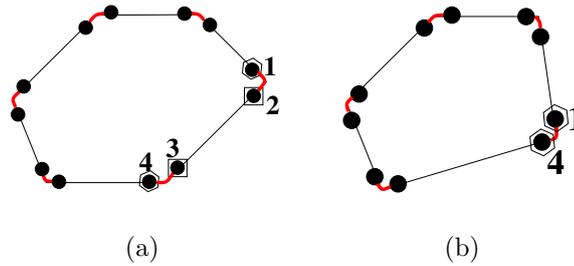


Figure 3.19: 1-contraction en $1D$. Les brins de l'arête contractée sont marqués avec des carrés. (a) La 2-G-carte initiale. (b) Le résultat après la 1-contraction.

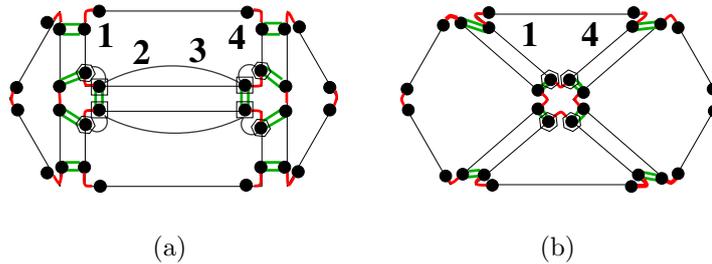


Figure 3.20: 1-contraction en $2D$. (a) La 2-G-carte initiale. (b) Le résultat après la 1-contraction.

3.3.2.3 Généralisation

Les deux opérations précédentes permettent de supprimer ou de contracter une seule cellule à la fois. Pour certaines applications, il peut être plus efficace d'appliquer simultanément plusieurs fois ces deux opérations. Pour cela, [DL03] proposent un marquage des cellules à supprimer et/ou à contracter doit d'abord être effectué. Des marques différentes pour chaque dimension de cellule et pour chaque type d'opération sont utilisées à cet effet. De même, les opérations de suppression et de contraction ne peuvent être appliquées que si et seulement si :

- les cellules sont disjointes deux à deux. Ceci garantit qu'un brin n'a qu'une seule marque, et donc qu'il n'y a pas d'ambiguïté lors de la redéfinition des involutions α_i , $0 \leq i \leq n$;
- chaque cellule vérifie les préconditions des opérations unitaires.

La G-carte résultante peut donc après être calculée de la façon suivante :

Définition 3.5 (*Suppression et contraction simultanées de cellules*)

Soient $G = (B, \alpha_0, \dots, \alpha_n)$ une n - G -carte, $\{S_i\}_{0 \leq i \leq n-1}$ les ensembles de i -cellules à supprimer, $\{C_i\}_{1 \leq i \leq n}$ les ensembles de i -cellules à contracter et $C_0 = S_n = \emptyset$. Posons $S = \bigcup_{i=0}^n S_i$ et $C = \bigcup_{i=0}^n C_i$. Les cellules de ces deux ensembles satisfont les préconditions suivantes :

- elles sont deux à deux disjointes (i. e. $\forall c, c' \in C \cup S, c \cap c' = \emptyset$);
- les cellules de S sont de degré local supérieur égal à 2;
- les cellules de C sont de degré local inférieur égal à 2;

Soit $BV_i = (S_i \cup C_i)\alpha_i - (S_i \cup C_i)$, $\forall i, 0 \leq i \leq n$ l'ensemble des brins « survivants » voisins des i -cellules à supprimer et/ou à contracter.

La n - G -carte résultant des suppressions et contractions simultanées des cellules appartenant à $S \cup C$, appelée $G' = (B', \alpha'_0, \dots, \alpha'_n)$, est définie par :

1. $B' = B - (C \cup S)$;
2. $\forall i, 0 \leq i \leq n, \forall b \in B' - BV_i, b\alpha'_i = b\alpha_i$;
3. $\forall i, 0 \leq i \leq n, \forall b \in BV_i, b\alpha'_i = b' = b(\alpha_i\alpha_{l_1}) \dots (\alpha_i\alpha_{l_r})\alpha_i$, où :
 - r est le plus petit entier tel que $b' \in BV_i$;
 - $\forall j, 1 \leq j \leq r, l_j = \begin{cases} i+1 & \text{si } b_j = b(\alpha_i\alpha_{l_1}) \dots (\alpha_i\alpha_{l_{j-1}})\alpha_i \in S_i, \\ i-1 & \text{sinon } (b_j \in C_i). \end{cases}$

La redéfinition des involutions α_i , reliant les brins *survivants*, repose sur un parcours guidé (chemin) des cellules à supprimer et/ou à contracter. Ainsi, pour chaque brin $b \in BV_i$, le parcours dont il est question débute par un autre brin $b' = b\alpha_i$. À ce stade, il y a deux possibilités. Soit

1. b' appartient à une cellule supprimée. Dans ce cas, le parcours itère sur le brin $b'\alpha_{i+1}\alpha_i$ sans oublier de mettre à jour b' avec cette valeur;
2. b' appartient à une cellule contractée. Dès lors, une mise à jour de b' est effectuée en utilisant le brin $b'\alpha_{i-1}\alpha_i$.

Au sortir de ces deux alternatives, un test est effectué pour savoir si b' (le nouveau brin) appartient ou non à l'ensemble BV_i . Dans l'affirmative, l'opération de suppression/contraction simultanée redéfinit α_i comme $\alpha'_i : b' = b\alpha'_i$. Dans le cas contraire, $b' \notin BV_i$, le processus est réitéré au niveau du choix entre les deux possibilités précédentes.

Cet algorithme ne peut produire une n - G -carte valide que si les cellules à supprimer et/ou à contracter sont disjointes. Cette précondition implique que les chemins parcourus à partir des brins $b \in BV_i, 0 \leq i \leq n$, sont constitués exclusivement de cellules de même dimension i . Ces mêmes chemins ne peuvent jamais être confondus lorsqu'ils traversent des cellules de dimensions différentes. Par voie de conséquence, cette propriété garantit le fait que la G -carte résultante est indépendante de l'ordre dans lequel sont redéfinis les α_i . Donc, pour des raisons pratiques, il est plus simple de modifier directement la n - G -carte initiale en mettant à jour ses involutions, sans avoir à la dupliquer.

Enfin, [DL03] montrent que la n - G -carte résultant de l'application simultanée d'un ensemble de contractions et de suppressions est la même que si l'on applique de manière consécutive chaque opération unitaire et ce dans n'importe quel ordre. Le principe de la démonstration repose sur la propriété des cellules disjointes, mais également sur le fait que le chemin traversant deux i -cellules adjacentes est la concaténation des deux chemins unitaires traversant chacune d'elles. De ce fait, la redéfinition simultanée d'un α'_i peut être ramenée à une suite de redéfinition successives de α'_i , une pour chaque opération unitaire.

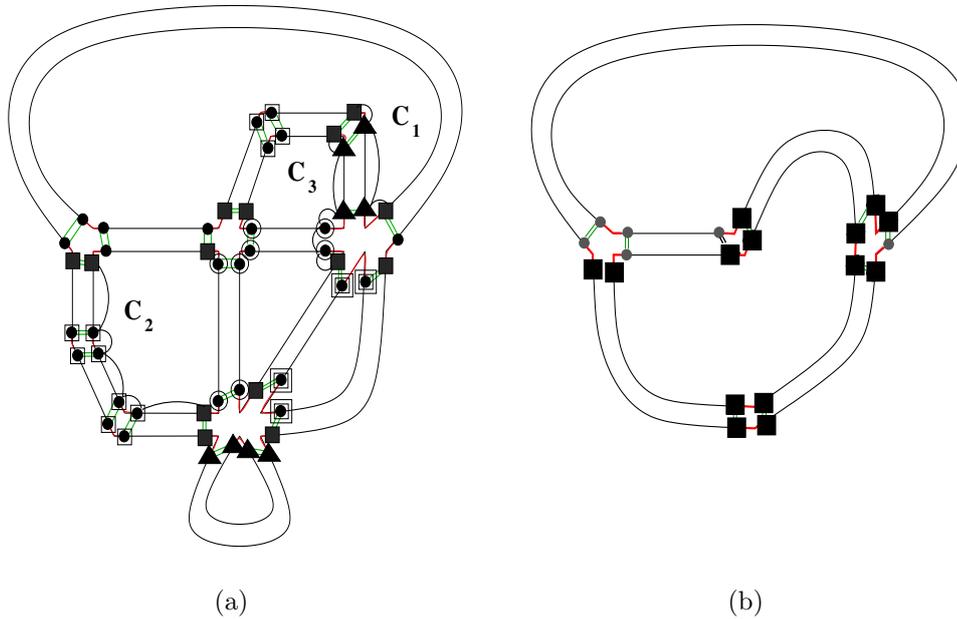


Figure 3.21: Un exemple 2D de suppression et contraction simultanées de cellules de différentes dimensions. (a) La 2-G-carte initiale. (b) La 2-G-carte obtenue après application de l'opération. Les brins appartenant à une 1-cellule supprimée (resp. 0-cellule supprimée, 1-cellule contractée, 2-cellule contractée) sont marqués avec un cercle (resp. un carré vide, un triangle plein, un double carrés). Les brins marqués avec des carrés pleins appartiennent à $\cup BV_i$. Trois chemins de connexion sont représentés : C_1 qui traverse une arête contractée, C_2 qui traverse deux sommets contractés, et C_3 qui traverse une arête contractée puis une arête supprimée.

Un exemple illustrant l'opération de contraction/suppression simultanée d'un ensemble de cellules est présenté dans la figure 3.21. Sur cet exemple, les quatre opérations existantes en dimension deux sont simultanément utilisées : la 1- et 2-contraction, et la 0- et 1-suppression.

3.3.3 Insertion et éclatement

Les deux autres opérations de base, que sont l'*insertion* et l'*éclatement* [BADSM08], sont les opérations inverses des opérations précédentes, suppression et contraction. Ces opérations permettent de raffiner une subdivision en y ajoutant des cellules.

Des exemples d'insertion et d'éclatement sont montrés dans les figures 3.22, 3.23 et 3.24. De prime abord, une comparaison entre ces exemples et ceux des opérations de suppression et de contraction permet de constater que les paramètres d'entrées d'une insertion ou d'un éclatement sont plus complexes.

En effet, pour les deux opérations présentées dans la section 3.3.2, la donnée seule de la cellule suffit. Les modifications sont guidées par la configuration des cellules voisines déduites par les involutions de la G-carte. Pour l'insertion et l'éclatement, la donnée seule d'une cellule ne suffit plus car il y a plusieurs manières d'ajouter cette cellule à la subdivision existante : il faut également indiquer à l'opération la façon de relier cette cellule à la G-carte existante.

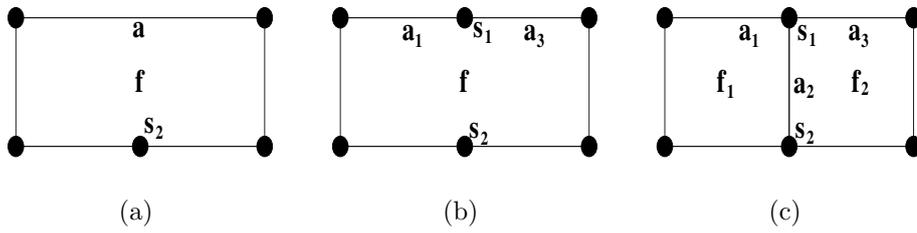


Figure 3.22: Exemples d'insertions en $2D$. (a) Une subdivision $2D$ initiale. (b) Le résultat après l'insertion du sommet s_1 sur l'arête a : l'arête a a été découpée en deux arêtes a_1 et a_3 . (c) Le résultat après l'insertion de l'arête a_2 entre les sommets s_1 et s_2 . La face f a été découpée en deux faces f_1 et f_2 .

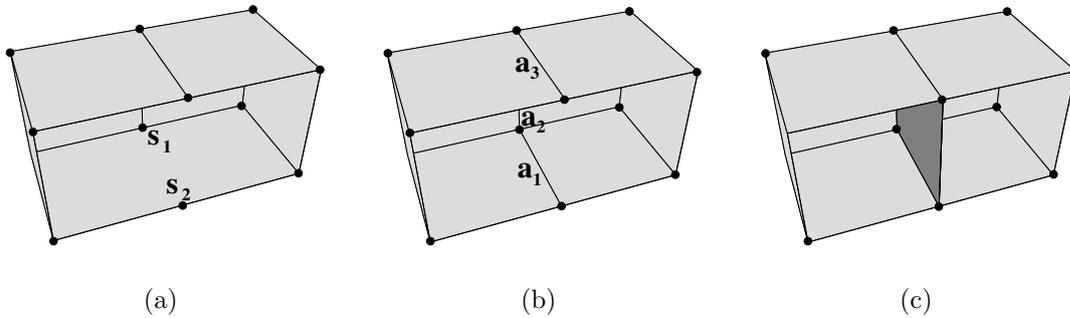


Figure 3.23: Exemples d'insertions en $3D$. (a) Une subdivision $3D$ initiale. (b) Le résultat après l'insertion d'une arête entre les sommets s_1 et s_2 . (c) Le résultat après l'insertion d'une face le long des arêtes a_1 , a_2 et a_3 .

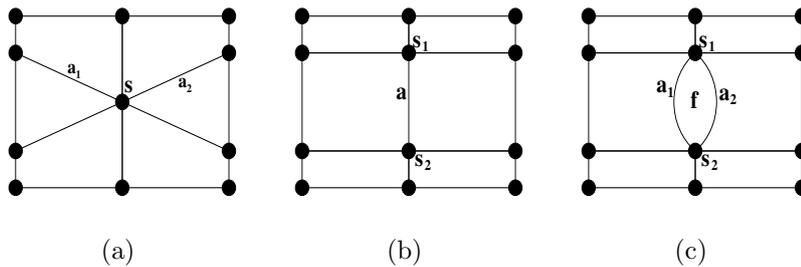


Figure 3.24: Exemples d'éclatements en $2D$. (a) Une subdivision $2D$ initiale. (b) Le résultat après l'éclatement du sommet s le long des arêtes a_1 et a_2 . (c) Le résultat après l'éclatement de l'arête a .

3.3.3.1 Insertion

L'insertion d'une i -cellule c dans une n - G -carte G consiste (inversement à l'opération de suppression) à ajouter cette cellule à G puis à scinder une $(i + 1)$ -cellule (là où c doit être

3.3. LES OPÉRATIONS DE BASE

insérée). Il est à noter que l'opération d'insertion peut être définie pour les cellules dont les dimensions vont de 0 à $n - 1$. La différence principale avec la suppression se situe au niveau des paramètres d'entrée. Dans ce cas, une involution γ explicitant comment relier les brins de G avec ceux de c s'avère nécessaire. En effet, il existe plusieurs façons de relier les brins de c avec ceux de la $(i + 1)$ -cellule sujette à la scission.

Définition 3.6 (*i*-insertion)

Soient $G = (B, \alpha_0, \alpha_1, \dots, \alpha_n)$ une n -G-carte, $i \in \{0, \dots, n - 1\}$, C une i -cellule à insérer, BV et BV' deux sous-ensembles de brins où $BV \subseteq B$ et $BV' \subseteq C$, et γ une involution définie sur $BV \cup BV'$ tel que $b \in BV \Leftrightarrow b\gamma \in BV'$. C peut être insérée si :

1. $\forall b \in BV', b$ est i -libre ($b\alpha_i = b$);
2. $\forall b \in C, b\alpha_{i+1}\alpha_{i+2} = b\alpha_{i+2}\alpha_{i+1}$ ¹⁰;
3. $\forall b \in BV \cup BV', \forall j$ $0 \leq j \leq n$ tel que $|i - j| \geq 2$ et $b\alpha_j \in BV \cup BV'$ alors $b\alpha_j\gamma = b\gamma\alpha_j$;
4. $\forall b \in BV, b\alpha_i = b\gamma\alpha_{i+1}(\alpha_i\alpha_{i+1})^k\gamma$ (avec k le plus petit entier tel que $b\gamma\alpha_{i+1}(\alpha_i\alpha_{i+1})^k \in BV'$).

La n -G-carte obtenue en insérant C dans G est $G' = (B', \alpha'_0, \dots, \alpha'_n)$, définie par :

- $B' = B \cup C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \forall b \in B' : b\alpha'_j = b\alpha_j$;
- $\forall b \in B' - (E \cup F) : b\alpha'_i = b\alpha_i$;
- $\forall b \in BV \cup BV' : b\alpha'_i = b\gamma$.

Dans cette définition, seule est redéfinie l'involution α_i pour les brins de $BV \cup BV'$. En effet, l'insertion entraîne la couture des brins de BV avec ceux de BV' par α_i (qui est donnée en paramètre par γ). Toutes les autres involutions restent inchangées. Par ailleurs, nous démontrons la validité de cette opération (c'est-à-dire la validité de la G-carte résultante), en annexe, à la fin de ce mémoire.

Les préconditions de l'opération d'insertion (au nombre de 4) permettent de garantir un certain nombre de choses. En l'occurrence :

1. l'ensemble des brins supprimés ne constitue qu'une seule cellule de dimension i . En effet, s'il existe un brin $b \in C$ lié à un autre brin $b' \neq b$ (b n'est pas i -libre) par une involution α_i , alors b et b' n'appartiennent pas à la même i -cellule. Cela est en contradiction avec la définition 3.6;
2. C doit être de degré local supérieur 2. L'objectif est de garantir que le résultat de l'insertion est une quasi-variété mais aussi que l'insertion est l'inverse de la suppression. Dans le cas contraire (c'est-à-dire, le degré local supérieur de C est différent de 2), on ne peut pas envisager la suppression de C après l'avoir insérée puisque son degré local supérieur ne satisfait pas à la précondition de l'opération de suppression (voir la définition 3.3);
3. la troisième condition contraint l'opération d'insertion à définir une G-carte valide (la n -G-carte G' doit respecter la condition de la quasi-variété);
4. le dernier point garantit que les opérations d'insertion et de suppression (telles que présentées dans les définitions 3.6 et 3.3) sont inverses l'une de l'autre. Cette précondition n'est rien

10. Cette condition ne concerne pas les cellules de dimension $n - 1$. En effet, il est toujours possible d'insérer une $(n - 1)$ -cellule.

d'autre que l'expression des chemins supprimés lors de l'application de l'opération de suppression. Il suffit juste de considérer les brins de $BV \cup BV'$ comme ceux survivants à la suppression de C et l'involution γ comme l'involution α_i reliant les brins survivant aux brins de la cellule à supprimer.

Pour illustrer la pertinence des préconditions requises et les résultats de cette opération dans différentes configurations, passons en revue les cas de la 0-insertion (cas 1 à 3) et de la 1-insertion (cas 4 et 5) dans les espaces $1D$ et $2D$. Ainsi,

1. La figure 3.25(b) montre un exemple d'insertion de sommet vérifiant les quatre préconditions de l'opération. En revanche, la figure 3.25(c) illustre un exemple où l'insertion d'un autre sommet pose problème à cause du non respect de la quatrième précondition de la définition. Pour ce dernier cas, $3\alpha_0 = 4$ est différent de $3\gamma\alpha_1\gamma = 3$. L'insertion d'un tel sommet (constitué d'un seul brin 7) donne un résultat qui n'est plus une G-carte car l'involution α_0 du brin 3 est modifiée mais pas celle du brin 4.

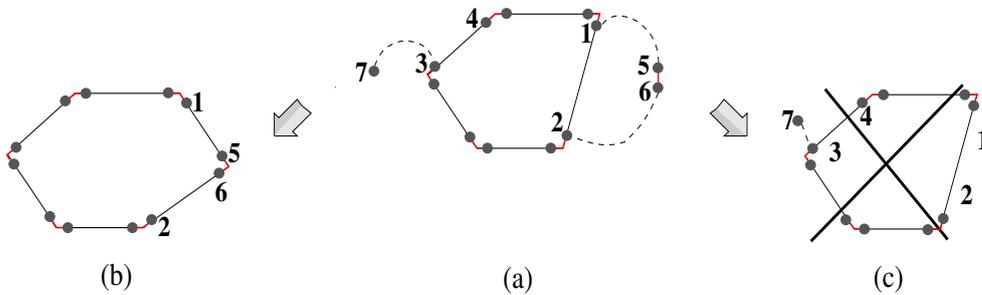


Figure 3.25: 0-insertion en $1D$. (a) 1-G-carte initiale G qui correspond à l'orbite $\langle \alpha_0, \alpha_1 \rangle (1)$, une 0-cellule $C_1 = \langle \alpha_1 \rangle (7) = \{7\}$, $BV_1 = \{3\}$, $BV'_1 = \{7\}$, l'involution γ est représentée par les traits pointillés : γ relie respectivement les brins 1, 2 et 3 avec les brins 5, 6 et 7. Une seconde 0-cellule $C_2 = \langle \alpha_1 \rangle (5) = \{5, 6\}$, $BV_2 = \{1, 2\}$, $BV'_2 = \{5, 6\}$. (b) Résultat invalide après insertion de C_1 dans G (α_0 n'est plus une involution car $3\alpha_0 = 7$ et $4\alpha_0 = 3$). Ce cas est interdit par les préconditions de l'opération d'insertion. (c) Résultat valide après insertion de C_2 dans G (la précondition $b\alpha_0 = b\gamma\alpha_1\gamma$ est satisfaite pour tout les brins de E_2).

2. L'exemple de la figure 3.26 illustre un autre cas où la quatrième condition de l'opération d'insertion n'est pas vérifiée. En effet, $1\alpha_0 = 1$ est différent de $1\gamma\alpha_1\gamma = 2$. Dans ce cas, l'insertion de la cellule donne une G-carte valide (car représentant une quasi-variété cellulaire), mais n'est pas l'inverse de l'opération de suppression.
3. De même, la figure 3.27 illustre deux exemples de 0-insertion dans une 2-G-carte. Bien que satisfaisant les préconditions de l'opération d'insertion, ce cas de figure montre la nécessité de l'involution γ , car il y a plusieurs manières d'insérer la même cellule dans la même carte à travers les mêmes brins (voir les figures 3.27(b) et 3.27(c)).
4. Un premier exemple de 1-insertion dans une 2-G-carte est présenté dans la figure 3.28. Les quatre préconditions de l'opération sont vérifiées et le résultat est une 2-G-carte valide.
5. La figure 3.29 illustre un second exemple de 1-insertion dans une 2-G-carte mais cette fois pour un cas particulier : l'insertion d'une boucle. La seule différence avec l'exemple précédent est qu'ici, l'exposant $k = 1$ alors qu'avant k était égal à 0. En effet, pour redéfinir

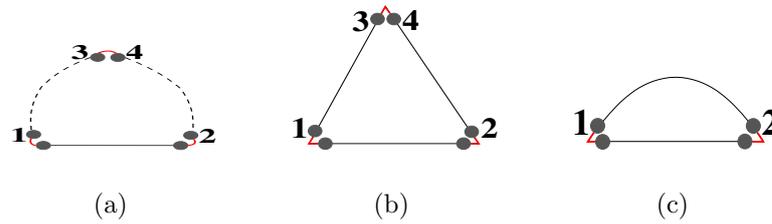


Figure 3.26: Exemple où la quatrième précondition de la 0-insertion n'est pas satisfaite. (a) Une 1-G-carte G et un sommet à insérer $C = \{3, 4\}$. L'involution γ est représentée par les traits pointillés. $BV = \{1, 2\}$ et $BV' = \{3, 4\}$ sont deux sous-ensembles de brins 0-libres. (b) La 1-G-carte G' obtenue après insertion de C dans G . (c) La 1-G-carte obtenue après suppression de C dans G' : les brins 1 et 2 sont maintenant 0-cousus alors qu'ils étaient 0-libre avant l'insertion : la suppression n'est pas l'inverse de l'insertion.

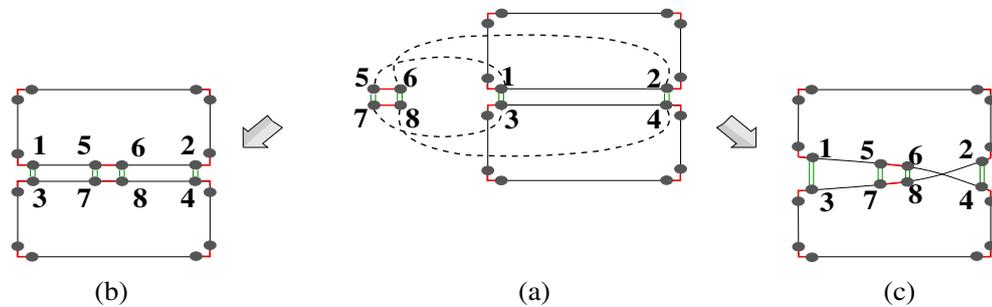


Figure 3.27: 0-insertion en 2D. (a) 0-cellule $C = \langle \alpha_1, \alpha_2 \rangle (5) = \{5, 6, 7, 8\}$, $BV = \{1, 2, 3, 4\}$, $BV' = C$. La 2-G-carte initiale G égale à l'orbite $\langle \alpha_0, \alpha_1, \alpha_2 \rangle (1)$. L'involution γ_1 est représentée par les traits pointillés (respectivement entre les brins 1, 2, 3, 4 et les brins 5, 6, 7, 8). L'involution γ_2 (non représentée sur la figure) relie respectivement les brins 1, 2, 3, 4 avec les brins 5, 8, 7, 6. (b) La 2-G-carte obtenue après insertion de C dans G en utilisant l'involution γ_1 . (c) La 2-G-carte obtenue après insertion de C dans G en utilisant l'involution γ_2 .

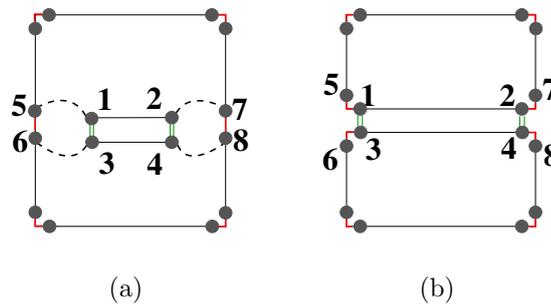


Figure 3.28: 1-insertion en 2D dans le cas général. (a) Les brins de l'arête à insérer sont numérotés de 1 à 4. L'involution γ est représentée par les traits pointillés. (b) Le résultat de l'insertion.

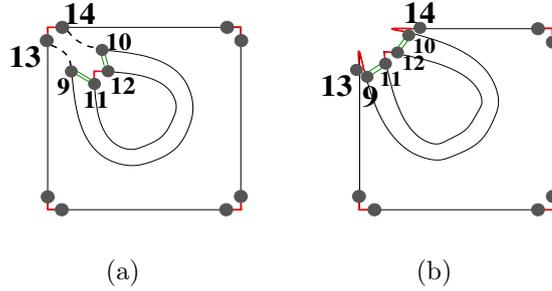


Figure 3.29: 1-insertion en 2D dans un cas particulier : l'insertion d'une boucle. (a) Les brins de l'arête à insérer sont numérotés de 9 à 12. L'involution γ est représentée par les traits pointillés. La troisième précondition de l'opération est vérifiée car $13\alpha_1 = 13\gamma\alpha_2(\alpha_1\alpha_2)\gamma = 14$. (b) Le résultat de l'insertion.

l'involution α_1 reliant les brins 13 et 14, il faut traverser l'arête avant de retomber sur un brin de BV' . Comme les quatre préconditions de l'opération sont vérifiées, le résultat est une 2-G-carte valide.

3.3.3.2 Éclatement

De manière analogue à la contraction pour la suppression, l'opération d'éclatement est définie de manière similaire à l'opération d'insertion. Il suffit donc de remplacer, dans la définition de l'opération d'insertion, les indices $i + 1$ par $(i - 1)$ et $i + 2$ par $(i - 2)$.

Définition 3.7 (*i*-éclatement)

Soit $G = (B, \alpha_0, \alpha_1, \dots, \alpha_n)$ une n -G-carte, $i \in \{1, \dots, n\}$, $C = \langle \rangle_{\{1, 2, \dots, n\} - \{i\}}$ (b) une i -cellule à éclater, BV et BV' deux sous-ensembles de brins avec $BV \subseteq B$ et $BV' \subseteq C$, et γ une involution sur $BV \cup BV'$ avec $b \in BV \Leftrightarrow b\gamma \in BV'$. C peut être éclatée si :

- $\forall b \in BV', b$ est i -libre ;
- $\forall b \in C, b\alpha_{i-1}\alpha_{i-2} = b\alpha_{i-2}\alpha_{i-1}$;¹¹ ;
- $\forall b \in BV \cup BV', \forall j$ $0 \leq j \leq n$ tel que $|i - j| \geq 2$ et $b\alpha_j \in BV \cup BV'$ alors $b\alpha_j\gamma = b\gamma\alpha_j$;
- $\forall b \in BV, b\alpha_i = b\gamma\alpha_{i-1}(\alpha_i\alpha_{i-1})^k\gamma$ (avec k le plus petit entier tel que $b\gamma\alpha_{i-1}(\alpha_i\alpha_{i-1})^k \in BV'$).

La n G-carte obtenue en éclatant C dans G est $G' = (B', \alpha'_0, \dots, \alpha'_n)$ définie par :

- $B' = B \cup C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \forall b \in B' : b\alpha'_j = b\alpha_j$;
- $\forall b \in B' - (BV \cup BV') : b\alpha'_i = b\alpha_i$;
- $\forall b \in BV \cup BV' : b\alpha'_i = b\gamma$.

Les figures 3.30 et 3.31 montrent deux exemples d'éclatement, le premier en 1D et le second en 2D. Le principe est le même que pour l'insertion : la différence réside uniquement au niveau des préconditions de l'opération qui garantissent que l'éclatement est l'opération inverse de la contraction.

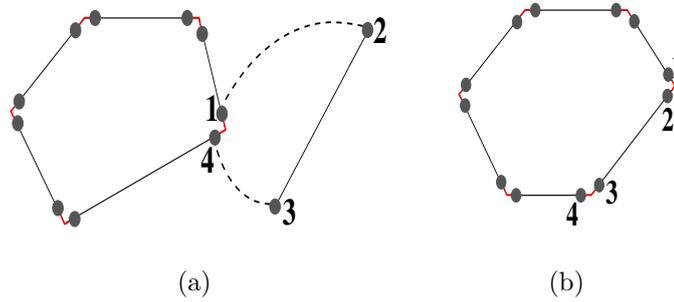


Figure 3.30: 1-éclatement en 2D. (a) Les brins de l'arête à éclater sont numérotés 2 et 3. L'involution γ est représentée par les traits pointillés. (b) Le résultat de l'éclatement.

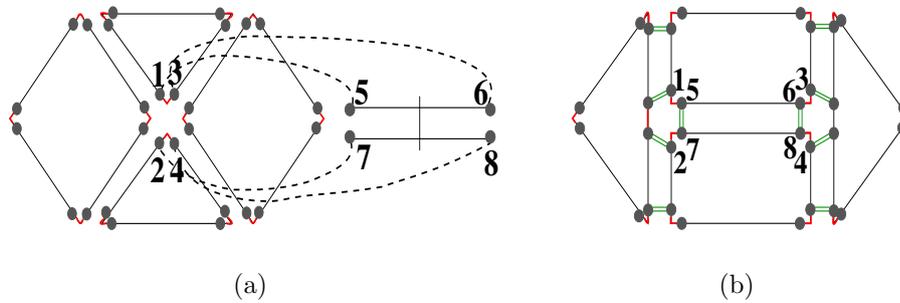


Figure 3.31: 1-éclatement en 2D. (a) La 2-G-carte initiale et l'arête à éclater (brins numérotés de 5 à 8). L'involution γ est représentée par les traits pointillés. (b) Le résultat de l'éclatement.

3.3.3.3 Généralisation

Les deux définitions précédentes permettent d'effectuer l'insertion ou l'éclatement d'une seule cellule. Comme pour les opérations de suppression et de contraction, il est intéressant de pouvoir effectuer plusieurs fois ces opérations de façon simultanée. Les paramètres d'entrée de l'opération généralisée comprennent une n -G-carte initiale G , une seconde n -G-carte G' contenant les cellules à insérer et/ou à éclater et une involution γ reliant les brins de la première à la seconde. Chaque brin de G' est marqué avec la dimension et le type de l'opération (insertion ou éclatement). Les opérations peuvent être appliquées simultanément si les cellules sont disjointes et si chaque cellule vérifie la précondition éventuelle des opérations unitaires. La G-carte résultante peut alors être calculée directement. Les nouvelles involutions sont alors les involutions de G et G'

mais aussi les involutions γ (voir la définition 3.8).

Définition 3.8 (Insertion et éclatement simultanées de cellules)

Soient $G = (B, \alpha_0, \dots, \alpha_n)$ une n - G -carte, $G' = (B', \alpha_0, \dots, \alpha_n)$ une seconde n - G -carte¹², $\{I_i\}_{0 \leq i \leq n-1}$ les ensembles de i -cellules à insérer et $\{E_i\}_{1 \leq i \leq n}$ les ensembles de i -cellules à éclater et $I_n = E_0 = \emptyset$. Posons $I = \bigcup_{i=0}^n I_i$ et $E = \bigcup_{i=0}^n E_i$. Les cellules de $I \cup E$ satisfont les préconditions suivantes :

- elles sont deux à deux disjointes (i. e. $\forall c, c' \in I \cup E, c \cap c' = \emptyset$);
- les cellules de I sont de degré local supérieur égal à 2;
- les cellules de E sont de degré local inférieur égal à 2;
- toutes les cellules de G' sont à insérer ou à éclater : $I \cup E = B'$.

Pour tout i , $0 \leq i \leq n$, $BV_i \subseteq B$, $BV'_i \subseteq B'$ et γ_i une involution définie sur $BV_i \cup BV'_i$ avec $b \in BV_i \Leftrightarrow b\gamma_i \in BV'_i$. L'opération peut être appliquée si $\forall i$, $0 \leq i \leq n$:

- $\forall b \in BV'_i$, b est i -libre;
- $\forall b \in BV_i \cup BV'_i$, $\forall j$, $0 \leq j \leq n$ tel que $|i - j| \geq 2$ et $b\alpha_j \in BV_i \cup BV'_i$ alors $b\alpha_j\gamma_i = b\gamma_i\alpha_j$;
- $\forall b \in BV_i$, $b\alpha_i = b' = b\gamma_i\alpha_{l_1}(\alpha_i\alpha_{l_2}) \dots (\alpha_i\alpha_{l_k})\gamma_i$ avec :
 - k le plus petit entier tel que $b' \in BV'_i$;
 - $l_1 = i + 1$ si $b\gamma_i \in I$; $l_1 = i - 1$ sinon (i.e. $b\gamma_i \in E$);
 - $\forall j$, $2 \leq j \leq k$, $l_j = \begin{cases} i + 1 & \text{si } b_j = b\gamma_i\alpha_{l_1}(\alpha_i\alpha_{l_2}) \dots (\alpha_i\alpha_{l_{j-1}}) \in I, \\ i - 1 & \text{sinon (i.e. } b_j \in E). \end{cases}$

La n - G -carte résultant des insertions et des éclatements simultanés des cellules appartenant aux ensembles I et E est $G'' = (B'', \alpha''_0, \dots, \alpha''_n)$ définie par :

- $B'' = B \cup B'$;
- $\forall i$, $0 \leq i \leq n$, $\forall b \in B'' - BV_i \cup BV'_i$, $b\alpha''_i = b\alpha_i$;
- $\forall i$, $0 \leq i \leq n$, $\forall b \in BV_i \cup BV'_i$, $b\alpha''_i = b\gamma_i$.

Comme pour l'opération de suppression et/ou de contraction simultanées, les cellules doivent être disjointes, et de degré local (inférieur pour les cellules à insérer et supérieur pour les cellules à éclater) égal à 2. Pour cette opération généralisée, il y a, au plus, $n + 1$ involutions γ_i (une par dimension de la subdivision). Chaque involution doit vérifier les préconditions de l'opération de base :

- les brins de BV'_i doivent être i -libres pour garantir que les α''_i sont des involutions;
- $b\alpha_j\gamma_i = b\gamma_i\alpha_j$ pour garantir que la composition des $\alpha''_i \circ \alpha''_j$ est une involution (garantir la quasi-variété de la n - G -carte G'');
- $b\alpha_i = b\gamma_i\alpha_{l_1}(\alpha_i\alpha_{l_2}) \dots (\alpha_i\alpha_{l_k})$ pour garantir qu'il existe un chemin traversant les cellules insérées ou éclatées entre b et $b\alpha_i$.

La figure 3.32 montre un exemple de l'opération d'insertion et/ou d'éclatement simultanée. Sur cet exemple, les quatre opérations existant en dimension 2 sont simultanément utilisées : le 1- et 2-éclatement, et la 0- et 1-insertion.

12. Par abus de notation, nous utilisons également α_i pour les involutions de G' afin de simplifier les écritures. Il est possible d'utiliser α'_i mais il faut alors différencier les formules pour les brins de B et ceux de B' alors que ces formules peuvent être factorisées en utilisant une même notation.

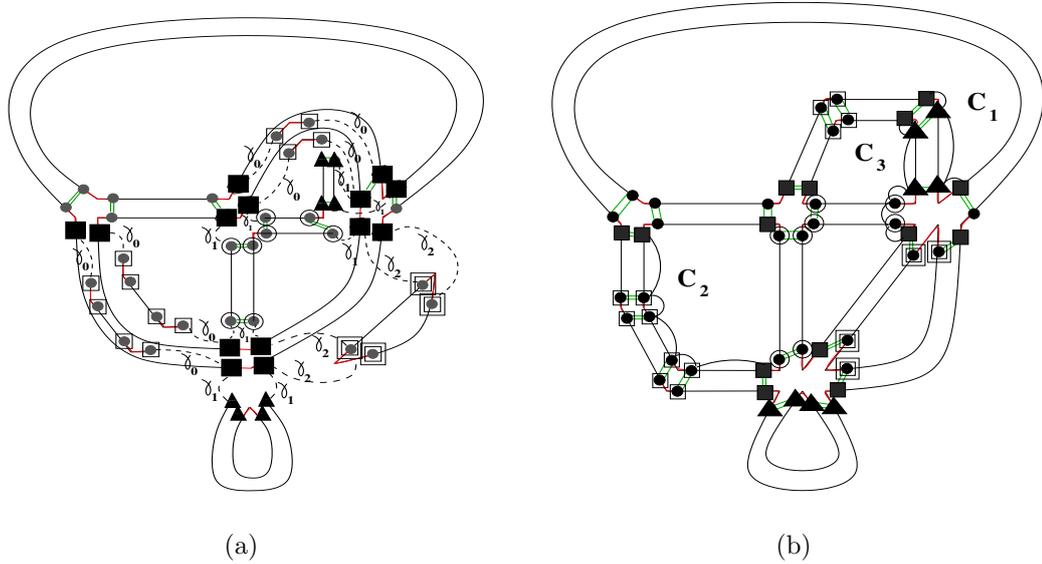


Figure 3.32: Un exemple $2D$ d'insertions et d'éclatements simultanés de cellules de différentes dimensions. (a) La 2-G-carte initiale et les cellules à insérer et éclater. Les involutions γ_i sont représentées par les traits pointillés, avec un numéro indiquant la dimension de l'opération. (b) La 2-G-carte obtenue après application de l'opération. Les brins appartenant à une 1-cellule insérée (resp. 0-cellule insérée, 1-cellule éclatée, 2-cellule éclatée) sont marqués avec un cercle (resp. un carré vide, un triangle plein, un double carré). Les brins marqués avec des carrés pleins appartiennent à $\cup BV_i$.

3.4 Implantation des mécanismes de suivi d'événements sur le modèle des cartes généralisées

Suivre l'évolution des cellules, dans le cas d'un noyau géométrique réel, revient à implanter les mécanismes de suivi d'événements précédemment décrits. Comme nous l'avons expliqué, ces mécanismes se déploient en associant des ancêtres aux cellules constituant l'instance courante de la géométrie. Or, dans une G-carte (voir la section 2.1.2.2), les cellules correspondent à une collection de brins. Donc, tracer l'évolution des cellules revient à associer des ancêtres aux brins constituant ces cellules. Pour cela, une structure de données nommée *tag* a été définie.

3.4.1 Solution apportée par la structure de *tag*

La structure de *tag* correspond à un couple, constitué de quadruplets d'entiers (un entier par dimension de cellules lorsqu'on travaille dans un espace $3D$) que nous associons à chaque brin d'une carte généralisée. Les deux composantes du couple, nommées respectivement *tag_actuel* et *tag_ancêtre*, sont définies par :

- *tag_actuel* : correspond au quadruplet d'identifiants $(id_{act_0}, id_{act_1}, id_{act_2}, id_{act_3})$ et caractérise respectivement la 0, 1, 2 et 3-cellule contenant le brin b ;
- *tag_ancêtre* : correspond au quadruplet d'identifiants $(id_{anc_0}, id_{anc_1}, id_{anc_2}, id_{anc_3})$ et référence respectivement les 0, 1, 2 et 3-cellules, ancêtres des 0, 1, 2 et 3-cellules contenant

b.

La figure 3.33 montre les tags associés aux brins d'une arête avant et après l'opération élémentaire de scission d'arête. En particulier, le quadruplet $(4, 1, 2, 3)$ désignant le *tag_actuel* associé au brin b_1 dans la figure 3.33(a) signifie que b_1 appartient à la 0-cellule numéro 4, à la 1-cellule numéro 1, à la 2-cellule numéro 2 et à la 3-cellule numéro 3. Le principe est le même pour tous les autres brins.

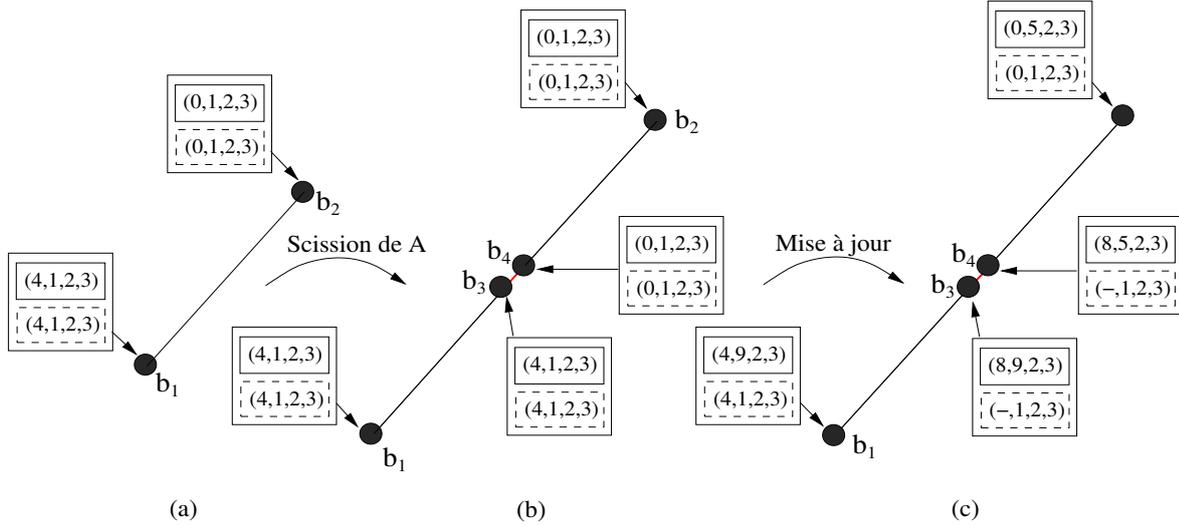


Figure 3.33: Scission d'une arête. Pour chaque brin, les tags actuels sont entourés d'un cadre plein, tandis que les tags ancêtres sont entourés par des pointillés. (a) Les brins b_1 et b_2 composent l'arête A avant la scission. (b) A est scindée en $A_1 = (b_1, b_3 = \alpha_0(b_1))$ et $A_2 = (b_2, b_4 = \alpha_0(b_2))$. Le sommet S joignant A_1 et A_2 est composé de b_3 et $b_4 = \alpha_1(b_3)$. b_1 et b_2 gardent leurs tags et les transmettent respectivement aux nouveaux brins de A_1 et A_2 : b_3 et b_4 . (c) Mise à jour du *tag_actuel* de b_1 et b_2 , et mise à jour des deux tags de b_3 et b_4 .

Afin de connaître la dimension de chaque cellule à travers son identifiant, tous les éléments des deux précédents quadruplets sont définis modulo 4. En effet, l'espace de modélisation est un espace $3D$ et le calcul de la dimension s'effectue de la manière suivante : $id_{act_i} \bmod 4$.

Tous les *tag_actuels* associés aux brins de la même i -cellule portent le même identifiant id_{act_i} . Dans l'exemple de la figure 3.33(a) (resp. de la figure 3.33(c)), les brins b_1 et b_2 (resp. les brins b_3 et b_4) de la 1-cellule A (resp. de la 0-cellule S) portent le même id_{act_1} de valeur 1 (resp. le même id_{act_0} de valeur 8). En revanche, une i -cellule est associée à un ensemble d'identifiants. Celui-ci peut être vide ou composé de plusieurs éléments. Chacun d'eux référence un ancêtre différent. Dans l'exemple de la figure 3.33(c), le sommet S est nouvellement créé. De ce fait, S n'a pas d'ancêtre. Donc, les brins b_3 et b_4 de S portent des *tag_ancêtres* dont la première composante, id_{anc_0} , est à chaque fois égal à $NULL$ (symbolisé par « - » dans la figure). De cette façon, Les structures de *tag_actuel* et de *tag_ancêtre* permettent d'associer l'identifiant d'une cellule à son ancêtre.

Pour saisir le fonctionnement des structures de tags dans le cadre d'une application réelle, nous avons choisi d'étudier le cas de l'opération globale de co-raffinement.

Jusqu'à présent, pour réaliser des opérations d'usinage sur des objets géométriques, le recours aux opérations booléennes classiques telles que l'union, la soustraction ou bien l'intersection

3.4. IMPLANTATION DES MÉCANISMES DE SUIVI D'ÉVÉNEMENTS

s'imposait. Le coraffinement s'avère être un dénominateur commun pour le calcul d'intersections et permet d'obtenir, après une phase d'extraction¹³, les mêmes résultats que les trois opérations booléennes réunies.

L'opération de co-raffinement en dimension 3 consiste à générer la subdivision de l'espace résultant de l'interaction de deux subdivisions 3D originales. Les subdivisions en dimension 3 étant formées de volumes, le traitement principal consiste alors à calculer les intersections existant entre les faces bordant ces volumes et à mettre à jour la topologie pour obtenir une subdivision finale correcte. Ces intersections sont alors représentées sous la forme de segments de coupe (figure 3.34).

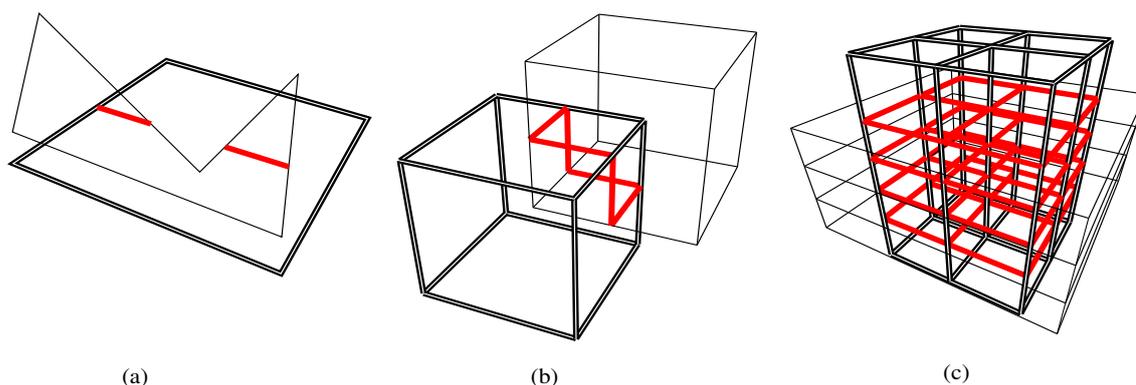


Figure 3.34: Intersection entre différents éléments : les lignes rouges en gras représentent les lignes ou graphes de coupe. (a) Intersection entre deux faces. (b) Intersection entre deux volumes. (c) Intersection entre deux maillages.

Comme dans le cas des opérations booléennes, de nombreux problèmes peuvent être rencontrés dans le calcul des intersections ([MT83], [MT88], [KY92], [GMP03] et [Per05]). En effet, l'algorithme de co-raffinement 3D que nous avons choisi d'utiliser, reprend des approches fréquemment utilisées dans les algorithmes des opérations booléennes d'objets surfaciques. De plus, Guiard [Gui06] l'a étendu à des maillages 3D. Pour ce faire, il utilise un algorithme basé sur l'intersection de couples de faces et il l'implante sur le modèle des G-cartes. Par ailleurs, le nombre important et le caractère non prévisible des événements qui surviennent lors d'une opération de co-raffinement rend difficile, et donc particulièrement intéressante, l'étude de leur suivi.

3.4.2 Opérations élémentaires et mécanismes de suivi d'événements intégrés

Pour suivre l'évolution des cellules résultant de cette opération de modélisation, nous avons répertorié, avec l'aide de Guiard, un ensemble d'opérations élémentaires : insertion d'une arête pendante dans une face, scission d'une arête, scission d'une face et identification de faces. Nous avons souhaité, à travers cette décomposition, mettre en exergue des parties de l'opération ayant une sémantique et un fonctionnement toujours prévisible. À partir de là, les mécanismes de suivi d'événements sont associés et implantés à chaque opération élémentaire.

13. Il s'agit d'une étape visant à ne garder qu'un seul volume de la subdivision 3D résultant de l'opération de co-raffinement. Ce volume correspond au résultat de l'opération booléenne que l'on souhaite appliquer.

Cet ensemble ne représente qu'une partie des opérations élémentaires pouvant être définies dans un noyau géométrique tel que celui des G-cartes. Cependant, l'approche utilisée pour intégrer les mécanismes de suivi d'événements reste identique et peut donc, de façon similaire, être étendue à toute autre opération. Pour décrire les traitements réalisés sur les structures de tags, nous commençons par définir plusieurs fonctions en donnant à chaque fois leur prototype et leur fonctionnement :

- `Dupliquer_Tag`($b : \text{Brin}, O : \text{Orbite}$) : permet de propager le tag (tag_actuel et tag_ancêtre) associé au brin b vers tous les brins de l'orbite O ,
- `Remplacer_ID_Ancêtre`($O : \text{Orbite}, d : [0..\text{Dim_max}], id : \text{ID_Cellule}$) : cette fonction permet de mettre à jour la valeur de l'identifiant id_anc_d ¹⁴ de chaque tag_ancêtre associé aux brins de l'orbite O avec la valeur id (id doit vérifier la condition suivante : $\text{id} \bmod 4 = d$),
- `Remplacer_ID_Actuel`($O : \text{Orbite}, d : [0..\text{Dim_max}], id : \text{ID_Cellule}$) : joue le même rôle que la fonction `Remplacer_ID_Ancêtre`. Cependant, comme son nom l'indique, elle manipule la structure de tag_actuel ,
- `Récupérer_Tag_Ancêtre`($O : \text{Brin}, d : [0..\text{Dim_max}]$) $\rightarrow \text{ID_Cellule}$: permet de récupérer et de retourner la valeur de l'identifiant id_anc_d constituant le tag_ancêtre associé au brin b ,
- `Nouveau_ID`($d : [0..\text{Dim_max}]$) $\rightarrow \text{ID_Cellule}$: permet de générer et de retourner un nouvel identifiant id_act_d ¹⁵ du tag_actuel non encore attribué,
- `Remplacer_ID_Ancêtre`($b_1 : \text{Brin}, b_2 : \text{Brin}$) : permet de remplacer le tag_ancêtre associé au brin b_1 par celui associé au brin b_2 .

3.4.2.1 Insertion d'une arête pendante dans une face

Lorsqu'une intersection est effectuée, la géométrie des objets entraîne souvent l'insertion de segments à l'intérieur d'une face sans aucune liaison avec son bord (figure 3.35(a)). Cependant, nous devons trouver un moyen de conserver les liens entre chaque face et les segments de droite qu'elle contient (s'ils existent). Il existe deux manières de stocker cette information, avec chacune ses avantages et ses inconvénients.

La première approche, que l'on ne détaillera pas ici, consiste à associer à chaque face une liste contenant les segments ou les faces qu'elle contient. Cette approche a pour inconvénient majeur la lourdeur dans la gestion des listes utilisées (perte de temps sur les tests d'appartenance de segments lors d'opérations entraînant la modification de la topologie).

La seconde consiste à représenter explicitement la relation d'appartenance d'un segment à une face en le reliant au bord de cette dernière à l'aide d'une « arête fictive ». Cependant, bien que la topologie soit simple à mettre à jour lors d'un calcul d'intersection, l'ajout d'une arête fictive dans une face est plus compliqué.

Lorsqu'une arête est insérée, on doit éviter toute auto-intersection avec le bord de la face afin de garder une topologie correcte (figure 3.35(b)). Cette arête est considérée comme n'importe quelle autre arête et peut également être intersectée. Ainsi, lorsqu'une face est découpée en plusieurs morceaux, les arêtes fictives peuvent également être découpées et la nouvelle relation d'appartenance découle directement de la nouvelle topologie calculée (figure 3.36).

Afin de gérer le problème d'inclusion d'éléments dans une face, Guiard décide d'utiliser la seconde méthode. Outre cette configuration (insertion de segment dans une face), l'ajout d'un

14. id_anc_d est le $d^{\text{ème}}$ identifiant d'un tag_ancêtre .

15. id_act_d est le $d^{\text{ème}}$ identifiant d'un tag_actuel .

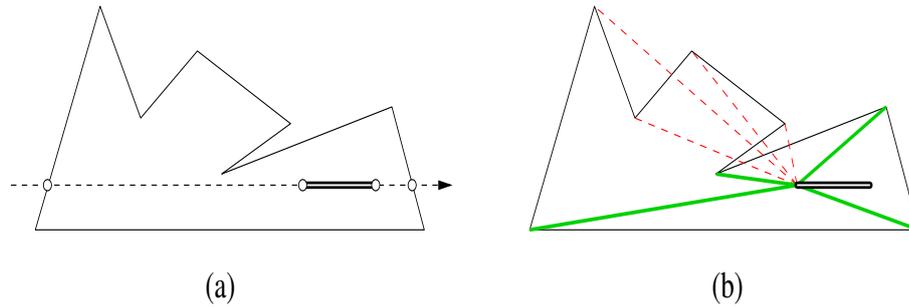


Figure 3.35: Inclusion d'un segment au sein d'une face à l'aide d'une arête fictive. (a) La ligne en pointillés représente la ligne de coupe, les cercles représentent des points d'intersections et le segment double correspond à un segment de coupe devant être ajouté sur la face. (b) Les lignes en pointillés définissent des arêtes fictives géométriquement non valides et les lignes en gras des arêtes fictives valides.

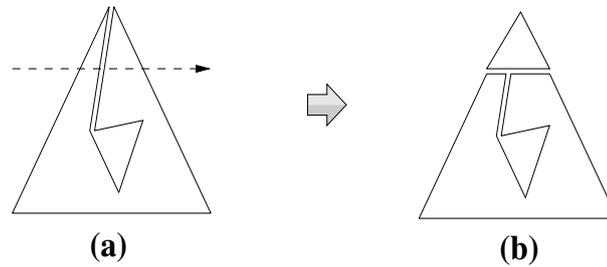


Figure 3.36: Exemple d'arête fictive découpée lors d'un calcul d'intersection. (a) La ligne en pointillés correspond à la ligne de coupe et la ligne double à une arête fictive. (b) Topologie résultant du calcul d'intersection (la partie haute de l'arête fictive a été supprimée).

sommet au milieu d'une face est un autre cas de figure pouvant se présenter. L'utilisation d'une arête fictive¹⁶, dans ce dernier cas, constitue également une solution appréciable au même titre que celle exposée auparavant.

L'insertion d'une arête pendante nommée A dans une face nommée F s'effectue *via* la fonction `Insérer_Arête` dont le prototype et le fonctionnement sont décrits par :

`Insérer_Arête`(b : Brin, p : Point) \rightarrow Brin : permet d'insérer une arête dans la face incidente au brin b . La nouvelle arête est cousue (liée par l'involution α_1) à b d'un côté et plongée par p de l'autre. La fonction retourne le brin de la nouvelle arête¹⁷ correspondant à $\alpha_1(b)$.

Comme nous le voyons dans la figure 3.37, cette fonction entraîne la création d'un sommet S_1 (composé des brins 5 et 6), plongée en p , la création d'une arête A (composée des brins 3, 4, 5 et 6) et la modification du sommet S (composée des brins 1 et 2) engendrant, de fait, les mécanismes *créa_élém* et *modif_élém*.

La fonction `Nommer_arête_pendante` décrite dans l'algorithme 6 permet d'implanter ces mécanismes. En effet, elle associe les tags adéquats aux brins de l'arête pendante A insérée dans F . `Nommer_arête_pendante` intervient après l'insertion de A et prend en paramètre le brin restitué par la fonction `Insérer_Arête`. Dans l'exemple de la figure 3.37, l'arête pendante est

16. Le sommet que l'on souhaite insérer constitue une extrémité de l'arête fictive alors que l'autre extrémité correspond à un sommet du bord de la face considérée.

17. Ce brin est incident au sommet de l'arête pendante qui se trouve relié au bord de la face par α_1 .

insérée dans la face incidente au brin 1 et le brin restitué par la fonction `Insérer_Arête` est celui qui porte le numéro 3.

Algorithme. 6 : Nommer_arête_pendante

Données : un brin b de l'arête pendante A à insérer.

Résultat : attribution des tags adéquats aux brins de l'arête pendante A .

début

```

1  Dupliquer_Tag( $\alpha_1(b)$ ,  $\langle \alpha_0, \alpha_3 \rangle (b)$ ) ;
2  Dupliquer_Tag( $\alpha_1(\alpha_2(b))$ ,  $\langle \alpha_0, \alpha_3 \rangle (\alpha_2(b))$ ) ;
3  Remplacer_ID_Ancêtre( $\langle \rangle_0 (\alpha_0(b))$ , 0, ID_NULL) ;
4  Remplacer_ID_Ancêtre( $\langle \rangle_1 (b)$ , 1, ID_NULL) ;
5  Remplacer_ID_Actuel( $\langle \rangle_0 (\alpha_0(b))$ , 0, Nouveau_ID( $\emptyset$ )) ;
6  Remplacer_ID_Actuel( $\langle \rangle_1 (b)$ , 1, Nouveau_ID(1)) ;
7  Remplacer_ID_Actuel( $\langle \rangle_2 (b)$ , 1, Nouveau_ID(2)) ;

```

fin

Afin de comprendre le fonctionnement de la fonction `Nommer_arête_pendante`, nous décrivons le déroulement de l'algorithme 6 sur l'exemple de la figure 3.37. On suppose que le brin b porte le numéro 3 sur la figure et que les brins 1 et 2 portent respectivement les tags $((4, 5, 2, 3), (4, 5, 2, 3))$ et $((4, 9, 2, 3), (4, 9, 2, 3))$ (voir la table 3.4).

1. Le brin $\alpha_1(b)$ (le brin numéro 1 dans la figure 3.37) transmet une copie des tags (tag_actuel , $tag_ancêtre$) qui lui sont associés aux brins de l'orbite $\langle \alpha_0, \alpha_3 \rangle (b)$ ¹⁸. Dans la figure 3.37, le brin numéro 1 transmet le couple de quadruplets $((4, 5, 2, 3), (4, 5, 2, 3))$ aux brins numérotés 3 et 5.
2. Les valeurs du tag_actuel et du $tag_ancêtre$ portés par le brin $\alpha_1(\alpha_2(b))$ sont propagées aux brins constituant l'orbite $\langle \alpha_0, \alpha_3 \rangle (\alpha_2(b))$. Dans la figure 3.37, il s'agit de propager le couple de quadruplets $((4, 9, 2, 3), (4, 9, 2, 3))$ associé au brin 2 aux brins numérotés 4 et 6.
3. La valeur de l'identifiant id_anc_0 est mise à jour pour chaque $tag_ancêtre$ associé aux brins de l'orbite $S_1 = \langle \rangle_0 (\alpha_0(b))$ avec la valeur ID_NULL . Dans la figure 3.37, il s'agit de mettre à jour l'identifiant id_anc_0 de chaque $tag_ancêtre$ associé aux brins numérotés 5 et 6 avec la valeur ID_NULL . En effet, l'orbite S_1 est un sommet nouvellement créé.
4. Nous réitérons le traitement effectué à l'étape précédente en mettant à jour l'identifiant id_anc_1 de chaque $tag_ancêtre$ associé aux brins de l'arête $\langle \rangle_1 (b)$ avec la valeur ID_NULL . Ce traitement se traduit dans la figure 3.37 par la mise à jour des identifiants id_anc_1 de chaque $tag_ancêtre$ associé aux brins numérotés 3, 4, 5 et 6. En effet, l'arête A est nouvellement créée.
5. Nous mettons à jour la valeur de l'identifiant id_act_0 de chaque tag_actuel associé aux brins du sommet $\langle \rangle_0 (\alpha_0(b))$ avec un nouvel identifiant généré par la fonction `Nouveau_ID(\emptyset)`. Dans la figure 3.37, l'identifiant id_act_0 de chaque tag_actuel associé aux brins 5 et 6 prend la valeur 8.

¹⁸. L'involution α_3 n'est pas représentée dans la figure 3.37 pour simplifier la visualisation de l'orbite O qu'elle participe à définir. En effet, cette involution remplit son rôle lorsqu'il s'agit d'insérer une arête dans une face adjacente à deux volumes adjacents.

3.4. IMPLANTATION DES MÉCANISMES DE SUIVI D'ÉVÉNEMENTS

6. Nous mettons à jour la valeur de l'identifiant id_{act_1} de chaque tag_{actuel} associé aux brins de l'arête A avec une nouvelle valeur. Dans la figure 3.37, cela concerne l'identifiant id_{act_1} de chaque tag_{actuel} associé aux brins numérotés 3, 4, 5 et 6 qui reçoit un identifiant non encore attribué. Dans ce cas, il s'agit de la valeur 13.
7. De même que pour les étape 5 et 6, nous mettons à jour la valeur de l'identifiant id_{act_2} de chaque tag_{actuel} associé aux brins de la face dans laquelle est insérée l'arête pendante $\langle \rangle_0$ ($\alpha_2(b)$) avec un nouvel identifiant généré par la fonction $Nouveau_ID(2)$. Dans la figure 3.37, l'identifiant id_{act_2} de chaque tag_{actuel} associé aux brins 1, 2, ..., 6, ... prend la valeur 6.

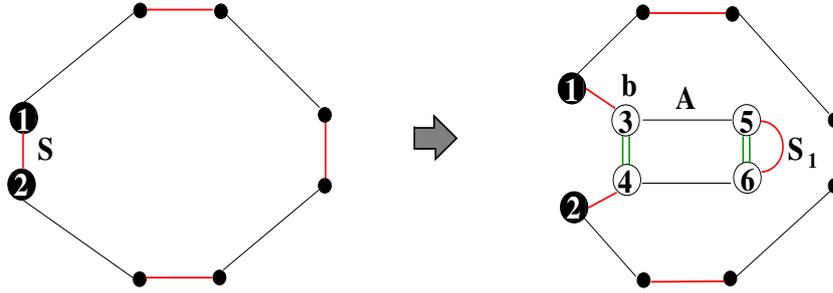


Figure 3.37: Insertion d'une arête pendante dans une face. La nouvelle arête, constituée des brins 3, 4, 5 et 6, est cousue au sommet constitué des brins 1 et 2 par des involutions α_1 . Le contenu de chaque tag associé à chacun de ces brins est décrit dans la table 3.4.

brin	Avant								Après							
	$tag_{ancêtre}$				tag_{actuel}				$tag_{ancêtre}$				tag_{actuel}			
1	4	5	2	3	4	5	2	3	4	5	2	3	4	5	6	3
2	4	9	2	3	4	9	2	3	4	9	2	3	4	9	6	3
3									4	-	2	3	4	13	6	3
4									4	-	2	3	4	13	6	3
5									-	-	2	3	8	13	6	3
6									-	-	2	3	8	13	6	3

Table 3.4: Structures de tags (tag_{actuel} et $tag_{ancêtre}$) associées aux brins de la figure 3.37. Dans cette table, « - » représente la valeur ID_NULL qui peut être associée à un identifiant d'un $tag_{ancêtre}$.

Les étapes 3 et 5 (resp. 4 et 6) correspondent au mécanisme *créa.élé*m. En effet, ces étapes associent au nouvel identifiant 8 (resp. au nouvel identifiant 13) caractérisant S_1 (resp. A) un ensemble vide d'identifiants (symbolisé par la valeur ID_NULL) signifiant que S_1 (resp. A) est dépourvu d'ancêtre. De plus, les étapes 1, 2 et 7 implantent le mécanisme *modif.élé*m appliqué sur le cas de la face modifiée dans laquelle a été insérée l'arête pendante A .

3.4.2.2 Scission d'une arête

L'opération élémentaire de scission d'une arête notée A , se traduit par l'insertion d'un sommet noté S , dans A . De ce fait, elle engendre deux nouvelles arêtes, notées respectivement A_1

et A_2 incidentes à S . Il est alors clair que cette opération invoque les mécanismes *sci_élém* et *créa_élém* car elle met en scène deux événements que sont la scission de A et la création de S .

Regardons cela de plus près en définissant tout d'abord la fonction qui couvre les traitements topologiques de cette opération, à savoir la fonction **Scinder_Arête** définie comme suit :

Scinder_Arête($b : \text{Brin}, p : \text{Point}$) \rightarrow **Brin** : permet d'éclater l'arête incidente à b en y ajoutant un sommet topologique plongé en p . Elle retourne le brin du nouveau sommet correspondant à $\alpha_1(\alpha_0(b))$.

Vient ensuite, la gestion des structures de tags associées aux brins des arêtes résultant de la scission. Elle se traduit par la fonction **Nommer_arête_scindée** décrite dans l'algorithme 7. **Nommer_arête_scindée** intervient après **Scinder_Arête** et prend en paramètre un brin b incident à l'arête A .

Algorithme. 7 : **Nommer_arêtes_scindées**

Données : un brin b incident à l'arête A .

Résultat : attribution des tags adéquats aux brins constituant les arêtes A_1 et A_2 résultant de la scission de A .

début

```

1  |   pour chaque  $b' \in \langle \alpha_2, \alpha_3 \rangle (b)$  faire
   |   |   Dupliquer_Tag( $b', \langle \alpha_1 \rangle (\alpha_0(b'))$ ) ;
2  |   Remplacer_ID_Ancêtre( $\langle \rangle_0 (\alpha_0(b)), 0, \text{ID\_NULL}$ ) ;
3  |   Remplacer_ID_Actuel( $\langle \rangle_0 (\alpha_0(b)), 0, \text{Nouveau\_ID}(0)$ ) ;
4  |   Remplacer_ID_Actuel( $\langle \rangle_1 (b), 1, \text{Nouveau\_ID}(1)$ ) ;
5  |   Remplacer_ID_Actuel( $\langle \rangle_1 (\alpha_1(\alpha_0(b))), 1, \text{Nouveau\_ID}(1)$ ) ;

```

fin

Pour comprendre le fonctionnement de l'algorithme 7, nous le déroulons sur l'exemple de la figure 3.38. Dès lors, on suppose que le brin b porte le numéro 1, que les arêtes A , A_1 et A_2 sont respectivement composées des brins $\{1, 2, 3, 4\}$, $\{1, 3, 5, 7\}$ et $\{2, 4, 6, 8\}$ et que le sommet S est formé des brins 5, 6, 7 et 8.

1. L'ensemble des brins constituant l'orbite $O = \langle \alpha_2, \alpha_3 \rangle (b)$ est parcouru. L'objectif est de faire en sorte que le nouveau sommet S hérite des ancêtres associés aux faces et volumes qui leurs est incident (d'où les involutions α_2 et α_3 qui caractérisent O). Donc, le tag de chaque brin b' de O (brin appartenant aux faces et volumes incidents à S) est propagé aux brins $\alpha_0(b')$ et $\alpha_1(\alpha_0(b'))$ qui constituent S . Pour l'exemple de la figure 3.38, O est constitué des brins 1 et 3. À la première itération du parcours, le tag associé au brin 1 est propagé aux brins 5 et 6 ($5 = \alpha_0(1)$ et $6 = \alpha_1(\alpha_0(1))$). De même, à la deuxième itération du parcours, le tag associé au brin 3 est propagé aux brins 7 et 8 ($7 = \alpha_0(3)$ et $8 = \alpha_1(\alpha_0(3))$).
2. Du fait que S est nouvellement créé, il est dépourvu d'ancêtre. Dans les faits, la valeur **ID_NULL** initialise la valeur de l'identifiant *id_anc₀* des *tag_ancêtres* associés aux brins de S . Dans la figure 3.38, les brins de S 5, 6, 7 et 8 sont associés chacun à un *tag_ancêtre* dont le premier identifiant du quadruplet *id_anc₀* est initialisé à **ID_NULL**.
3. Pour la même raison, une nouvelle valeur est attribuée à l'identifiant *id_act₀* de chaque *tag_actuel* associé aux brins de S (5, 6, 7, et 8). Dans la figure 3.38, S est caractérisé par

3.4. IMPLANTATION DES MÉCANISMES DE SUIVI D'ÉVÉNEMENTS

une nouvelle valeur générée par la fonction `Nouveau_ID(0)`. Dans ce cas, elle est égale à 12.

4. L'arête A_1 est caractérisée avec une nouvelle valeur. Cette valeur est attribuée à l'identifiant id_act_1 constituant les $tag_actuels$ associés aux brins de cette arête. Dans la figure 3.38, l'identifiant id_act_1 de chaque tag_actuel associé aux brins numérotés 1, 3, 5 et 7 est mis à jour avec une nouvelle valeur 9 générée par la fonction `Nouveau_ID(1)`.
5. Le même traitement que l'étape précédente est exercé sur l'arête A_2 . Dans la figure 3.38, l'identifiant id_act_1 de chaque tag_actuel associé aux brins numérotés 2, 4, 6 et 8 reçoit une nouvelle valeur 13 générée par la fonction `Nouveau_ID(1)`.

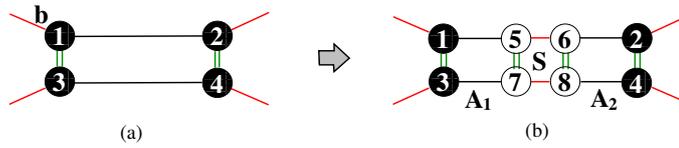


Figure 3.38: Scission d'une arête. (a) Les brins 1, 2, 3 et 4 composent l'arête A avant la scission. (b) A est scindée en A_1 formée des brins 1, 3, 5, 7 et A_2 formée des brins 2, 4, 6, 8. Le sommet S incident à A_1 et A_2 est composé des brins 5, 6, 7 et 8. Le contenu de chaque tag associé à l'un de ces brins est décrit dans la table 3.5.

brin	Avant								Après							
	$tag_ancêtre$				tag_actuel				$tag_ancêtre$				tag_actuel			
1	4	5	2	3	4	5	2	3	4	5	2	3	4	9	2	3
2	8	5	2	3	8	5	2	3	8	5	2	3	8	13	2	3
3	4	5	2	3	4	5	2	3	4	5	2	3	4	9	2	3
4	8	5	2	3	8	5	2	3	8	5	2	3	8	13	2	3
5									-	5	2	3	12	9	2	3
6									-	5	2	3	12	13	2	3
7									-	5	2	3	12	9	2	3
8									-	5	2	3	12	13	2	3

Table 3.5: Structures de tags (tag_actuel et $tag_ancêtre$) associées aux brins de la figure 3.38. Dans cette table, « - » représente la valeur ID_NULL qui peut être associée à un identifiant d'un $tag_ancêtre$.

Au final, les étapes 2 et 3 correspondent au mécanisme $créa_élé$ m appliqué suite à la création de S . De plus, les étapes 1, 4 et 5 implantent le mécanisme $sci_élé$ m appliqué suite à la scission de A . En effet, il y a, d'une part, la propagation de tags associés aux brins de A vers A_1 et A_2 (étape 1) puis, d'autre part, la caractérisation des deux nouvelles arêtes avec deux nouveaux identifiants 9 et 13 (étapes 4 et 5).

3.4.2.3 Scission d'une face

L'opération élémentaire de scission d'une cellule face notée F , se traduit par l'insertion d'une arête notée A , dans F . De ce fait, elle engendre deux nouvelles faces, notées respectivement F_1 et F_2 , incidentes à A . Ainsi, cette opération invoque les mécanismes $sci_élé$ m et $créa_élé$ m car elle met en scène deux événements que sont la scission de F et la création de A .

Au niveau du noyau géométrique, cette opération élémentaire se déploie au travers de la fonction `Scinder_Face` dont le prototype et le fonctionnement sont décrits par :

`Scinder_Face(b1, b2 : Brin) → Brin` : cette fonction permet d'éclater la face incidente aux brins b_1 et b_2 . Elle insère une arête entre b_1 et b_2 de manière à respecter la condition $\alpha_2(\alpha_1(b_1)) = \alpha_0(\alpha_1(b_2))$. Le brin retourné correspond à $\alpha_2(\alpha_1(b_1))$.

Les mécanismes *sci_élé*m et *créa_élé*m sont invoqués suite à l'application de cette opération élémentaire.

Ainsi, la fonction `Nommer_faces_identifiées` décrite par l'algorithme 8 permet d'associer les tags adéquats aux brins constituant les faces F_1 et F_2 . `Nommer_faces_identifiées` prend en paramètres deux brins, b_1 et b_2 , incidents à la face F .

Algorithme. 8 : `Nommer_faces_scindées`

Données : b_1 et b_2 : deux brins constituant les mêmes paramètres que ceux de la fonction `Scinder_Face`.

Résultat : attribution des tags adéquats aux brins constituant les faces F_1 et F_2 résultant de la scission de F .

début

```

1  |   pour chaque  $b' \in \langle \alpha_0, \alpha_2, \alpha_3 \rangle (\alpha_1(b_1))$  faire
    |   |   Dupliquer_Tag( $\alpha_1(b')$ ,  $\langle \rangle (\alpha_0(b'))$ ) ;
2  |   Remplacer_ID_Ancêtre( $\langle \alpha_0, \alpha_2, \alpha_3 \rangle (\alpha_1(b_1))$ , 1, ID_NULL) ;
3  |   Remplacer_ID_Actuel( $\langle \alpha_0, \alpha_2, \alpha_3 \rangle (\alpha_1(b_1))$ , 1, Nouveau_ID(1)) ;
4  |   Remplacer_ID_Actuel( $\langle \alpha_0, \alpha_1, \alpha_3 \rangle (b_1)$ , 2, Nouveau_ID(2)) ;
5  |   Remplacer_ID_Actuel( $\langle \alpha_0, \alpha_1, \alpha_3 \rangle (b_2)$ , 2, Nouveau_ID(2)) ;

```

fin

Pour illustrer le fonctionnement de l'algorithme 8, nous le déroulons sur l'exemple de la figure 3.39. On suppose alors que les brins b_1 et b_2 , qui respectent la condition $\alpha_2(\alpha_1(b_1)) = \alpha_0(\alpha_1(b_2))$, portent respectivement les numéros 1 et 4. On suppose également que les faces F , F_1 et F_2 sont respectivement composées des brins $\{1, 2, 3, 4, 9, 10, 11, 12\}$, $\{1, 3, 5, 6, 9, 10\}$ et $\{2, 4, 7, 8, 11, 12\}$ et que l'arête A est formée des brins 5, 6, 7 et 8.

1. L'ensemble des brins constituant l'arête insérée A est parcouru (soit b' l'un de ces brins). Chaque brin b' reçoit une copie des tags associés au brin $\alpha_1(b')$. Ceci permet à l'arête insérée d'hériter des ancêtres de son voisinage topologique. Dans la figure 3.39, les brins 1, 2, 3, 4 propagent respectivement leurs tags aux brins numérotés 5, 7, 6, 8.
2. La valeur de l'identifiant id_{anc_1} de chaque *tag_ancêtre* associé aux brins de A est mise à jour avec la valeur ID_NULL car cette dernière est une arête nouvellement créée. Dans la figure 3.39, les brins 5, 6, 7 et 8 portent un *tag_ancêtre* dont l'identifiant id_{anc_1} est égal à ID_NULL.
3. Une nouvelle valeur est attribuée à l'identifiant id_{act_1} de chaque *tag_actuel* associé aux brins de l'arête A pour la même raison. Dans la figure 3.39, une nouvelle valeur est attribuée à l'identifiant id_{act_1} de chaque *tag_actuel* associé aux brins 5, 6, 7 et 8. Elle est générée par la fonction `Nouveau_ID(1)` et elle est égale à 21.
4. Une nouvelle valeur est également attribuée à l'identifiant id_{act_2} du *tag_actuel* à chaque brin de F_1 . Dans la figure 3.39, l'identifiant id_{act_2} de chaque *tag_actuel* associé aux brins

3.4. IMPLANTATION DES MÉCANISMES DE SUIVI D'ÉVÉNEMENTS

numérotés 1, 3, 5, 6, 9 et 10 est mis à jour avec une valeur qui n'est pas encore attribuée. Cette valeur est générée par la fonction `Nouveau_ID(2)` et elle est égale à 6.

5. Le même traitement est effectué sur la face F_2 . Dans l'exemple de la figure 3.39, l'identifiant id_{act_2} de chaque tag_{actuel} associé aux brins numérotés 2, 4, 7, 8, 11 et 12 prend une nouvelle valeur. Cette valeur est générée par la fonction `Nouveau_ID(2)` et elle est égale à 10.

Au final, les étapes 2 et 3 correspondent au mécanisme *créa_élé*m appliqué sur le cas de la cellule A . Ces étapes n'associent à l'identifiant 21 caractérisant A aucun ancêtre. De plus, les étapes 1, 4 et 5 implantent le mécanisme *sci_élé*m appliqué sur le cas de la cellule F . En effet, la propagation (resp. la mise à jour) de tags effectuée lors de l'étape 1 (resp. lors des étapes 4 et 5) permet aux cellules F_1 et F_2 d'hériter de l'ancêtre de la cellule F , à savoir l'identifiant id_{anc_2} de valeur 2 (resp. d'être caractérisées par de nouvelles valeurs d'identifiants 6 et 10).

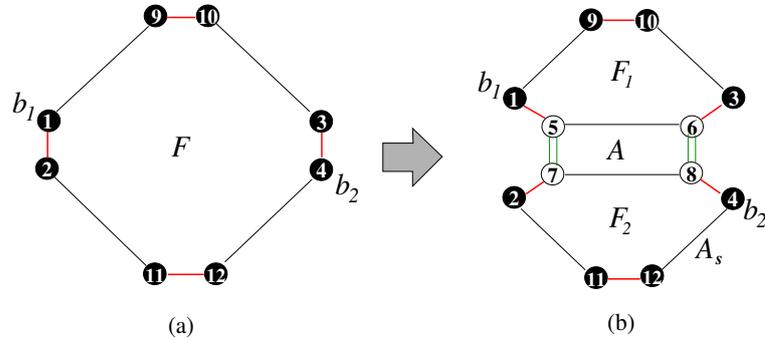


Figure 3.39: Scission d'une face. (a) Les brins $\{1, 2, 3, 4, 9, 10, 11, 12\}$ appartiennent à la face F avant la scission. (b) F est scindée en F_1 , formée des brins $\{1, 3, 5, 6, 9, 10\}$ et F_2 , formée des brins $\{2, 4, 7, 8, 11, 12\}$. L'arête A incidente à F_1 et F_2 est composée des brins $\{5, 6, 7, 8\}$. Le contenu de chaque tag associé à l'un de ces brins est décrit dans la table 3.6.

3.4.2.4 Identification de faces

L'intersection de deux faces coplanaires produit des artefacts qui surviennent lors de la mise à jour de la topologie [Gui06]. En effet, ces artefacts apparaissent sous la forme de faces doubles et doivent par conséquent être fusionnés afin d'obtenir une subdivision valide de l'espace 3D (voir la figure 3.40).

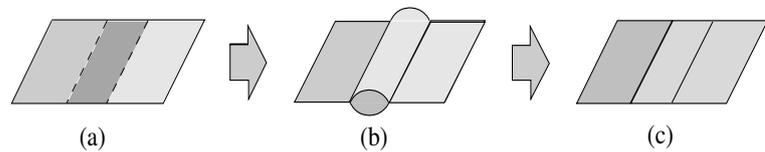


Figure 3.40: Correction de la topologie suite à l'intersection de deux faces coplanaires. (a) Faces coplanaires ayant une partie commune. (b) Face double apparaissant suite au calcul d'intersection. (c) Suppression d'une des deux faces communes.

La fusion entre deux faces homéomorphes, ayant des plongements identiques, est réalisée *via* l'opération élémentaire d'identification de faces. Cette opération prend en paramètre deux faces

brin	Avant								Après							
	<i>tag_ancêtre</i>				<i>tag_actuel</i>				<i>tag_ancêtre</i>				<i>tag_actuel</i>			
1	4	5	2	3	4	5	2	3	4	5	2	3	4	5	6	3
2	4	9	2	3	4	9	2	3	4	9	2	3	4	9	10	3
3	8	13	2	3	8	13	2	3	8	13	2	3	8	13	6	3
4	8	17	2	3	8	17	2	3	8	17	2	3	8	17	10	3
5									4	-	2	3	4	21	6	3
6									8	-	2	3	8	21	6	3
7									4	-	2	3	4	21	10	3
8									8	-	2	3	8	21	10	3
9	12	5	2	3	12	5	2	3	12	5	2	3	12	5	6	3
10	12	13	2	3	12	13	2	3	12	13	2	3	12	13	6	3
11	16	9	2	3	16	9	2	3	16	9	2	3	16	9	10	3
12	16	17	2	3	16	17	2	3	16	17	2	3	16	17	10	3

Table 3.6: Structures de tags (*tag_actuel* et *tag_ancêtre*) associées aux brins de la figure 3.39. Dans cette table, « - » représente la valeur *ID_NULL* qui peut être associé à un identifiant d'un *tag_ancêtre*.

(par exemple, les faces F_1 et F_2 de la figure 3.41(b)) et restitué une nouvelle face¹⁹ (par exemple la face F de la figure 3.41(c)). Nous avons choisi, compte tenu des artefacts pouvant apparaître, de supprimer une des deux faces en entrée et de considérer la face restante comme résultant de l'opération élémentaire d'identification de faces.

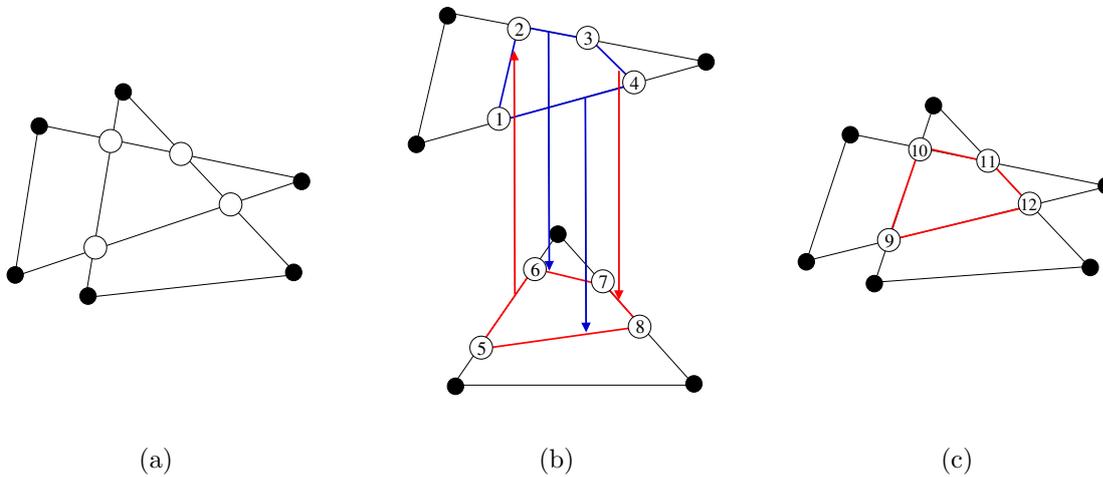


Figure 3.41: Intersection de faces coplanaires. (a) Calcul et création des points d'intersection sur chacune des faces (les cercles en blanc correspondent aux points d'intersection nouvellement créés). (b) Apparition des faces en double F_1 et F_2 après la création des segments d'intersection à l'intérieur de chaque face (les segments sont en bleu pour la face F_1 et en rouge pour la face F_2). (c) Identification des faces en double F_1 et F_2 en une nouvelle face F .

19. Les faces F_1, F_2 et F sont identiques d'un point de vue géométrique et topologique.

3.4. IMPLANTATION DES MÉCANISMES DE SUIVI D'ÉVÉNEMENTS

L'identification de faces engendre la fusion des sommets et des arêtes ayant les mêmes plongements, utilisant de fait le mécanisme *fus-élé*. Dans la figure 3.41, les sommets 1, 2, 3, 4 (resp. les arêtes [1, 2], [2, 3], [3, 4], [4, 1]) de la face F_1 sont fusionnés respectivement avec les sommets 5, 6, 7, 8 (resp. avec les arêtes [5, 6], [6, 7], [7, 8], [8, 5]) de la face F_2 afin de générer dans l'ordre les sommets 9, 10, 11, 12 (resp. les arêtes [9, 10], [10, 11], [11, 12], [12, 9]) constituant la face F .

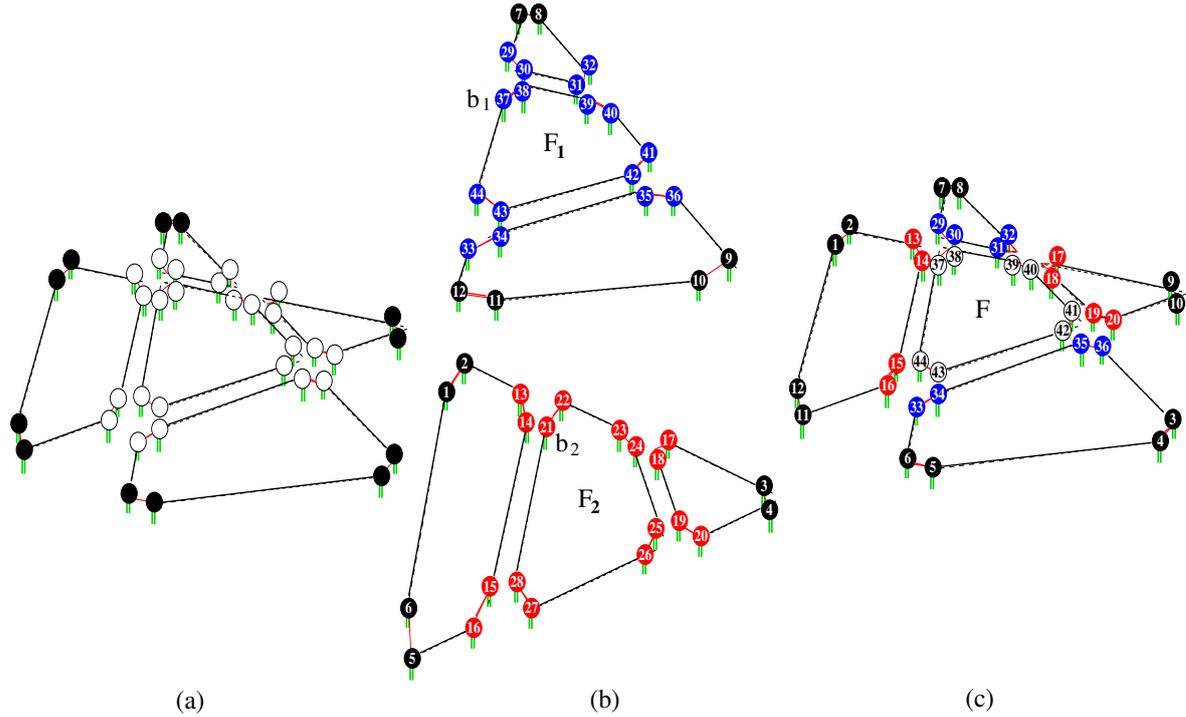


Figure 3.42: Intersection de faces coplanaires (topologie correspondant à la configuration de la figure 3.41). (a) Calcul et création des points d'intersection sur chaque face (les cercles en blanc représentent les brins nouvellement créés). (b) Mise en évidence des faces en double F_1 et F_2 après la création des segments d'intersection à l'intérieur de chaque face (les brins $\{29, 30, 37, 38\}$, $\{31, 32, 39, 40\}$, $\{35, 36, 41, 42\}$, $\{33, 34, 43, 44\}$, $\{17, 18, 23, 24\}$, $\{19, 20, 25, 26\}$, $\{15, 16, 27, 28\}$, $\{13, 14, 21, 22\}$ forment respectivement les sommets 2, 3, 4, 1, 7, 8, 5, 6 de la figure 3.41(b)). (c) Identification des faces F_1 et F_2 en une nouvelle face F (les brins $\{37, 38, 13, 14, 29, 30\}$, $\{39, 40, 17, 18, 31, 32\}$, $\{41, 42, 19, 20, 35, 36\}$, $\{43, 44, 33, 34, 15, 16\}$ forment respectivement les sommets 10, 11, 12, 9 de la figure 3.41(c)). Le contenu des tags associés aux brins constituant les faces F_1 , F_2 et F est décrit dans la table 3.7.

La fonction `Nommer_faces_identifiées` (voir l'algorithme 9) permet de transcrire tous les événements de fusion en appliquant le mécanisme *fus-élé* chaque fois que cela est nécessaire. `Nommer_faces_identifiées` prend en paramètres deux brins b_1 et b_2 respectivement incidents aux faces F_1 et F_2 . Elle permet d'associer les structures de tags aux brins constituant la face F résultant de l'opération élémentaire d'identification des faces F_1 et F_2 .

Pour appréhender le fonctionnement de l'algorithme 9, étudions l'exemple de la figure 3.42. La fonction `Nommer_faces_identifiées` intervient avant l'exécution de l'opération élémentaire d'identification des faces. On suppose que les brins b_1 et b_2 portent respectivement les numéros 21 et 37 et que les faces F , F_1 et F_2 sont respectivement composées des brins $\{37, 38, 39, 40, 41, 42, 43,$

Algorithme. 9 : Nommer_faces_identifiées

Données : ■ un brin b_1 de la face F_1 ;
 ■ un brin b_2 de la face F_2 .

Résultat : attribution des tags adéquats aux brins de la face F résultant de l'identification des faces F_1 et F_2 ($F = F_1$).

début

```

1 | ( $it_1, it_2$ )  $\leftarrow$  ( $b_1, b_2$ );
   | répéter
2 |   Remplacer_Tag_Ancêtre( $it_1, it_2$ ) ;
3 |   ( $it_1, it_2$ )  $\leftarrow$  ( $\alpha_0(\alpha_1(it_1)), \alpha_0(\alpha_1(it_2))$ ) ;
4 |   Remplacer_ID_Actuel( $\langle \rangle_0(it_1), 0, \text{Nouveau\_ID}(0)$ ) ;
5 |   Remplacer_ID_Actuel( $\langle \rangle_1(it_1), 1, \text{Nouveau\_ID}(1)$ ) ;
   | jusqu'à  $it_1 = b_1$  ;
6 |   Remplacer_ID_Actuel( $\langle \rangle_2(it_1), 2, \text{Nouveau\_ID}(2)$ ) ;
   | fin

```

44}, {21, 22, 23, 24, 25, 26, 27, 28} et {37, 38, 39, 40, 41, 42, 43, 44} (la face résultante F correspond à la face F_1 non supprimée).

1. Les deux itérateurs it_1 et it_2 sont respectivement initialisés avec les brins b_1 et b_2 afin de débiter le parcours des faces F_1 et F_2 . Dans la figure 3.42, it_1 et it_2 sont respectivement initialisés avec les brins portant les numéros 21 et 37. De plus, le parcours des faces F_1 et F_2 passe respectivement par les brins {37, 39, 41, 43} et {21, 23, 25, 27} (un brin sur deux est parcouru, correspondant respectivement aux orbites $\langle \alpha_1 \alpha_0 \rangle (b_1)$ et $\langle \alpha_1 \alpha_0 \rangle (b_2)$).
2. Le *tag_ancêtre* associé au brin désigné par it_1 est mis à jour, à chaque étape du parcours, avec le *tag_ancêtre* associé au brin désigné par it_2 . L'objectif de cette mise à jour est de signifier que la face F provient de la fusion de F_1 et F_2 . Elle permet également de rendre compte de la fusion de chaque sommet et de chaque arête des deux faces en un sommet et une arête de la face résultant de l'opération d'identification. Dans la figure 3.42, les *tag_ancêtres* associés aux brins {37, 39, 41, 43} sont respectivement mis à jour avec les valeurs $\{(-, -, 34, 7), (-, 57, 34, 7), (-, -, 34, 7), (-, 69, 34, 7)\}$ des *tag_ancêtres* associés aux brins {21, 23, 25, 27}.
3. L'itération suivante du parcours est atteinte. Ce dernier s'effectue d'une manière synchrone sur les faces F_1 et F_2 .
4. La valeur de chaque identifiant $id.act_0$ d'un *tag_actuel* associé aux brins {37, 39, 41, 43} est mise à jour avec une nouvelle valeur. Pour la figure 3.42, ces identifiants sont générés par la fonction $\text{Nouveau_ID}(0)$ et correspondent, dans l'ordre, aux valeurs 116, 120, 124 et 128.
5. La valeur de chaque identifiant $id.act_1$ d'un *tag_actuel* associé aux brins {37, 39, 41, 43} est également mise à jour avec une nouvelle valeur. Pour la figure 3.42, ces identifiants sont générés par la fonction $\text{Nouveau_ID}(1)$ et correspondent, dans l'ordre, aux valeurs 249, 237, 241 et 245.
6. À la fin du parcours, la valeur de l'identifiant $id.act_2$ du *tag_actuel* associé à chaque brin

3.5. CONCLUSION

constituant la face résultante F est mise à jour avec une nouvelle valeur. Dans la figure 3.42, cette valeur est générée par la fonction `Nouveau_ID(2)` et est égale à 138.

<i>Face F_1</i>			<i>Face F_2</i>		
brin	<i>tag_ancêtre</i>	<i>tag_actuel</i>	brin	<i>tag_ancêtre</i>	<i>tag_actuel</i>
21	(-, -, 34, 7)	(64, 109, 62, 7)	37	(-, 5, 14, 3)	(56, 89, 70, 3)
22	(-, 57, 34, 7)	(64, 121, 62, 7)	38	(-, ;, 14, 3)	(56, 145, 70, 3)
23	(-, 57, 34, 7)	(72, 121, 62, 7)	39	(-, ;, 14, 3)	(68, 145, 70, 3)
24	(-, -, 34, 7)	(72, 149, 62, 7)	40	(-, 33, 14, 3)	(68, 129, 70, 3)
25	(-, -, 34, 7)	(80, 149, 62, 7)	41	(-, 33, 14, 3)	(76, 129, 70, 3)
26	(-, 69, 10, 3)	(80, 414, 62, 7)	42	(-, -, 14, 3)	(76, 153, 70, 3)
27	(-, 69, 34, 7)	(60, 141, 62, 7)	43	(-, -, 14, 3)	(52, 153, 70, 3)
28	(-, -, 34, 7)	(60, 109, 62, 7)	44	(-, 5, 14, 3)	(52, 89, 70, 3)

<i>Face F</i>		
brin	<i>tag_ancêtre</i>	<i>tag_actuel</i>
37	(-, -, 34, 7)	(116, 249, 138, 19)
38	(-, -, 14, 3)	(116, 237, 138, 19)
39	(-, 57, 34, 7)	(120, 237, 138, 19)
40	(-, 33, 14, 3)	(120, 241, 138, 19)
41	(-, -, 34, 7)	(124, 241, 138, 19)
42	(-, -, 14, 3)	(124, 245, 138, 19)
43	(-, 69, 34, 7)	(128, 245, 138, 19)
44	(-, 5, 14, 3)	(128, 249, 138, 19)

Table 3.7: Structures de tags (*tag_actuel* et *tag_ancêtre*) associées aux brins de la figure 3.42. Dans cette table, « - » représente la valeur `ID_NULL` qui peut être associé à un identifiant d'un *tag_ancêtre*.

Finalement, chaque itération de l'algorithme 9 permet de propager les ancêtres des faces F_1 et F_2 à la face F . De la même façon, les cellules constituant le bord de F héritent des ancêtres associés aux cellules constituant les bord de F_1 et F_2 (étape 2 de l'algorithme 9). De plus, comme il y a apparition d'une multitude de nouvelles cellules (la face F et toutes les arêtes et sommets constituant son bord), il est donc nécessaire d'attribuer à chacune de ces entités de nouveaux identifiants compatibles avec leurs dimension. Cette tâche est remplie par les étapes 4, 5 et 6 du même algorithme.

3.5 Conclusion

Après avoir décrit le problème de la nomination persistante, nous avons montré les limites des solutions proposées pour le résoudre et la nécessité de suivre précisément l'évolution des entités topologiques au sein du noyau géométrique d'un système paramétrique. Dès lors, pour déterminer si des cellules (sommets, arêtes, faces, volumes, etc.) ont été créées, supprimées, scindées, fusionnées ou modifiées de manière générale, nous avons proposé un ensemble de mécanismes de suivi de ces événements.

Ces mécanismes sont *créa_élém*, *sup_élém*, *sci_élém*, *fus_élém* et *modif_élém*. Ils permettent la détection d'une liste d'événements survenus sur les entités topologiques d'un modèle géométrique au cours d'une opération de modélisation comme suit :

- le mécanisme *créa_élém* permet de suivre l'événement de création d'entités. Il n'associe l'identifiant de l'entité nouvellement créée à aucun ancêtre ;
- le mécanisme *sup_élém* permet de suivre l'événement de suppression d'entités. Il n'associe l'identifiant de l'entité supprimée à aucun autre identifiant ;
- le mécanisme *sci_élém* permet de suivre l'événement de scission d'entités. Il n'associe l'identifiant de l'entité scindée aux identifiants des différentes entités issues de cette scission ;
- le mécanisme *fus_élém* permet de suivre l'événement de fusion d'entités. Il n'associe les identifiants des différentes entités fusionnées à l'identifiant de l'entité résultant de cette fusion ;
- le mécanisme *modif_élém* permet de suivre l'événement de modification de la structure topologique d'une entité. Il associe l'identifiant de l'entité d'origine à l'identifiant de l'entité modifiée.

Les événements ainsi détectés sont regroupés au sein de la structure du « journal de bord ». À cet effet, nous avons décomposé toute opération de modélisation géométrique de haut niveau (dite *globale*) en un ensemble fini d'*opérations élémentaires*. Chacune d'entre elles est analysée par avance afin de connaître les transformations topologiques locales qu'elle engendre sur le modèle géométrique. Ensuite, en fonction de ces transformations, les mécanismes de suivi d'événements sont associés aux opérations élémentaires. Ces derniers sont définis pour les entités topologiques de toute dimension. De plus, ils sont décrits de façon indépendante du modèle géométrique utilisé en s'appuyant uniquement sur la base topologique sous-jacente.

Nous avons choisi dans le cadre de nos travaux le modèle des cartes généralisées. L'étude de ce modèle a permis de mettre en évidence quatre opérations de modélisation sur la base desquelles peuvent être définies toutes les autres : la suppression, la contraction, l'insertion et l'éclatement. Nous avons rappelé la définition de la suppression et de l'éclatement et nous avons défini formellement l'insertion et l'éclatement.

Cette définition nous a permis par la suite de proposer une implantation des mécanismes de suivi d'événements sur le modèle des cartes généralisées. La plate-forme mise en œuvre dans le cadre de cette implantation se base principalement sur la structure de « tag ». Cette structure caractérise, au travers des *tag_actuels*, chaque entité topologique de l'instance courante du modèle géométrique ; et lie au travers des *tags_ancêtres* ces entités à celles existantes au tout début d'une opération de modélisation globale. Ces structures étant définies, la plate-forme se soumet aux mécanismes proposés afin de répartir les tags au niveau de chaque brin. L'exemple de l'opération globale de co-raffinement illustre parfaitement ce principe. En effet, de part son importance et sa complexité dans le domaine de la modélisation géométrique, celle-ci constitue un très bon cas d'étude. Le journal de bord est généré sur la base de parcours de brins et de reconnaissance d'événements. La solution implantée nous a permis de mettre en place des mécanismes génériques de suivi d'événements capables de supporter des applications et des domaines divers et variés, et de séparer la partie effectuant les traitements topologiques de celle supportant les mécanismes de génération du journal de bord.

Le journal de bord ainsi défini est indispensable pour de nombreuses opérations de haut niveau, et en particulier pour notre système hiérarchique de nomination persistante. En effet, le journal de bord permet, à chaque étape, de tracer l'évolution des entités topologiques et donc de

3.5. CONCLUSION

faire le lien entre chaque entité contingente et, s'il existe, son ancêtre invariant. Ceci est motivé par le fait que cet aspect invariant constitue l'élément dorsal de tout système de nomination qui se veut persistant.

Chapitre 4

Systeme hiérarchique de nomination persistante

Sommaire

4.1	Introduction	105
4.2	Nomination persistante d'arêtes	106
4.2.1	Trois point-clés pour la caractérisation des arêtes	107
4.2.2	Caractérisation des arêtes	110
4.2.3	Appariement des arêtes	118
4.3	Systeme de nomination hiérarchique	125
4.3.1	Notion d'agrégat	125
4.3.2	Caractérisation des agrégats	127
4.3.3	Appariement des agrégats	131
4.4	Conclusion	133

4.1 Introduction

La modélisation paramétrique pose, entre autres problème, celui de la nomination persistante. Nous avons décrit dans le deuxième chapitre la genèse des systèmes paramétriques. Dès lors, nous avons expliqué que la difficulté dans ce type de modèle réside dans l'identification d'entités dans un modèle initial. Puis, elle s'accroît lors de l'appariement entre les entités entre ces entités initiales et celles du modèle réévalué dont la géométrie et la topologie évoluent.

Pour résoudre ce problème, plusieurs solutions ont été proposées. Ces solutions reposent sur un certain nombre de concepts communs. Parmi ces concepts, nous pouvons citer la distinction entre les entités invariantes et les entités contingentes. De part leurs caractères stables, les entités invariantes constituent les éléments de base autour desquelles s'articule la nomination des entités contingentes, plus volatiles. Tout d'abord, les entités contingentes sont caractérisées par la donnée de leurs voisinages topologiques constituées d'entités invariantes. Ensuite, les processus d'appariements utilisent diverses méthodes pour comparer ces noms et déterminer les entités les plus ressemblantes. En pratique, ce voisinage ne comprend pas que des entités invariantes. Par conséquent, les solutions proposées remplacent les entités contingentes par leurs ancêtres invariants.

Dans la liste des entités topologiques constituant le voisinage topologique, les éléments retenus sont les faces en dimension 3. Chaque approche définie dans ce domaine considère, en effet, que les faces sont les entités les plus stables comparée aux sommets et aux arêtes. Ces approches considèrent, à juste titre, qu'en $3D$, une face a toujours un ancêtre invariant alors qu'une arête ou un sommet peut résulter de l'intersection de faces ou d'arêtes de façon non prévisible. Ceci implique une structure de nom et un mécanisme d'appariement spécifique à chaque type d'entité. D'où le caractère hétérogène des mécanismes de nomination déployés. Par conséquent, lorsque nous envisageons de généraliser les procédés de caractérisation et d'appariement aux entités et aux agrégats d'entités de toute dimension, nous nous sommes retrouvés limité.

Donc, afin d'homogénéiser les mécanismes de nomination à toutes les entités et les agrégats d'entités en dimension 3, nous avons choisis les arêtes comme élément de base d'un système hiérarchique de nomination persistante. La nécessité est alors de définir un système de caractérisation et d'appariement assez robuste pour les arêtes. C'est ce que nous allons détailler dans la prochaine section. Ensuite, la caractérisation et l'appariement des entités et des agrégats d'entités, de toute dimension, s'effectue en exploitant celui des arêtes et en utilisant le fait que ces dernières peuvent se définir en termes d'un ensemble d'arêtes (voir la figure 4.1). Nous décrivons la caractérisation et l'appariement des entités et des agrégats d'entités de dimension supérieure dans la troisième section.

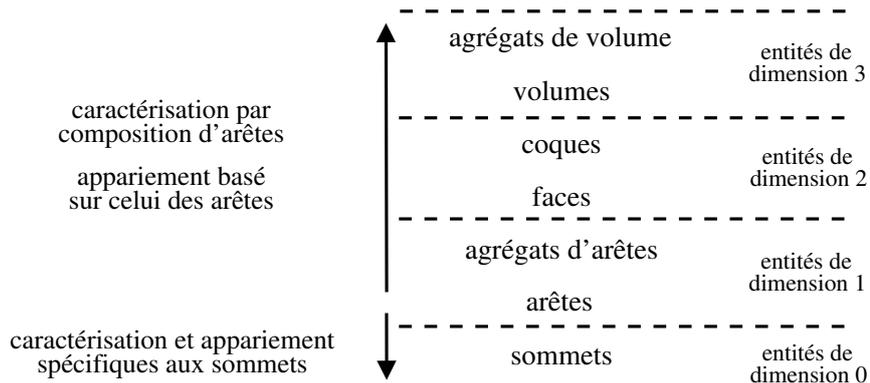


Figure 4.1: Schéma hiérarchique de nomination persistante basé sur les arêtes.

4.2 Nomination persistante d'arêtes

Communément à toutes les solutions présentées dans la littérature, une arête est présentée comme le résultat de l'intersection de deux faces et dont les bords sont représentés par deux ensembles de faces adjacentes. Donc, le nom d'une arête correspond *toujours* au quadruplet :

$$\begin{aligned}
 &< \text{numéro d'étape} ; \\
 &\text{ensemble des faces incidentes} ; \\
 &\text{ensemble des faces incidentes au premier sommet de l'arête}; \\
 &\text{ensemble des faces incidentes au second sommet de l'arête} >
 \end{aligned}
 \tag{4.1}$$

Après la réévaluation du modèle initial, la topologie change, ce qui entraîne une modification des arêtes contenues dans le modèle. L'appariement entre les arêtes des modèles initial et

réévalué est alors effectué en comparant les noms de ces arêtes. Cette comparaison porte sur les éléments du quadruplet constituant leur nom. De façon intuitive, elle s'effectue en prenant en compte toutes les faces constituant le nom d'une arête. Cependant, un appariement basé sur une comparaison, partielle ou totale, de ces éléments ne donne pas des résultats satisfaisants. Nous pouvons effectivement le constater lorsque la comparaison est faite sur la donnée :

- des faces incidentes à l'arête considérée et de toutes les faces incidentes aux sommets bordant cette même arête : ce critère est très restrictif puisqu'il implique un appariement entre l'arête du modèle initial et au plus une arête du modèle réévalué. Ainsi, dans la figure 4.2(b), aucune des arêtes e_4 et e_5 ne peut être mise en correspondance avec l'arête e_2 car les deux premières n'ont chacune qu'une seule face incidente à leurs bords parmi les deux faces incidentes aux bords de la troisième ;
- des faces incidentes à l'arête considérée et au moins une face incidente à un des sommets bordant cette même arête : ce critère reste restrictif puisque dans le cas de fortes scissions d'arêtes, les faces incidentes aux sommets de certaines arêtes peuvent être totalement différentes. La figure 4.2(c) montre l'exemple de l'arête e_8 dont les faces incidentes à ses bords sont totalement différentes des faces incidentes aux bords de e_2 ;
- des faces incidentes *uniquement* à l'arête considérée : ce critère, en revanche, est très lâche puisque toutes les arêtes ayant le même couple de faces incidentes sont appariées. Même pour une réévaluation à l'identique (c'est-à-dire où il n'y a aucun changement de paramètres), le résultat peut être erroné. Par exemple, si le modèle de la figure 4.2(a) est réévalué à l'identique, les arêtes e_1 et e_2 du modèle réévalué sont appariées à l'arête e_2 du modèle initial au lieu de la seule arête e_2 du modèle réévalué.

Entre le premier cas (très restrictif) et le dernier (plus lâche), d'autres configurations purement combinatoires peuvent être envisagées (cas où une seule face incidente à l'arête est considérée, ou bien uniquement les faces incidentes aux bords des arêtes, etc.). Malheureusement, ces critères ne donnent pas des résultats satisfaisants.

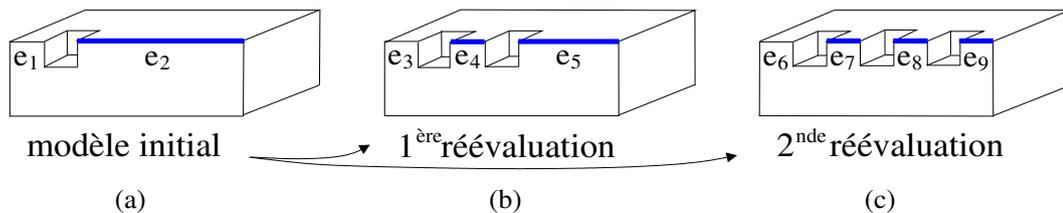


Figure 4.2: (a) L'arête e_2 est référencée dans l'objet initial. (b) Premier cas de réévaluation : l'arête e_2 est scindée en deux arêtes par une rainure. (c) Second cas de réévaluation : l'arête e_2 est scindée en trois.

4.2.1 Trois point-clés pour la caractérisation des arêtes

Après étude du problème de la nomination persistante et analyse des différentes approches proposées par la communauté scientifique, nous avons extraits, dans le premier chapitre, un certain nombre de concepts communs et de points différents. Ces éléments nous permettent, en les replaçant dans le contexte de la nomination persistante d'arêtes d'établir trois point-clés qui nous semblent essentiels pour élaborer une approche supportant une caractérisation et un appariement robuste des arêtes.

4.2.1.1 Utilisation d'entités invariantes

Le nom d'une arête, tel que décrit par le quadruplet 4.1, ne doit contenir que des références à des entités invariantes. En effet, lorsque les faces incidentes à une arête donnée ou les faces incidentes aux bords de cette même arête sont contingentes, les résultats du processus d'appariement peuvent être inexacts. Par conséquent, la recherche des ancêtres invariants de ces entités contingentes s'avère nécessaire afin de remplacer les informations contingentes par celles invariantes. Dans l'exemple de la figure 4.3(a) et (b), les arêtes e_2 et e_4 sont incidentes à différents couples de faces. Le premier est constitué de la face du haut du bloc et de la face f_3 alors que le second est constitué de la face du haut du bloc et de la face f_5 . Dès lors, la comparaison des noms de ces arêtes ne donne aucun résultat. Pour éviter cela, la référence aux faces f_3 et f_5 est remplacée dans les noms de e_2 et e_4 par leur ancêtre invariant, la face f .

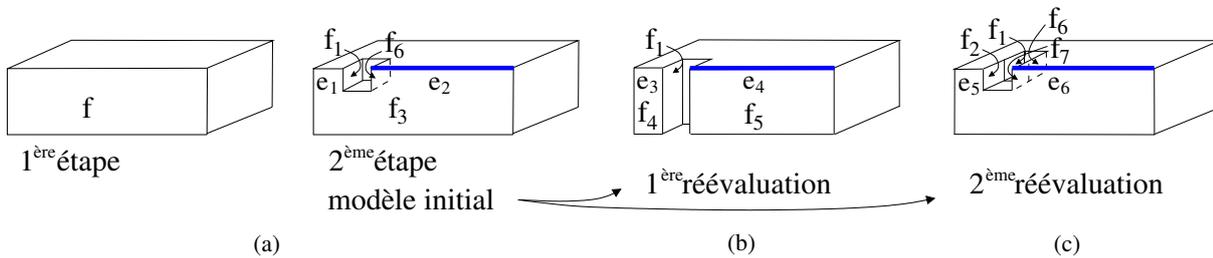


Figure 4.3: (a) Création du bloc par extrusion de la face f (1^{ère} étape) puis ajout de la rainure sur la face du haut (2^{ème} étape). (b) Premier cas de réévaluation : la rainure verticale scinde la face f_3 en deux faces f_4 et f_5 . (c) Second cas de réévaluation : la rainure est plus profonde et sa coque latérale (composée, entre autres des faces f_1 et f_2) est la même que celle de la rainure du modèle initial.

4.2.1.2 Exploitation des informations issues de différents niveaux de granularité

La caractérisation des bords d'une arête par deux ensembles de faces ne permet pas l'expression des différentes sémantiques associées aux opérations de modélisation. La conséquence est que l'appariement conduit à des résultats inexacts. Pour illustrer ce problème, prenons le cas de la figure 4.3(c). Dans cet exemple, l'objet du modèle réévalué subit la même rainure que celle appliquée sur l'objet du modèle initial. La différence est que dans le second cas, la rainure est plongée un peu plus profondément dans le bloc. Les arêtes e_5 et e_6 ne sont alors plus incidentes aux faces f_1 et f_6 du modèle initial mais à deux autres faces. Il s'agit respectivement des faces f_2 et f_7 . Dans ce cas, le bord de e_7 incident à f_7 est différent du bord de e_2 incident à f_6 alors qu'intuitivement, e_2 correspond à e_7 puisque ces deux arêtes sont à droite de la même rainure. Pour supporter cette sémantique, le nom des arêtes doit intégrer des entités invariantes de haut niveau telles que les coques ou les volumes.

4.2.1.3 Utilisation d'informations géométriques

Comme nous pouvons le constater, la caractérisation d'une arête utilisant exclusivement son voisinage topologique ne suffit pas à lever les ambiguïtés qui apparaissent dans le cas non-planaire. En effet, lorsque les volumes bordés par des faces non-planaires interagissent, deux entités, voir plus, peuvent présenter la même caractérisation. C'est le cas, par exemple, des

4.2. NOMINATION PERSISTANTE D'ARÊTES

arêtes e_1 et e_2 de la figure 4.4 : e_1 et e_2 sont incidentes aux faces f_1 et f_2 et leurs sommets extrémités sont incidents aux faces f_3 et f_4 .

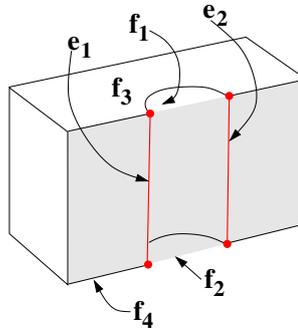


Figure 4.4: Arêtes avec le même voisinage topologique.

De plus, lorsqu'on envisage la nomination des entités de dimension plus élevée (4 dimensions, voire au-delà), se reposer uniquement sur les faces entraîne des limitations. En effet, comme dans le cas des arêtes en $3D$, l'intersection de deux entités de dimension n , $n \geq 4$ (hypervolumes en $4D$), peut engendrer des faces sans sources. La solution est alors de remplacer les faces par les entités de dimension $n - 1$ comme pour le cas des faces en $3D$. Dans ce cas, l'intersection de deux entités de dimension $n - 1$ génère plus d'une arête et la seule donnée de ces deux entités ne suffit pas à lever l'ambiguïté de leurs caractérisations.

Dans la perspective de lever les ambiguïtés dues aux faces non-planaires, les solutions proposées dans la littérature (notamment, les solutions de [WZZZ01], [WN05] et [BNB05] présentées dans la section 2.3.2.2) utilisent des informations géométriques supplémentaires (les demi-espaces pour Bidarra, le gradient pour Wang ou les coordonnées paramétriques des points représentant les entités à nommer pour Wu). Nous adoptons le même principe pour lever les ambiguïtés lorsque les arêtes sont caractérisées par le même voisinage topologique. Les informations géométriques ont deux objectifs :

- caractériser de façon unique et non ambigu les entités concernées (dans notre cas les arêtes) ;
- conduire à un appariement valide.

La structure de nom ainsi complétée assure, dans les solutions précédemment citées, l'unicité de la référence à une arête. Toutefois, l'information géométrique de cette structure ne garantit pas la pertinence des appariements obtenus (voir les lacunes de chaque approche présentée dans la section 2.3.2.2).

Dans les deux prochaines sections, nous proposons, d'une part, une structure de nom pour une arête qui prend en compte toutes ces exigences (informations invariantes, agrégatives et géométriques), et, d'autre part un processus d'appariement suffisamment complet pour proposer des méthodes d'appariement fournissant des résultats au moins aussi satisfaisants que ceux des méthodes de la littérature, et suffisamment complet pour supporter des configurations qui ne sont pas traitées par ces méthodes.

4.2.2 Caractérisation des arêtes

En général, dans un processus de modélisation paramétrique, les objets modélisés sont conçus par l'interaction entre une structure topologique connue (structure invariante issue d'une caractéristique de forme par exemple) et l'instance courante du modèle géométrique à une étape déterminée. Cette instance comprend des entités topologiques apparues et d'autres transformées par scission, fusion ou modification d'autres entités ayant existé à une étape antérieure. Ces entités contingentes doivent être caractérisées. Nous sommes donc en face de deux types de nomination à définir : une pour les arêtes et les agrégats invariants d'entités et une autre pour les arêtes et les agrégats contingents d'entités. Nous traitons dans un premier temps la nomination des arêtes invariantes et contingentes. En nous appuyant sur cette méthodologie, nous traitons dans la section 4.3 la nomination des agrégats invariants d'entités et contingentes de toute dimension.

4.2.2.1 Caractérisation des arêtes invariantes

Pour caractériser les arêtes invariantes, nous allons nous baser sur la structure invariante de la topologie. Cette caractérisation est robuste par rapport aux variations géométriques (dimension, position, etc.) puisque ces variations n'entraînent pas de transformations au niveau de la structure topologique. À partir de là, nous proposons deux méthodes pour caractériser les arêtes invariantes selon la nature de l'opération de modélisation :

- la première méthode utilise un parcours topologique (toujours le même) de l'objet résultant d'une opération de modélisation¹ (en général, une opération de création de formes classiques telles que les blocs, les cylindres, etc.). À l'issue de ce parcours, un numéro d'ordre est attribué à chaque arête (voir la figure 4.5(a)). Par conséquent, le nom de chaque arête invariante est constitué de la manière suivante : $\langle \text{numéro d'étape}, \text{numéro d'ordre} \rangle$;
- la seconde se base sur les références aux entités $2D$ ayant fait l'objet d'une extrusion ou d'une révolution. Ainsi, le nom de chaque arête invariante est constituée de la façon suivante :

$\langle \text{numéro d'étape} ;$
nom de l'arête génératrice (resp. du sommet générateur) appartenant au chemin ;
nom du sommet générateur (resp. de l'arête génératrice) appartenant au profil \rangle

Par analogie avec la caractérisation adoptée par la méthode de Wu (voir section 2.3.1.2) :

$$(\text{ID}, \text{ID}_p, \text{ID}_{\text{élément}}, \text{ID}_{\text{chemin}}, \text{ID}_{\text{trajectoire}})$$

le numéro d'étape correspond à l'identifiant de l'opération d'extrusion ou de révolution ID , le nom de l'élément générateur appartenant à la trajectoire correspond à l'arête, caractérisée par $\text{ID}_{\text{trajectoire}}$, faisant partie du chemin caractérisé par l'identifiant $\text{ID}_{\text{chemin}}$, et enfin le nom de l'élément générateur appartenant au profil correspond au sommet caractérisé par l'identifiant $\text{ID}_{\text{élément}}$ faisant partie du profil caractérisé par l'identifiant $\text{ID}_{\text{chemin}}$. Sur la figure 4.5(b), l'arête $v'_1 e_1$ de l'objet extrudé correspond au produit cartésien du sommet générateur v'_1 issu du profil, avec l'arête génératrice e_1 issue de la

1. En CAO, cet objet est appelé outil.

4.2. NOMINATION PERSISTANTE D'ARÊTES

trajectoire. Il est à noter que l'extrusion de contours $2D$ a été généralisé en toute dimension, notamment pour des besoins en animation. Dans ce cadre, les objets sont générés par l'opération de produit cartésien [SL01]. Lorsque les objets utilisés comme paramètres de cette opération sont invariants alors les arêtes constituant leurs bords sont le résultat du produit cartésien entre un sommet et une arête. La caractérisation employée pour identifier ces arêtes est la même que celle décrite par le quadruplet 4.2.

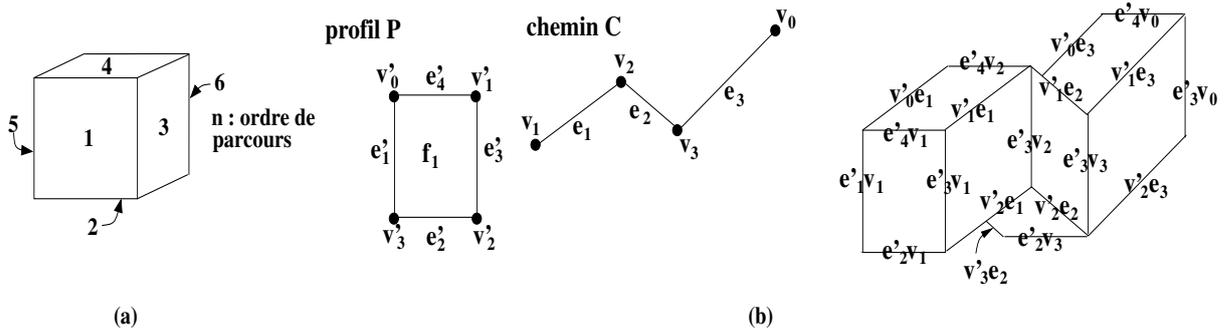


Figure 4.5: Caractérisation d'arêtes invariantes. (a) Les arêtes sont caractérisées par rapport à un parcours topologique. (b) Les arêtes sont caractérisées par rapport au produit cartésien des sommets et des arêtes du profil et des sommets et des arêtes de la trajectoire.

4.2.2.2 Caractérisation des arêtes contingentes

La caractérisation d'une arête contingente est basée sur des éléments qui ne varient pas (entités invariantes), des éléments agrégatifs qui permettent l'appréhension des différentes sémantiques de modélisation (agrégats invariants d'entités) et éventuellement des informations géométriques lorsque le voisinage topologique de l'arête ne suffit pas à l'identifier de façon unique et non ambigu.

En ce qui concerne son voisinage topologique, une arête est incidente à deux faces F_1 et F_2 ayant deux sommets S_1 et S_2 à son bord. Dans la caractérisation présentée en 4.1, S_1 et S_2 sont caractérisés par deux ensembles de faces Ens_1 et Ens_2 . Avec le premier point-clé que nous avons évoqué tout à l'heure, les ensembles Ens_1 et Ens_2 sont constitués de faces invariantes seulement. De plus, pour capturer les différentes sémantiques inhérentes aux opérations de modélisation, ces ensembles sont enrichis avec des entités et des agrégats invariants d'entités de toute dimension ($2^{\text{ème}}$ point-clé). Donc, le nom d'une arête contingente est donné par le quadruplet suivant :

$$\begin{aligned} &< \text{numéro d'étape} ; \\ &\text{ensemble des faces invariantes incidentes} ; \\ &\text{ensemble des entités invariantes et agrégats invariants d'entités incidents au premier sommet} ; \\ &\text{ensemble des entités invariantes et agrégats invariants d'entités incidents au second sommet} > \end{aligned} \quad (4.2)$$

Dans la figure 4.6, l'arête e_1 est caractérisée par le quadruplet

$$\langle i, \{f_1, f_2\}, \{s_1, e_2, e_3, f_3\}, \{s_2, e_4, e_5, f_4\} \rangle,$$

les arêtes e_6 et e_7 sont respectivement caractérisées par les quadruplets

$$\langle i + 2, \{f_1, f_2\}, \{s_1, e_2, e_3, f_3\}, \{f_7, \text{coque_latérale}_{\text{rainure}}, \text{volume}_2\} \rangle \text{ et}$$

$$\langle i + 2, \{f_1, f_2\}, \{f_8, \text{coque_latérale}_{\text{rainure}}, \text{volume}_2\}, \{s_2, e_4, e_5, f_4\} \rangle$$

alors que les arêtes e_8 et e_9 sont respectivement caractérisées par les quadruplets

$$\langle i + 2, \{f_1, f_2\}, \{s_1, e_2, e_3, f_3\}, \{f_9, \text{coque_latérale}_{\text{rainure}}, \text{volume}_2\} \rangle \text{ et}$$

$$\langle i + 2, \{f_1, f_2\}, \{f_{10}, \text{coque_latérale}_{\text{rainure}}, \text{volume}_2\}, \{s_2, e_4, e_5, f_4\} \rangle .$$

Nous notons au vu de ces caractéristiques que :

1. f_7 et f_8 sont respectivement les ancêtres invariants des faces f_{11} et f_{12} qui sont à leurs tours respectivement incidentes aux bords des arêtes e_6 et e_7 . Il en est de même pour les faces f_9 et f_{10} avec les faces f_{13} et f_{14} ;
2. les coques latérales des rainures dans les figures 4.6(c) et (d) sont composées de toutes les faces de la rainure exceptée celle colorée en gris ;
3. le volume résultant de l'extrusion dans la figure 4.6(b) est caractérisé par volume_2 ;
4. l'ancêtre invariant des coques précédemment citées est la coque latérale du volume de la figure 4.6(b) constituée de toute les faces du volume $f_7, f_8, f_9, f_{10}, \dots$ excepté les faces f (face de devant) et la face de derrière colorées en gris ;
5. le voisinage topologique du second (resp. du premier) sommet des arêtes e_6 et e_8 (resp. e_7 et e_9) ne comprend ni sommets ni arêtes invariantes ou ayant des ancêtres invariants. En effet, le sommet et les arêtes incidents aux bords de e_6, e_7, e_8 et e_9 sont contingents et sans source.

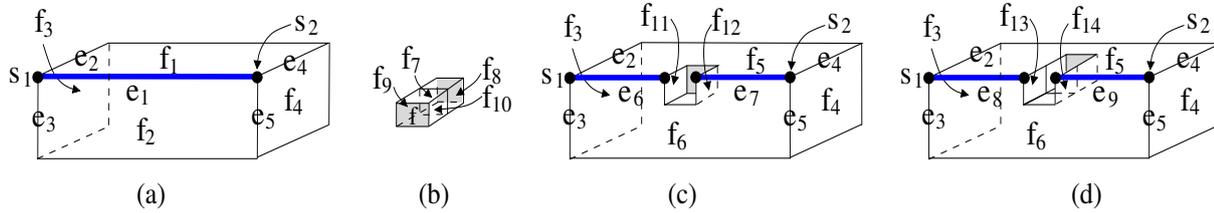


Figure 4.6: Caractérisation d'arêtes contingentes. (a) Construction d'un premier bloc par extrusion de la face f_2 . (b) Construction d'un second bloc par extrusion de la face f le long d'une multi-ligne. (c) Rainure effectuée par l'opération de différence booléenne entre le premier et le second bloc. (d) Même opération que précédemment mais en déplaçant le second bloc vers le fond.

La structure de nom décrite par le quadruplet 4.2 ne serait pas complète sans considérer des informations géométriques supplémentaires dans le cas général de la modélisation nD et celle traitant les objets avec des faces non planaires. Dès lors, le nom d'une arête contingente est décrit par :

$$\begin{aligned} &\langle \text{numéro d'étape} ; \\ &\text{ensemble des entités invariantes de dimension } n - 1 \text{ incidentes à l'arête}^2 ; \\ &\text{ensemble des entités et agrégats invariants d'entités incidents au premier sommet} ; \\ &\text{ensemble des entités et agrégats invariants d'entités invariants et incidentes au second sommet} \rangle \end{aligned} \quad (4.3)$$

En terme ensembliste, cette structure se traduit par :

$$\begin{aligned}
 &< \text{numéro d'étape ;} \\
 &\{EA_{n-1}\} ; \\
 &\{E_{0.1}, E_{1.1}, \dots, E_{n.1}\}; \\
 &\{E_{0.2}, E_{1.2}, \dots, E_{n.2}\} >
 \end{aligned} \tag{4.4}$$

(où EA_{n-1} est l'ensemble des entités invariantes de dimension $n - 1$ incidentes à l'arête, $\forall i = \{0, \dots, n\}$, $E_{i.1}$ est l'ensemble des entités et agrégats invariants d'entités de dimension i incidentes au premier sommet et $\forall i = \{0, \dots, n\}$, $E_{i.2}$ est l'ensemble des entités et agrégats invariants d'entités de dimension i incidentes au second sommet)

Dans ce qui suit, nous précisons la méthode utilisée pour obtenir les ancêtres invariants des entités et des agrégats d'entités de toute dimension. Nous précisons également les calculs effectués pour obtenir les informations géométriques que nous utilisons pour améliorer les résultats obtenus par les approches développées dans la bibliographie.

Calcul des ancêtres invariants

Comme nous venons de le décrire, la caractérisation d'une arête comprend des références à des entités invariantes de toute dimension. Ces entités sont en fait les ancêtres invariants des entités, incidentes à l'arête considérée ou à ses deux sommets extrémités (lorsque celles-ci sont contingentes). Pour retrouver ces ancêtres, il faut pouvoir d'une manière ou d'une autre, soit remonter dans l'historique de modélisation au travers d'une structure dédiée conservant l'historique des modifications survenues sur les entités, soit propager au cours de la modélisation sur chaque entité des attributs permettant de connaître son origine. Nous proposons d'utiliser des mécanismes de propagation d'attributs en se basant sur la structure de journal de bord étudiée et définie au chapitre 3. En effet, cette structure lie à travers les événements de création, scission, fusion, suppression et modification les entités topologiques à une étape du processus de modélisation. La propagation de ces ancêtres s'effectue comme suit :

- dans le cas d'un événement de création ou de suppression il n'y a pas lieu de propagation ;
- dans le cas de la scission d'une entité e_1 en plusieurs entités e'_1, e'_2, \dots, e'_n , les ancêtres de e_1 sont associés aux entités e'_1, e'_2, \dots, e'_n ;
- dans le cas de la fusion de plusieurs entités e_1, e_2, \dots, e_n en une seule e'_1 , les ancêtres de e_1, e_2, \dots, e_n sont associés à e'_1 ;
- dans le cas de la modification d'une entité e en une autre entité e' , l'ancêtre de e sont associées à ceux de e' .

Ainsi, si l'on reprend l'exemple de la figure 2.17 représentée dans la figure 4.7, nous pouvons constater :

1. qu'à l'étape 1', la face f_1 est une face invariante ;
2. qu'à l'étape 2', la face f_1 est modifiée en une face f_2 . Par conséquent, l'ancêtre invariant que constitue la face f_1 est associé à la face f_2 ;
3. qu'à l'étape 3', la face f_2 est scindée en 3 faces f_6, f_7 et f_8 . Par conséquent, l'ancêtre invariant qui est associé à la face f_2 (c'est-à-dire, f_1) est propagé aux faces f_6, f_7 et f_8 ;

2. Dans la suite de notre étude, et par souci de clarté, nous présentons le problème de la nomination persistante d'arête, essentiellement en 3D. Par conséquent, l'ensemble des entités invariantes de dimension $n - 1$ incidentes à l'arête devient l'ensemble des faces invariantes incidentes à l'arête considérée.

4. qu'à l'étape 4', les faces f_7 et f_8 ont subies des modifications et cela a engendré respectivement les faces f_9 et f_{10} . Par conséquent, l'ancêtre invariant f_1 est associé aux faces f_9 et f_{10} .

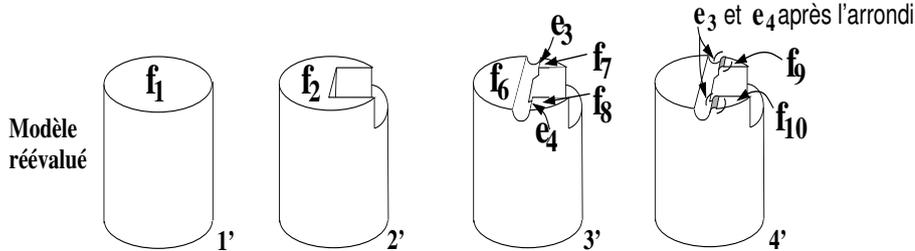


Figure 4.7: Calcul des ancêtres invariants.

Calcul des informations géométriques

Dans un contexte non-planaire (c'est-à-dire, lorsque les modèles géométriques 3D sont bordés par des faces non-planaires), toutes les solutions de la littérature utilisent des informations géométriques pour résoudre le problème de la nomination persistante. Bien qu'elles garantissent l'unicité des références aux entités topologiques dans le modèle paramétrique, elles négligent les propriétés géométriques des surfaces (plongement non-planaire des faces) et elles donnent des résultats différents de ceux escomptés. Notre objectif n'est pas de mettre en œuvre une autre approche pour résoudre le problème de la nomination persistante mais de fournir une solution qui améliore les résultats obtenus avec les solutions déjà existantes.

Lors de la manipulation du modèle géométrique, les caractéristiques de forme telles que la courbure sont plus faciles à percevoir que la topologie des surfaces car :

1. la forme est le principal critère qui permet à l'utilisateur de distinguer ce qu'il voit ;
2. la courbure d'une surface constitue un élément essentiel de la perception de formes.

Comme nous pouvons le voir sur la figure 4.8, la forme de la surface F indique la présence d'une bosse entre deux trous. Les solutions présentées dans la section 2.3.2.2 ont omis de prendre en compte cet élément important et nous avons montré que cela a conduit à des appariements incorrects entre arêtes. Cela nous a conduit à considérer les faces non-planaires comme une collection de sous-surfaces associées à des caractéristiques de forme que nous allons décrire [BaMS09].

Notre méthode décompose les faces non-planaires sur la base du critère de courbure. Elle permet à la suite de cette décomposition de constituer les informations géométriques nécessaires à la caractérisation des arêtes résultant de l'intersection de faces non-planaires. Elle est constituée de trois étapes :

1. Calcul des extremums locaux : si on considère la surface étudiée comme une fonction à deux variables $f(x, y)$, un extremum local est un point $p_e = (x_e, y_e)$ qui satisfait aux conditions suivantes³ :
 - (a) $f_x(x_e, y_e) = 0$ (où f_x est la dérivée partielle du premier ordre par rapport à x) ;
 - (b) $f_y(x_e, y_e) = 0$ (où f_y est la dérivée partielle du premier ordre par rapport à y) ;

3. Le calcul d'extremums locaux peut être généralisé pour des surfaces plongées en toute dimension.

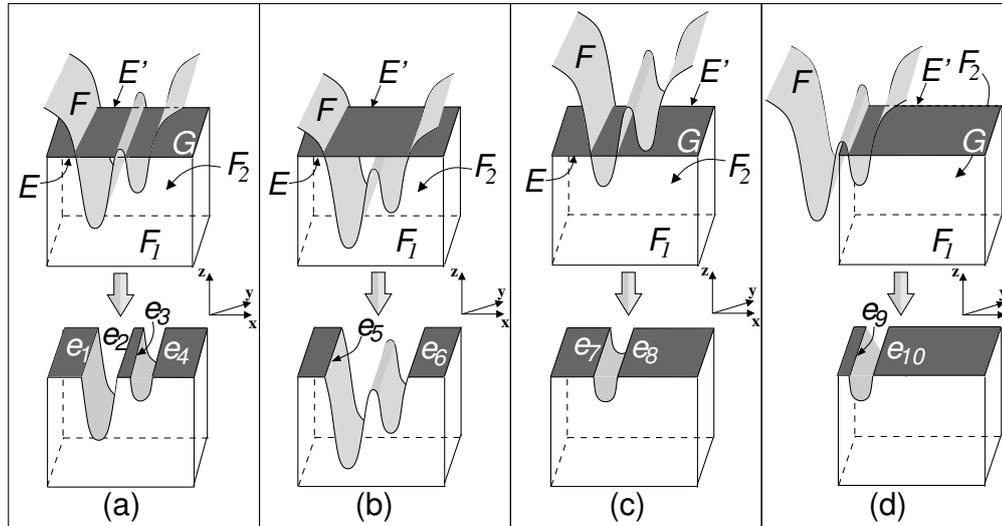


Figure 4.8: Différents modèles géométriques résultant de l'opération booléenne de différence lorsque la position de la surface F change.

(c) $f_{xx}(x_e, y_e) \times f_{yy}(x_e, y_e) - f_{xy}(x_e, y_e)^2 > 0$ (où f_{xx} , f_{yy} et f_{xy} sont respectivement les dérivées partielles du second ordre par rapport à x , y puis x et y .)

En qualité d'exemple de calcul d'extremums locaux, nous pouvons considérer la fonction suivante : $f(x, y) = x \times \exp(-x^2 - y^2)$ (voir la figure 4.9).

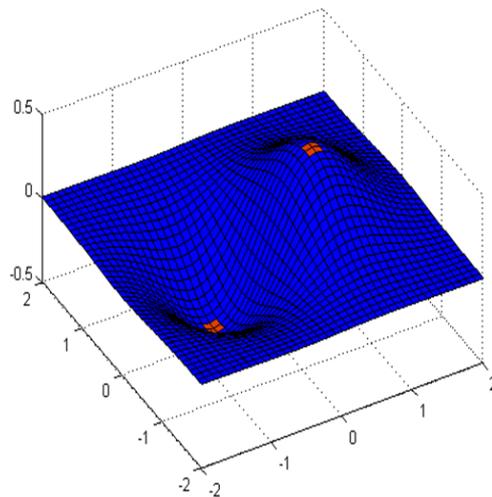


Figure 4.9: Calcul des extremums locaux. Les deux extremums locaux de la surface sont représentés en rouge.

2. Calcul des courbes d'inflexion : les courbes d'inflexion sont des collections de points $p_i = (x_i, y_i)$ qui satisfont aux conditions suivantes :

(a) $f_{xx}(x_i, y_i) = 0$;

(b) $f_{yy}(x_i, y_i) = 0$.

Elles délimitent des parties de surfaces (sous-surfaces) contenant chacune *au plus* un extremum local. Dans l'exemple précédent, les courbes d'inflexion de la surface $f(x, y) = x \times \exp(-x^2 - y^2)$ délimitent douze sous-surfaces notées R_1, R_2, \dots, R_{12} , comme l'illustre la figure 4.10. L'objectif escompté par cette étape est l'obtention d'une subdivision de sur-

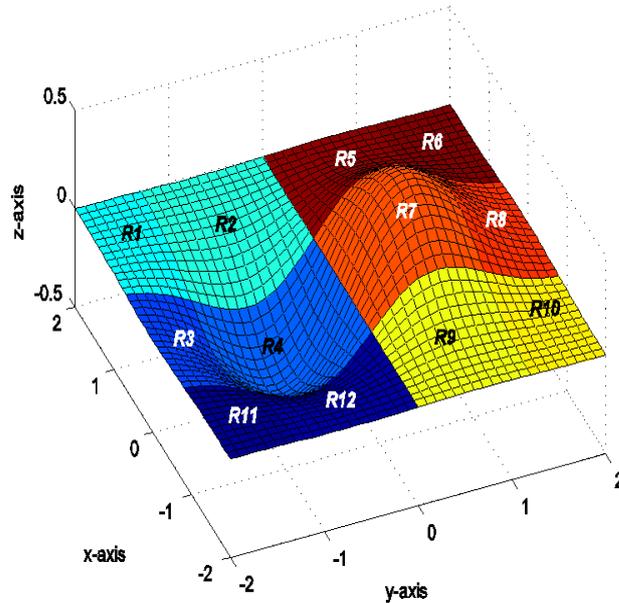


Figure 4.10: Calcul des courbes d'inflexion.

faces construisant autant de sous-surfaces qu'il y a d'extremums locaux. Cependant, cela est loin d'être le cas après le calcul des courbes d'inflexion. En effet, il est courant d'obtenir des sous-surfaces n'ayant aucun extremum local. Dans la figure 4.10, les sous-surfaces $R_1, R_2, R_3, R_5, R_6, R_8, R_9, R_{10}, R_{11}$ et R_{12} résultant du calcul des courbes d'inflexion ne contiennent pas d'extremum local.

3. Fusion des sous-surfaces : afin d'obtenir une décomposition de surface avec un extremum local par sous-surface, nous fusionnons les sous-surfaces bordées par les courbes d'inflexion. Pour ce faire, nous calculons la distance entre chaque point d'une sous-surface sans extremum local et un extremum local. Chaque point ainsi traité est assigné à la sous-surface qui comprend l'extremum local le plus proche. Pour l'exemple de la figure 4.11, les sous-surfaces R_1, R_2, R_3, R_{11} et R_{12} sont fusionnées avec la sous-surface R_4 et les sous-surfaces R_5, R_6, R_8, R_9 et R_{10} sont fusionnées avec la sous-surface R_7 , produisant respectivement les sous-surfaces Z_1 et Z_2 .

L'information géométrique qui découle de la décomposition de surface et qui est rajoutée à la structure de nom décrite par le quadruplet 4.3 est constituée :

1. des deux sous-surfaces adjacentes à l'arête considérée ;

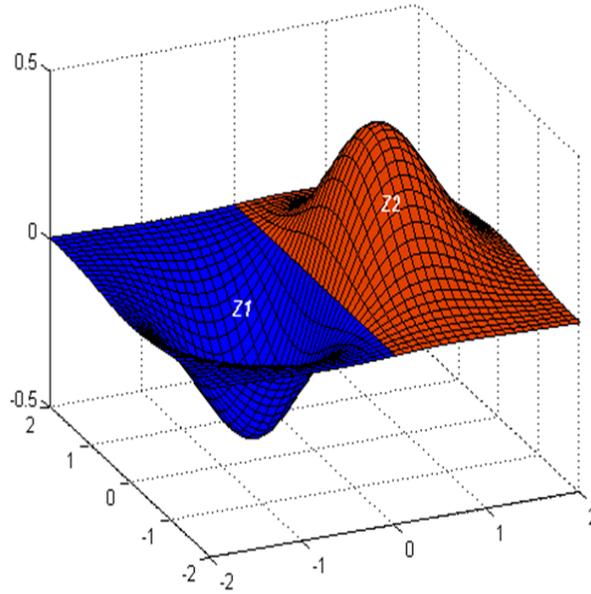


Figure 4.11: Résultat du processus de décomposition de surface.

2. de l'orientation de l'arête représentée par le gradient, calculée à l'une de ces deux extrémités.

Aussi, cette information géométrique se traduit par le triplet suivant :

$$\begin{aligned} &< 1^{\text{ère}} \text{ sous - surface incidente} ; \\ &2^{\text{ème}} \text{ sous - surface incidente} ; \\ &\text{orientation de l'arête} > \end{aligned} \quad (4.5)$$

Pour illustrer cela, reprenons l'exemple de la figure 4.8. Avec la seule donnée de leur voisinage, les arêtes e_1, e_2, \dots, e_8 sont caractérisées par le même nom

$$\langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\} \rangle .$$

Les différentes configurations représentées dans cette figure sont le résultat de l'intersection de la surface $F : z + \frac{1}{15}e^{-(x+5)^2+4} + e^{-0.09x^2+1} = 0$ et la face G plongée respectivement sur les plans $z = -2, z = 0.5, z = -3.5$ et $z = -2$. Le résultat de la décomposition de F est représenté par les différents modèles de la figure 4.12.

Par conséquent, le voisinage topologique des arêtes après cette décomposition change et les informations géométriques supplémentaires que nous utilisons pour lever l'ambiguïté précédente sont :

- $\langle S_1, G, [0 \ -3.1879 \ 0]^T \rangle$ pour e_1 ,
- $\langle S_2, G, [0 \ 2.2362 \ 0]^T \rangle$ pour e_2 ,
- $\langle S_2, G, [0 \ -0.6637 \ 0]^T \rangle$ pour e_3 ,
- $\langle S_3, G, [0 \ 0.6647 \ 0]^T \rangle$ pour e_4 ,

- $\langle S_1, G, [0 \ -1.3448 \ 0]^T \rangle$ pour e_5 ,
- $\langle S_3, G, [0 \ 0.3904 \ 0]^T \rangle$ pour e_6 ,
- $\langle S_1, G, [0 \ -2.3033 \ 0]^T \rangle$ pour e_7 ,
- $\langle S_1, G, [0 \ 2.1582 \ 0]^T \rangle$ pour e_8 ,
- $\langle S_3, G, [0 \ 0.6647 \ 0]^T \rangle$ pour e_9 et
- $\langle S_3, G, [0 \ -2.3033 \ 0]^T \rangle$ pour e_{10} .

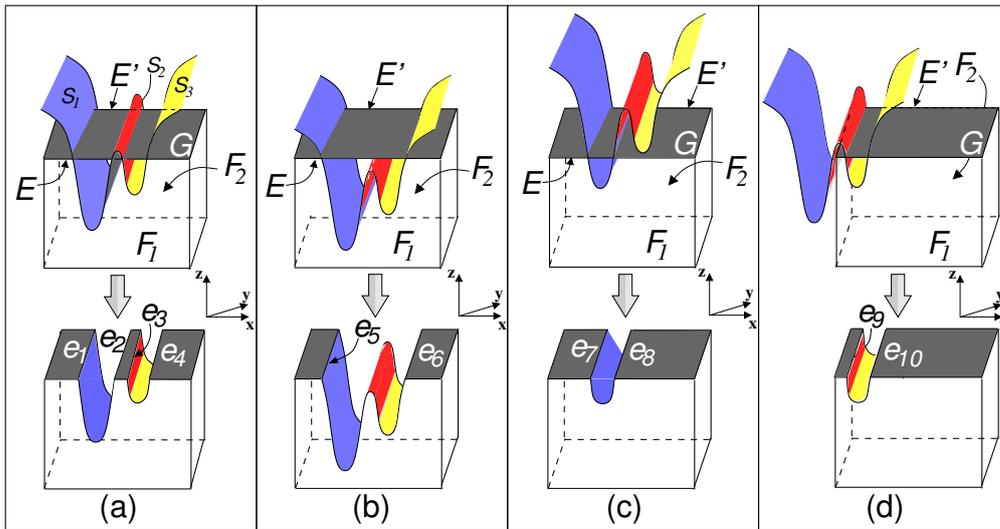


Figure 4.12: Les modèles de la figure 4.8 après la décomposition de la surface F .

4.2.3 Appariement des arêtes

L'appariement d'arêtes consiste à associer n arêtes du modèle initial à m arêtes du modèle réévalué. Le but est de déterminer, d'une part, pour chaque arête du modèle initial, une correspondance avec une ou plusieurs arêtes du modèle réévalué, et d'autre part, pour chaque arête du modèle réévalué, une correspondance avec une ou plusieurs arêtes du modèle initial. Conformément à la caractérisation des arêtes, nous distinguons l'appariement des arêtes invariantes, de celui des arêtes contingentes.

4.2.3.1 Appariement des arêtes invariantes

Dans cette configuration, les transformations topologiques des arêtes sont exclues. La caractérisation est triviale et l'appariement immédiat puisque la structure topologique des arêtes invariantes à appairer sont identiques. La caractérisation basée sur le parcours topologique, ainsi que celle s'appuyant sur les profils $2D$, suffisent car le nom généré identifie à l'identique toutes les arêtes invariantes quelle que soit la réévaluation.

4.2.3.2 Appariement des arêtes contingentes

Dans notre approche, l'appariement des arêtes contingentes est mis en œuvre par un procédé basé sur leur nom et fonctionnant par étape. D'abord, les arêtes à appairer sont celles incidentes à des faces ayant le même ancêtre invariant. Intuitivement, ces arêtes sont portées par deux droites virtuelles (une droite pour le modèle initial et une autre dans le modèle réévalué résultant chacune de l'intersection de deux faces invariantes) et ces *droites supports* peuvent être considérées comme des suites ordonnées d'arêtes à appairer. C'est le cas, par exemple, des arêtes e_2 et e_3 de la figure 4.13 qui doivent être appariées avec les arêtes e_4, e_5 et e_6 . De même, les deux faces invariantes qui s'intersectent sont représentées en gris et les droites supports sont représentées par des traits en pointillés.

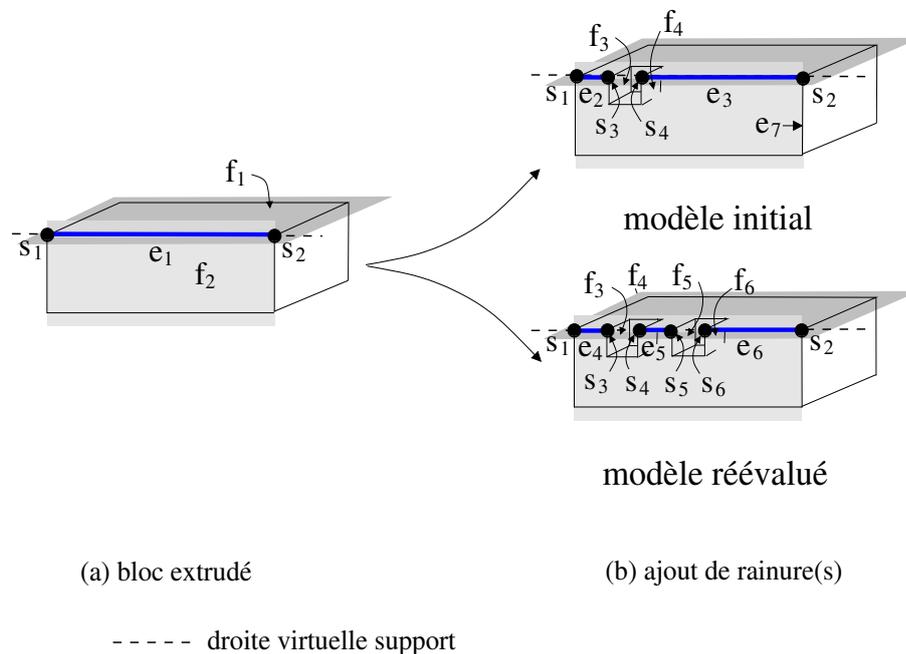


Figure 4.13: Détermination des arêtes à appairer.

Par ailleurs, à défaut de pouvoir appairer les arêtes ayant exactement le même nom (comme nous l'avons illustré au début de la section 4.2), nous pouvons établir des liens permettant aux arêtes du modèle initial de *recouvrir* les arêtes du modèle réévalué et inversement. En d'autres termes, nous pouvons établir des liens entre des suites d'une ou plusieurs arêtes du modèle initial bordées par des couples de sommets délimitant également une ou plusieurs suites d'arêtes du modèle réévalué et inversement. C'est le cas, par exemple de l'arête e_3 du modèle initial et des arêtes e_5 et e_6 du modèle réévalué de la figure 4.13(b), puisque celles-ci sont délimitées par le même couple de sommets s_4 et s_2 .

Les liens de *recouvrement* peuvent être différents selon qu'on les détermine à partir des arêtes du modèle initial ou des arêtes du modèle réévalué. C'est le cas, par exemple, de l'arête e_1 de la figure 4.14(a) qui recouvre les arêtes e_5, e_6 et e_7 de la figure 4.14(b) puisque ces deux suites d'arêtes sont délimitées par les sommets s_1 et s_3 (voir la partie grisée de la figure 4.14(c)). En revanche, les arêtes e_5, e_6 et e_7 recouvrent individuellement et réciproquement les arêtes e_1 et

e_2 dans le premier cas (puisque e_5 , d'une part, et e_1, e_2 , d'autre part, sont délimitées par les sommets s_1 et s_5), l'arête e_3 dans le deuxième cas (puisque les arêtes e_6 et e_3 sont délimitées par les sommets s_6 et s_7) et les arêtes e_2 et e_3 pour le troisième cas (puisque l'arête e_7 , d'une part, et les arêtes e_2, e_3 , d'autre part, sont délimitées par les sommets s_8 et s_3 (voir la partie délimitée par un trait rouge sur la figure 4.14(c)).

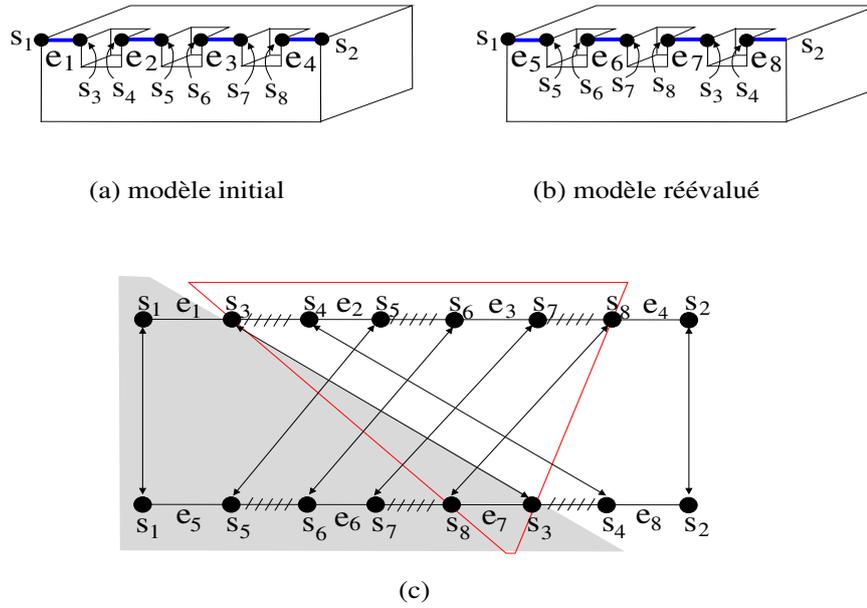


Figure 4.14: Liens de recouvrement entre arêtes. (a) modèle géométrique initial conçu par l'ajout de 3 rainures. (b) Réévaluation du modèle initial en modifiant la position de la rainure de gauche vers la droite. (c) Deux droites supports portant les arêtes à appairer avec des liens entre les extrémités des arêtes ayant le même voisinage topologique.

Donc, des liens de recouvrement depuis les arêtes du modèle initial vers celles du modèle réévalué, et réciproquement, sont calculés. Au final, un recouvrement des liens de recouvrement établis dans les deux sens, permet le *calcul effectif des liens d'appariement* reliant les arêtes du modèle initial à celles du modèle réévalué.

Notre mécanisme d'appariement des arêtes contingentes se décompose en trois principales étapes :

1. la *détermination des arêtes à appairer* : cette étape permet de déterminer, parmi toutes les arêtes des modèles initial et réévalué, celles devant être regroupées dans le but d'être appariées. Ces arêtes sont caractérisées dans chaque modèle par des noms ayant le même ensemble de faces invariantes qui leurs sont incidentes (voir le deuxième élément du quadruplet 4.3). L'intersection de ces faces invariantes incidentes forme la *Droite Support* du modèle *Initial* « *DSI* » et la *Droite Support* du modèle *Réévalué* « *DSR* ». Dans l'exemple de la figure 4.13(b), les arêtes qui doivent être appariées avec e_2 et e_3 sont e_4, e_5 et e_6 parce qu'elles sont incidentes à des faces ayant pour ancêtres invariants les faces f_1 et f_2 . En revanche, l'arête e_7 n'est pas concernée dans cet appariement car elle est adjacente à un autre couple de faces ;
2. le *calcul des recouvrements entre arêtes* : ce calcul permet d'évaluer dans quelle mesure

4.2. NOMINATION PERSISTANTE D'ARÊTES

les arêtes du modèle initial sont incluses dans celles du modèle réévalué (premier sens d'inclusion) et réciproquement (second sens d'inclusion). Pour cela, on définit deux types de relations : les relations de *Recouvrement* des arêtes du modèle *Initial* sur les arêtes du modèle *Réévalué* « *RIR* », et celles des arêtes du modèle *Réévalué* sur les arêtes du modèle *Initial* « *RRI* ».

Avant d'entamer le calcul de ces deux types de relations, nous devons faire correspondre les extrémités des arêtes portées sur la *DSI* avec les extrémités des arêtes portées sur la *DSR*. Comme nous pouvons le constater sur la structure de nom décrite par le quadruplet 4.4, une extrémité d'arête est caractérisée par l'ensemble des entités et des agrégats invariants d'entités incidents à la première ou à la seconde extrémité de l'arête $\{E_{0,j}, E_{1,j}, \dots, E_{n,j}\}$ (j est égale à 1 ou à 2 et indique s'il s'agit de la première ou de la deuxième extrémité de l'arête caractérisée). Dès lors, nous considérons que deux extrémités d'arêtes, e et e' , sont correspondantes si et seulement si $\exists i \in \{0, 1, \dots, n\}$ tel que $E_{e,i} = E_{e',i}$. Dans la figure 4.15, les extrémités s_3, s_4, s_5 et s_6 des arêtes e_6, e_7, e_8 et e_9 sont respectivement caractérisées par les voisinages topologiques suivants :

$$\begin{aligned} & \{\{\}, \{\}, \{f_9, coque_latérale_rainure\}, \{volume_2\}\}, \\ & \{\{\}, \{\}, \{f_{10}, coque_latérale_rainure\}, \{volume_2\}\}, \\ & \{\{\}, \{\}, \{f_7, coque_latérale_rainure\}, \{volume_2\}\} \text{ et} \\ & \{\{\}, \{\}, \{f_8, coque_latérale_rainure\}, \{volume_2\}\}. \end{aligned}$$

Par conséquent, s_3 et s_4 correspondent chacun à s_5 et s_6 puisque les voisinages qui les caractérisent ont en commun la coque latérale de la rainure et le volume de la figure 4.15(b) caractérisé par l'identifiant $volume_2$.

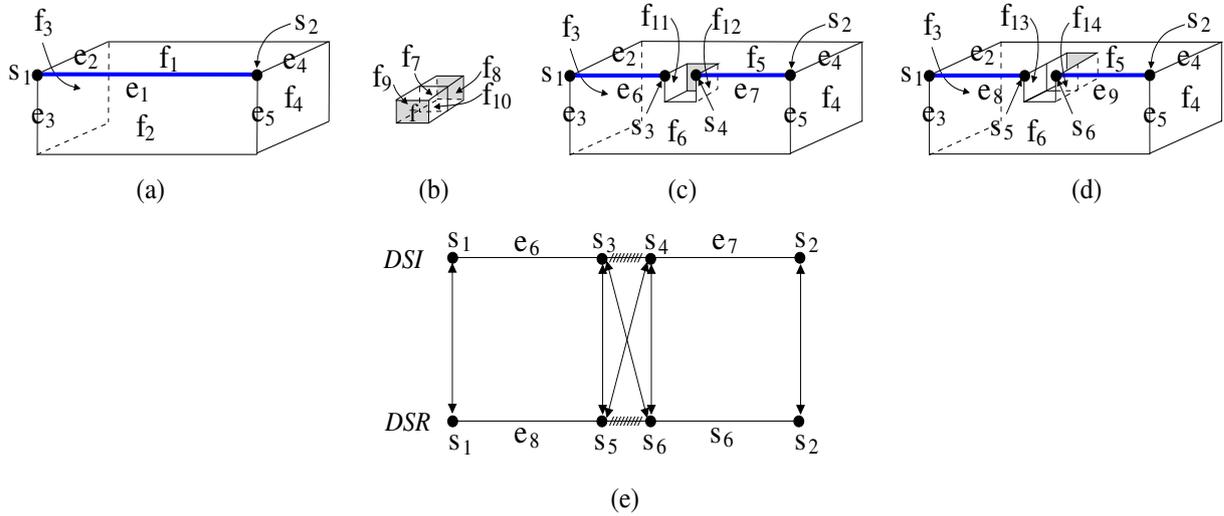


Figure 4.15: Correspondances entre des extrémités d'arêtes. Les modèles géométriques sont extraits de la figure 4.6.

Après la mise en correspondance de ces extrémités, le calcul des relations de recouvrement des arêtes du modèle initial sur celles du modèle réévalué est effectué selon l'algorithme suivant :

- (a) parcourir la suite de sommets portés sur la droite support du modèle initial DSI. Dans la figure 4.15(e), ce parcours passe par les sommets s_1, s_3, s_4 et s_2 dans cet ordre ;
- (b) extraire une ou plusieurs arêtes délimitées par deux extrémités appariées. Soient seq_1 cette suite d'arêtes et ext_1, ext_2 ces extrémités. Donc, comme illustré dans la figure 4.15, la première suite d'arêtes considérée correspond à $seq_1 = e_6$;
- (c) lorsque les extrémités ext_1 et ext_2 renvoient vers deux autres extrémités portées sur la DSR, dans un ordre inverse (c'est-à-dire, ext_2 apparaît avant ext_1 sur la DSR), alors la suite seq_1 est ignorée et on passe à l'étape (a). C'est le cas, par exemple dans figure 4.14(c), de la suite d'arêtes e_2 et e_3 , délimitée par s_3 et s_8 sur la DSI qui se présentent dans un ordre inverse sur la DSR ;
- (d) dans le cas où ces mêmes extrémités se présentent dans le même ordre, une deuxième suite d'arêtes, délimitée par ext_1 et ext_2 sur la DSR, doit être extraite. Soit seq_2 cette suite ;
- (e) rajouter la relation de recouvrement $seq_1 \rightarrow seq_2$ à l'ensemble des RIR. Ainsi, dans le cas de la figure 4.14(c), l'arête e_1 renvoie vers les arêtes e_5, e_6 et e_7 et cet état de fait est représenté par la relation de recouvrement suivante : $e_1 \rightarrow e_5, e_6, e_7$.
- (f) si le parcours de la DSI n'a pas encore atteint le dernier sommet alors le processus est réitéré à l'étape (a).

Les RRI sont calculées, de la même façon, en parcourant la droite support du modèle réévalué. Dans la configuration de la figure 4.14, les deux types de relations de recouvrements sont données comme suit :

$$RIR : \begin{cases} e_1 \longrightarrow & e_5, e_6, e_7 \\ e_3 \longrightarrow & e_6 \\ e_4 \longrightarrow & e_7, e_8 \end{cases} \quad (4.6)$$

$$RRI : \begin{cases} e_5 \longrightarrow & e_1, e_2 \\ e_6 \longrightarrow & e_3 \\ e_8 \longrightarrow & e_2, e_3, e_4 \end{cases} \quad (4.7)$$

3. le *calcul effectif de l'appariement* : il va permettre de définir des liens de correspondances entre les arêtes du modèle initial et celles du modèle réévalué en fonction des recouvrements précédemment calculés. Donc, le calcul des relations d'appariements est réalisé en exploitant les RIR et RRI. Avant de commencer à traiter ces deux ensembles de relations, une simplification s'impose. En effet, les arêtes du modèle initial qui ne recouvrent pas celles du modèle réévalué ou celles du modèle réévalué qui ne recouvrent pas les arêtes du modèle initial sont supprimées dans les règles RIR et RRI. En effet, à partir du moment où ces arêtes n'en recouvrent pas d'autres, on ne peut pas considérer qu'elles puissent être appariées. Dans l'exemple précédent, l'arête e_2 (resp. l'arête e_7) ne recouvre aucune arête du modèle réévalué (resp. du modèle initial). Donc, leurs occurrences sont supprimées dans les relations 4.6 et 4.7 pour mener au résultat suivant :

$$\begin{cases} e_1 \longrightarrow & e_5, e_6 \\ e_3 \longrightarrow & e_6 \\ e_4 \longrightarrow & e_8 \end{cases} \quad (4.8)$$

$$\begin{cases} e_5 \longrightarrow & e_1 \\ e_6 \longrightarrow & e_3 \\ e_8 \longrightarrow & e_3, e_4 \end{cases} \quad (4.9)$$

À partir de là, la construction des relations d'appariement repose sur un principe assez simple : pour appairer un ensemble de i arêtes A_1, A_2, \dots, A_i du modèle initial avec un autre ensemble de j arêtes B_1, B_2, \dots, B_j du modèle réévalué, il est nécessaire de trouver une relation de recouvrement $R = A_1, A_2, \dots, A_i \rightarrow B_1, B_2, \dots, B_j$ appartenant à l'ensemble RIR, et une autre relation de recouvrement $R' = B_1, B_2, \dots, B_j \rightarrow A_1, A_2, \dots, A_i$ appartenant à l'ensemble RRI qui relie le premier ensemble d'arêtes au second dans les deux sens. Après avoir trouvé R et R' , nous pouvons construire la relation d'appariement suivante : $A_1, A_2, \dots, A_i \leftrightarrow B_1, B_2, \dots, B_j$. Par la suite, les relations R et R' ne sont plus considérées. De même, les occurrences des arêtes A_1, A_2, \dots, A_i et celles des arêtes B_1, B_2, \dots, B_j sont supprimées dans les relations de recouvrements restantes. Ce processus est réitéré jusqu'à ce qu'il ne reste aucune relation dans RIR et RRI.

Le calcul des relations d'appariement en se basant sur les relations RIR 4.8 et RRI 4.9 s'effectue comme suit :

- (a) extraction de l'appariement $e_3 \leftrightarrow e_6$: les relations de recouvrement RIR et RRI deviennent

$$\begin{cases} e_1 \longrightarrow & e_5, e_6 \\ e_4 \longrightarrow & e_8 \end{cases}$$

$$\begin{cases} e_5 \longrightarrow & e_1 \\ e_8 \longrightarrow & e_3, e_4 \end{cases}$$

puis la suppression des occurrences des arêtes e_3 et e_6 engendre les deux systèmes de relations suivants :

$$\begin{cases} e_1 \longrightarrow & e_5 \\ e_4 \longrightarrow & e_8 \end{cases}$$

$$\begin{cases} e_5 \longrightarrow & e_1 \\ e_8 \longrightarrow & e_4 \end{cases}$$

- (b) extraction de l'appariement $e_1 \leftrightarrow e_5$: cela engendre les systèmes suivants :

$$\{ e_4 \longrightarrow e_8$$

$$\{ e_8 \longrightarrow e_4$$

- (c) extraction de l'appariement $e_4 \leftrightarrow e_8$.

Finalement, le résultat du processus d'appariement entre les arêtes contingentes e_1, e_2, \dots, e_8 donne le résultat suivant :

$$\begin{cases} e_3 \longleftrightarrow e_6 \\ e_1 \longleftrightarrow e_5 \\ e_4 \longleftrightarrow e_8 \end{cases} \quad (4.10)$$

Le mécanisme d'appariement que nous venons de définir est efficace lorsque les arêtes portent des caractérisations différentes. Si plusieurs arêtes portent des caractérisations identiques (c'est-à-dire, qu'elles sont plusieurs à être caractérisé par le même quadruplet 4.3), il convient pour les distinguer, d'ajouter les informations géométriques définies par le triplet 4.5. Remarquons que les arêtes qui sont incidentes au même couple de sous-surfaces sont appariées. Si plusieurs arêtes des modèle initial et réévalué sont incidentes au même couple de sous-surfaces alors les arêtes appariées sont celles qui ont la même orientation.

Ainsi, si on applique ce principe sur les arêtes des modèles de la figure 4.12 caractérisées par :

- $e_1 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_1, G, [0 \ -3.1879 \ 0]^T \rangle \rangle \rangle$,
- $e_2 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_2, G, [0 \ 2.2362 \ 0]^T \rangle \rangle \rangle$,
- $e_3 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_2, G, [0 \ -0.6637 \ 0]^T \rangle \rangle \rangle$,
- $e_4 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_3, G, [0 \ 0.6647 \ 0]^T \rangle \rangle \rangle$,
- $e_5 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_1, G, [0 \ -1.3448 \ 0]^T \rangle \rangle \rangle$,
- $e_6 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_3, G, [0 \ 0.3904 \ 0]^T \rangle \rangle \rangle$,
- $e_7 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_1, G, [0 \ -2.3033 \ 0]^T \rangle \rangle \rangle$,
- $e_8 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_1, G, [0 \ 2.1582 \ 0]^T \rangle \rangle \rangle$,
- $e_9 : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_3, G, [0 \ 0.6647 \ 0]^T \rangle \rangle \rangle$,
- $e_{10} : \langle i, \{F, G\}, \{\{\}, \{E\}, \{F_1\}, \{\}\}, \{\{\}, \{E'\}, \{F_2\}, \{\}\}, \langle S_3, G, [0 \ -2.3033 \ 0]^T \rangle \rangle \rangle$.

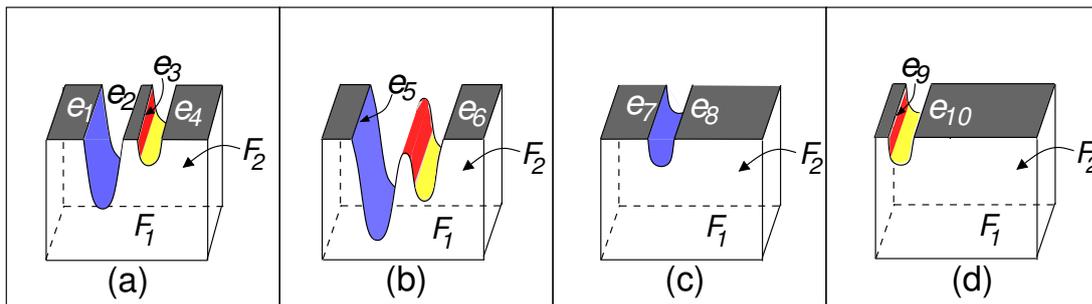


Figure 4.16: Les modèles résultant de l'interaction des faces F et G de la figure 4.8.

nous obtenons respectivement l'appariement des arêtes e_1 et e_4 sur les arêtes e_5 et e_6 lorsque le modèle de la figure 4.12(a) est réévalué en (b) puisque e_1 et e_5 (resp. e_4 et e_6) sont incidentes au même couple de sous-surfaces S_1 et G (resp. S_3 et G). L'appariement des arêtes dans les autres cas de réévaluations est représenté dans la table 4.1.

Cas	Appariement entre	Appariements attendus
1	le modèle a et le modèle b	$e_1 \leftrightarrow e_5$ $e_4 \leftrightarrow e_6$
2	le modèle a et le modèle c	$e_1 \leftrightarrow e_7$
3	le modèle a et le modèle d	$e_3 \leftrightarrow e_9$ $e_4 \leftrightarrow e_{10}$
4	le modèle b et le modèle c	$e_5 \leftrightarrow e_7$
5	le modèle b et le modèle d	$e_6 \leftrightarrow e_{10}$
6	le modèle c et le modèle d	aucun appariement

Table 4.1: Appariements des arêtes de la figure 4.16. arête₁ ↔ arête₂ signifie que arête₁ est appariée à arête₂.

4.3 Système de nomination hiérarchique

La majorité des mécanismes de nomination persistante traitent seulement les sommets, les arêtes et les faces (entités atomiques) (voir la section 2.3 des solutions proposées dans le chapitre 2). Peu d'entre eux considère les agrégats d'entités, en l'occurrence les agrégats de faces (coques) [Mar06] et les agrégats d'arêtes. De plus, ces contributions abordent la nomination persistante en dimension 3 seulement et aucune ne permet l'utilisation des entités et des agrégats d'entités de dimension supérieure. Tout cela s'avère être limitatif et ne permet de prendre en compte qu'une partie des intentions du concepteur. Notre solution consiste à proposer un système de nomination hiérarchique qui prend appui sur le système de nomination des arêtes pour mettre en place une nomination homogène des entités de plus haute dimension.

4.3.1 Notion d'agrégat

Un aspect important de la modélisation paramétrique est la possibilité d'exprimer différentes sémantiques en utilisant les agrégats d'entités de toute dimension (voir la section 2.2.3). Jusqu'à présent, les objets modélisés sont structurés en des agrégats de faces (coques). Les coques sont définies, par récursivité, en termes de sous-coques et de faces. En revanche, nous utilisons les arêtes comme unité de base.

Dans l'exemple de la figure 4.17, la *rainure débouchante* peut être décrite en termes d'agrégat de coque initiale (c_i), coque latérale (c_{lat}) et coque finale (c_f). De même, la coque latérale est structurée en deux autres coques (c_{lat_1} et c_{lat_2}). c_{lat_2} est ensuite structurée respectivement en face droite, gauche et de fond (c_d , c_g et c_{fd}). Finalement, chaque face de cette structure hiérarchique est à son tour structurée en un agrégat d'arêtes. Par exemple, la face c_{fd} est structurée en agrégat d'arêtes e_2 , e_5 , e_{11} et e_{12} .

De façon formelle, la structure hiérarchique que nous venons de décrire peut être représentée par un graphe direct acyclique (dag). Il est défini récursivement comme suit : un nœud représente un agrégat d'entités et un lien orienté du graphe (lien hiérarchique) dirigé d'un nœud a vers

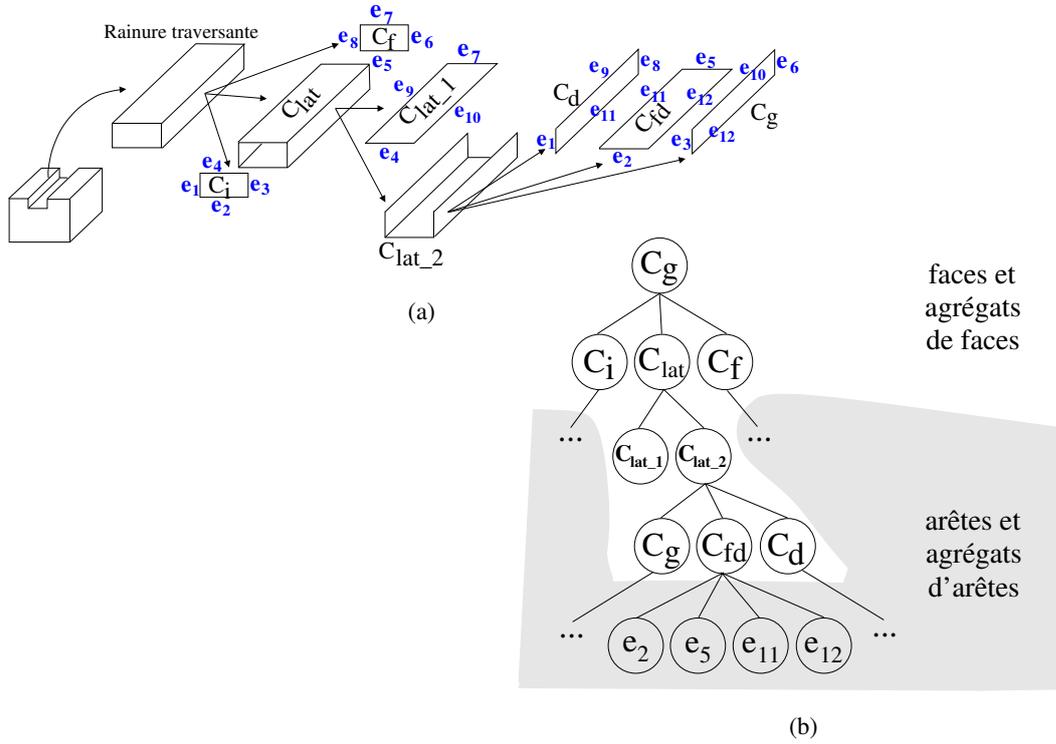


Figure 4.17: Notion d'agrégat d'entités. (a) Les modèles sont structurés en agrégats de coques, les coques sont structurées, par récursivité, en agrégats de sous-coques et de faces et les faces sont structurées en agrégats d'arêtes. (b) Représentation, sous forme de graphes, de la structure hiérarchique des agrégats d'entités correspondants.

b signifie que l'agrégat d'entités b est inclu dans l'agrégat d'entités a . Au niveau le plus bas de la structure hiérarchique, un nœud représente un agrégat d'entités particulier puisqu'il est constitué que d'un seul élément, l'entité atomique « arête ». Ce nœud est associé à un label égal à 0 correspondant au niveau hiérarchique le plus bas du graphe et il est noté A_0 . Ensuite, un agrégat d'entités de niveau supérieur est défini récursivement par l'ensemble :

$$A^p = \{A_1^{q_1}, A_2^{q_2}, \dots, A_n^{q_n}\}_{n \geq 1, q_i \leq p, p=1+\max(q_i)_{i=1 \dots n}}$$

où p est le niveau hiérarchique de l'agrégat A_p dans le graphe. A^p est représenté dans le graphe par un nœud connecté à travers des liens hiérarchiques orientés vers les nœuds $A_1^{q_1}, A_2^{q_2}, \dots, A_n^{q_n}$. Cela signifie que chaque agrégat $A_i^{q_i} | i \in \{1, \dots, n\}$ est inclus dans l'agrégat d'entité de niveau hiérarchique supérieur A^p . Dès lors, nous pouvons appeler A_p « super agrégat » et chaque $A_i^{q_i}$ est appelé « sous agrégat ». Dans l'exemple de la figure 4.17(b), les arêtes e_2, e_5, e_{11} et e_{12} sont les sous agrégats élémentaires (niveau 0 du graphe) du super agrégat c_{fd} qui est une face (niveau 1 du graphe). Par ailleurs, la rainure débouchante c_g (niveau 5 du graphe) est le super agrégat des sous agrégats c_i, c_{lat} et c_f .

Donc, le graphe est structuré en agrégats et sous agrégats d'entités de toute dimension. Il permet d'appréhender la topologie du modèle hiérarchique à différents niveaux de granularité. Il renforce par là-même les processus de nomination puisque ces derniers se basent avant tout

sur les entités invariantes pour caractériser et apparier tout ce qui est référencé au niveau du modèle paramétrique. Nous décrivons ci-dessous comment la notion d'agrégat est associée aux entités invariantes du modèle.

4.3.2 Caractérisation des agrégats

4.3.2.1 Caractérisation des agrégats invariants

Les agrégats invariants d'entités sont définis par agrégation de sous-agrégats invariants d'entités et/ou d'arêtes invariantes. Ils sont prédéfinis de manière unique et invariante par chaque opération de modélisation et portent en eux une sémantique. Leur construction peut se décliner sous deux formats différents :

- *agrégats invariants d'entités construits automatiquement* par le système paramétrique : les objets résultant de certaines opérations de modélisation (essentiellement, les opérations de création telles que l'extrusion ou la révolution) génèrent une topologie exprimant une sémantique qui va au-delà de la composition en termes de sommets, arêtes, faces ou volumes. En effet, comme le montre l'exemple de la figure 4.17, la structure des objets résultant de telles opérations est généralement connue d'avance à travers les caractéristiques de forme (par exemple, la coque initiale c_i , la coque latérale c_{lat} et la coque finale c_f de la rainure traversante). Ainsi, le système paramétrique prend en compte cet état de fait et construit le graphe hiérarchique en conséquence.
- *agrégats invariants d'entités construits par l'utilisateur* : parfois, le système n'est pas en mesure de générer automatiquement de tel agrégat ayant un intérêt particulier pour l'utilisateur. Il est alors nécessaire de permettre la manipulation et la nomination persistante d'agrégats invariants d'entités construits « à la main » parce qu'ils sont requis dans de nombreuses applications, notamment en CAO. Pour illustrer ce type d'agrégats, reprenons le modèle de la figure 2.34 à l'étape 2 du processus de modélisation initiale. Cette étape traduit l'ajout d'une protrusion sur un bloc. Supposons qu'au moment de la réévaluation, le nombre de protrusion augmente et atteint le nombre 4. À ce moment, une option intéressante au processus de modélisation serait de permettre la désignation de toutes les protrusions (illustrées en gris sur la figure 4.18) afin d'appliquer l'opération d'arrondi sur les arêtes situées entre ces protrusions et le bloc. Dans la suite de ce chapitre, seul le second format est considéré.

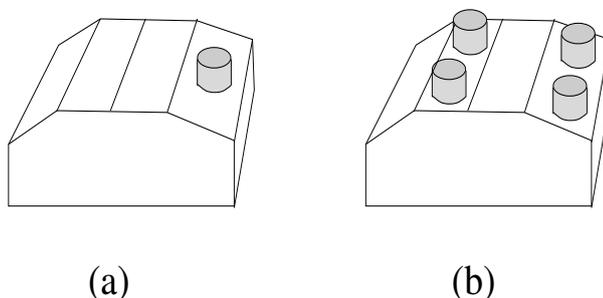


Figure 4.18: Agrégats invariants de faces construits par l'utilisateur et illustrés en gris sur la figure.

Ainsi, chaque agrégat invariants d'entités est associé à un identifiant unique qui labélise la sémantique qu'il porte et que nous appelons ID_A (*IDentifiant d'un Agrégat d'entités*). Dans le modèle paramétrique, la caractérisation d'un agrégat invariant est composé d'un numéro d'étape suivi du ID_A et s'écrit :

$$\langle \text{numéro d'étape}, ID_A \rangle$$

Dans ce cadre, les coques initiale (c_i), latérale (c_{lat}) et finale (c_f) de la figure 4.17 sont caractérisées respectivement par les noms $\langle 2, c_i \rangle$, $\langle 2, c_{lat} \rangle$ et $\langle 2, c_f \rangle$ si l'on considère la rainure débouchante comme la deuxième opération qui intervient lors du processus de modélisation.

4.3.2.2 Caractérisation des agrégats contingents

Les agrégats contingents d'entités résultent de l'évolution (scission, fusion, suppression et modification) d'un agrégat invariant d'entités. Ils apparaissent suite aux interactions qui peuvent survenir entre différents objets durant le processus de modélisation paramétrique. Aussi, des agrégats contingents d'entités possèdent nécessairement un ancêtre invariant (ou plusieurs dans le cas d'une fusion) comme c'est le cas pour les entités topologiques atomiques (sommets, arêtes, faces, etc.). Lorsqu'un agrégat contingent d'entités doit être référencé par une opération de modélisation, un nouveau nœud correspondant à cet agrégat est généré dans le graphe. Ce nœud est alors associé à la caractérisation suivante :

$$\langle \text{numéro d'étape}, \text{ancêtre invariant}, \text{liste d'arêtes} \rangle$$

où *numéro d'étape* indique l'étape de construction de cet agrégat, *ancêtre invariant* caractérise l'ancêtre invariant à partir duquel résulte l'agrégat contingent d'entités considéré et *liste d'arêtes* est la liste d'arêtes contenues dans cet agrégat contingent d'entités.

Dans l'ensemble des agrégats d'entités qui peuvent être traitées par notre système hiérarchique, il y a les entités topologiques de dimension supérieure ou égale à 2 (faces, volumes, hypervolumes, etc.) et tous les autres types d'agrégats. La sélection entités de dimension supérieure ou égale à 2 est une opération traitée par la plupart des noyaux géométriques à base topologique (c'est-à-dire, les noyaux à base de modèles topologiques comme c'est le cas pour le noyau de Moka qui utilise le modèle des cartes généralisées). En revanche, la plupart des noyaux n'utilisent pas de modèle topologique hiérarchique. Ils ne permettent donc pas de sélectionner et d'interagir avec les autres types d'agrégats. Dès lors, ces agrégats contingents d'entités doivent être calculés dynamiquement à travers un parcours topologique du modèle géométrique, au moment où l'utilisateur souhaite les référencer par une opération de modélisation. L'utilisateur sélectionne une entité topologique E appartenant à un agrégat. Ensuite, le système paramétrique doit être capable de calculer dynamiquement la liste L_{cont} des agrégats contingents d'entités contenant cette entité. L'utilisateur peut ensuite choisir dans cette liste l'agrégat qu'il souhaite référencer. Le calcul de L_{cont} est effectué en deux étapes :

1. le calcul de la liste L_{cont} passe par le calcul de la liste L_{inv} de tous les agrégats invariants d'entités contenant E . Il est effectué en remontant le graphe d'agrégats invariants à partir de l'ancêtre invariant de l'entité E et rajoutant à L_{inv} tous les agrégats invariant rencontrés. Pour étayer ces propos, reprenons le modèle initial de la figure 2.34 que nous représentons dans la figure 4.19.

4.3. SYSTÈME DE NOMINATION HIÉRARCHIQUE

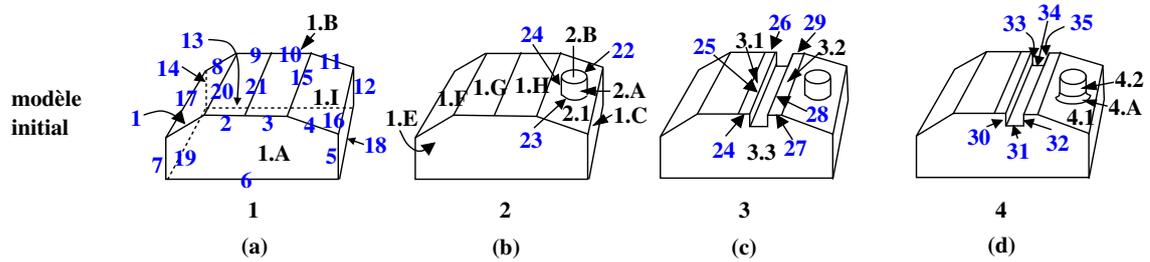


Figure 4.19: Sélection des agrégats contingents d'entités.

Les graphes des agrégats invariants d'entités qui découlent des étapes 1, 2, 3 et 4 sont respectivement représentés par les figures 4.20, 4.21, 4.21 et 4.21 (le graphe hiérarchique représentant la structure invariante n'évolue pas aux étapes 3 et 4). Ainsi, le graphe de la figure 4.20 traduit la structuration du bloc de la figure 4.19(a) en une coque de devant (1.devant qui correspond à la face 1.A), une coque de derrière (1.arrière qui correspond à la face 1.B) et une coque latérale (1.latérale) composée elle-même des coques de droite (1.droite), du bas (1.bas), de gauche (1.gauche) et du haut (1.haut). Ensuite, les coques de droite, bas et gauche sont respectivement correspondantes aux faces 1.C, 1.D et 1.E. De plus, la coque du haut (1.haut) est constituée des faces 1.G, 1.H, 1.I et 1.F.

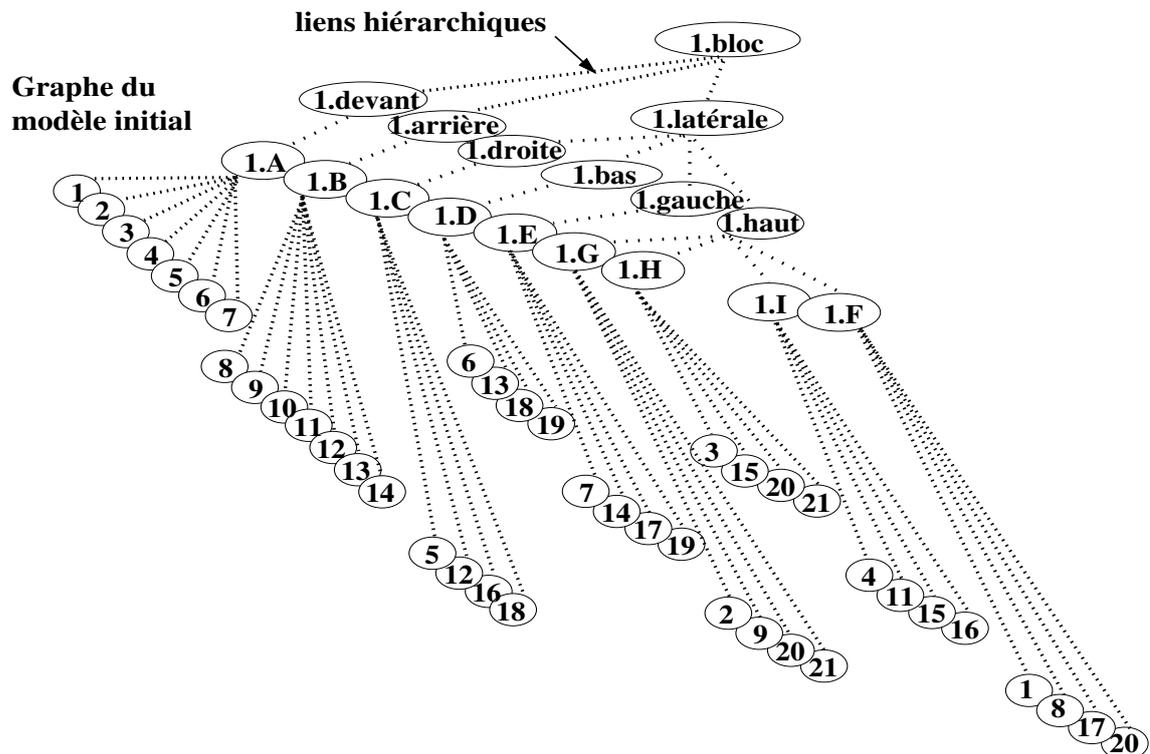


Figure 4.20: Graphe décrivant la structure d'agrégats invariants de la figure 4.19(a).

Le graphe de la figure 4.19(b) traduit la structuration du modèle géométrique résultant

de l'ajout d'une protrusion sur la coque du haut du bloc précédent. La protrusion est composée d'une coque du haut (2.haut qui correspond à la face 2.B), une coque latérale (2.latérale qui correspond à la face 2.A) et une coque du bas (2.bas qui correspond à la face 2.C). Enfin, pour signifier que la protrusion est posée sur la coque du haut du bloc extrudé, un agrégat 2.agré-prot est rajouté au graphe.

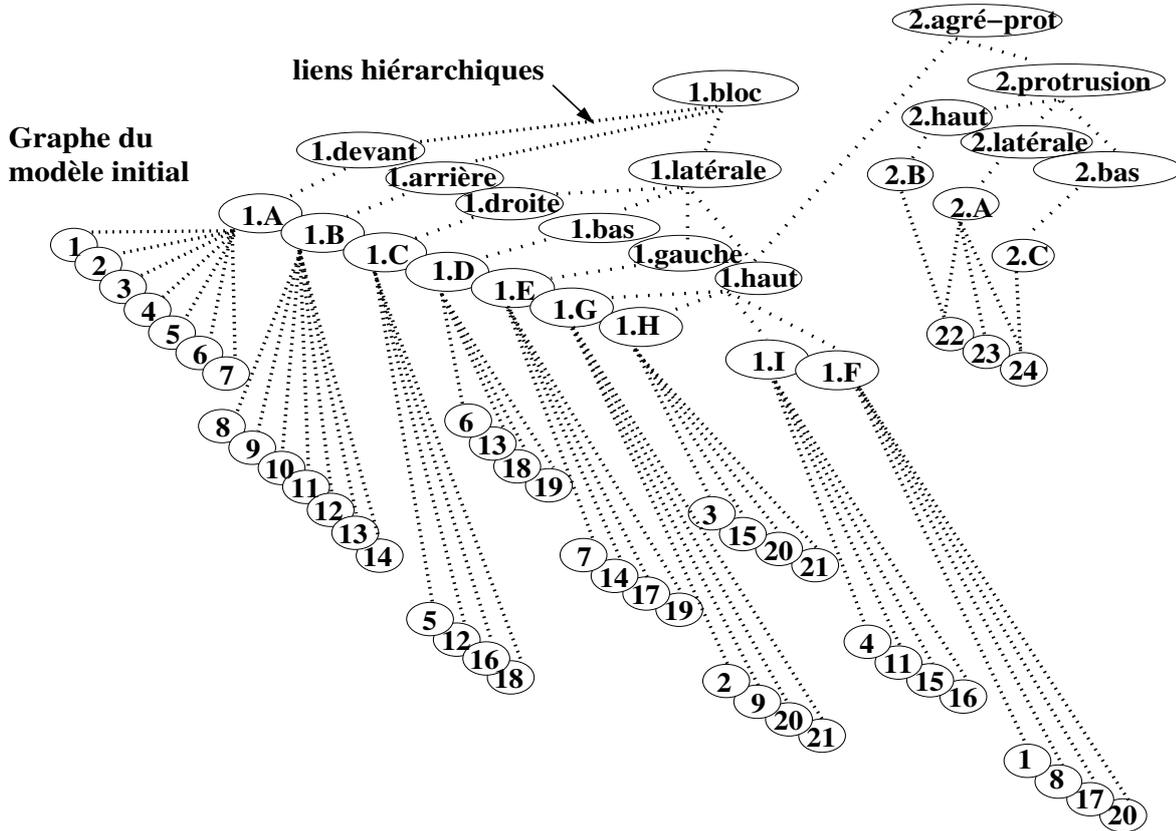


Figure 4.21: Graphe décrivant la structure d'agrégats invariants de la figure 4.19(b), (c) et (d).

Donc, la sélection de la face 3.2 du modèle de la figure 4.19(d) génère la liste L_{inv} suivante $\langle 1, haut \rangle, \langle 1, latérale \rangle, \langle 1, bloc \rangle, \langle 2, agré-prot \rangle$ puisque l'ancêtre invariant de 3.2 est la face 1.H et remonter le graphe des agrégats invariants à partir de cette face passe par les agrégats susmentionnés.

2. pour chaque agrégat d'entités E_{inv} appartenant à la liste L_{inv} , un agrégat contingent d'entités est calculé et rajouté à la liste L_{cont} . Un agrégat contingent d'entités est nécessairement un descendant historique d'un agrégat ancêtre invariant. Aussi, pour chaque E_{inv} un parcours topologique incrémental à partir de E est effectué et chaque entité topologique rencontrée dans ce parcours est ajoutée à E_{cont} à condition qu'elle soit un descendant historique d'une entité topologique appartenant à E_{inv} . De cette façon, E_{cont} est le plus grand composant connexe contenant E étant le descendant historique de E_{inv} . Par exemple, continuons avec l'exemple précédent de la sélection de la face 3.2, à l'étape 4 du processus de modélisation, dans la figure 4.19. Pour chaque agrégat invariant dans L_{inv} , un agrégat contingent d'entités est généré dynamiquement à travers la construction de composants

4.3. SYSTÈME DE NOMINATION HIÉRARCHIQUE

connexes utilisant un parcours topologique incrémental à partir de la face 3.2. Par exemple, le premier agrégat invariant $\langle 1, haut \rangle$ de la liste L_{inv} génère l'agrégat contingent d'entités $\langle 4, \langle 1, haut \rangle, \{4, 16, 11, 15, 27, 28, 29\} \rangle$. En effet, le parcours incrémental à partir de la face 3.2 est effectué sur le modèle géométrique selon l'adjacence topologique des entités. Chaque entité parcourue est utilisée à son tour pour continuer le parcours incrémental et elle ajoutée à l'agrégat contingent d'entités, si cette entité est un descendant historique appartenant à l'agrégat $\langle 1, haut \rangle$. Dans ce cas, il y a qu'une seule face parcourue 2.1. Les deux faces 3.2 et 2.1 sont constituée des arêtes $\{4, 16, 11, 15, 27, 28, 29\}$. En utilisant le même principe pour les autres agrégats invariants appartenant à L_{inv} , nous obtenons la liste L_{cont} suivante :

- $\langle 4, \langle 1, haut \rangle, \{4, 16, 11, 15, 27, 28, 29\} \rangle$,
- $\langle 4, \langle 1, latérale \rangle, \{4, 16, 11, 15, 27, 28, 29, 18, 12, 5, 6, 13, 19, 7, 14, 17, 1, 8, 20, 2, 9, 21, 24, 25, 26\} \rangle$,
- $\langle 4, \langle 1, bloc \rangle, \{4, 16, 11, 15, 27, 28, 29, 18, 12, 5, 6, 13, 19, 7, 14, 17, 1, 8, 20, 2, 9, 21, 24, 25, 26, 30, 31, 32, 33, 34, 35\} \rangle$,
- $\langle 4, \langle 2, agrgr - prot \rangle, \{15, 27, 28, 29, 4, 16, 11, 22, 23, 24\} \rangle$.

4.3.3 Appariement des agrégats

4.3.3.1 Appariement d'agrégats invariants d'entités

La méthode d'appariement d'agrégats invariants d'entités est similaire à la méthode d'appariement des arêtes invariants décrite dans la section 4.2.3.1. Dans le cas des agrégats invariants d'entités, la caractérisation $\langle numéro\ d'étape, ID_A \rangle$ est unique et non ambiguë durant les processus de construction et de réévaluation de modèles paramétriques. Alors, le calcul d'appariement est évident puisqu'il s'agit de mettre en correspondance les agrégats ayant exactement le même nom.

4.3.3.2 Appariement d'agrégats contingents d'entités

Apparier un agrégat contingent d'entités A^{ini} du modèle initial revient à calculer un ensemble $\Gamma^{réév}$ des agrégats d'entités $A_i^{réév}$ correspondant dans le modèle réévalué. Dans ce cadre, un agrégat d'entités $A_i^{réév}$ est dit correspondant à A^{ini} si

1. A^{ini} et $A_i^{réév}$ ont le même ancêtre invariant en terme d'agrégat ;
2. A^{ini} et $A_i^{réév}$ ont au moins une arête en commun liées par des relations d'appariement.

L'appariement des agrégats d'entités de dimension supérieure à 1 peut être donné soit en utilisant le voisinage topologique de ces agrégats soit directement en utilisant les appariements réalisés au niveau des arêtes composant ces agrégats. Le calcul des appariements ayant déjà été réalisé au niveau des arêtes, il est moins coûteux d'utiliser directement ces informations plutôt que les voisinages.

De façon pratique, le calcul des agrégats correspondant à A^{ini} est effectué à travers le processus de construction décrit dans la section 4.3.2.2. L'entité à partir de laquelle commence le parcours topologique incrémental du modèle géométrique réévalué est obtenue à travers les liens d'appariements entre les arêtes liant les arêtes du modèle réévalué à celles constituant l'agrégat d'entités A^{ini} .

Dans la figure 4.22, la réévaluation de l'agrégat contingent des faces 3.2 et 2.1 caractérisé par $\langle 4, \langle 1, haut \rangle, \{4, 16, 11, 15, 27, 28, 29\} \rangle$ (illustré en violet sur le modèle initial de la figure 4.22) génère l'ensemble

$$\Gamma^{réév} = \{ \langle 4, \langle 1, haut \rangle, \{27', 28', 29', 21, 3, 10, 15, 4, 11, 16\} \rangle \}$$

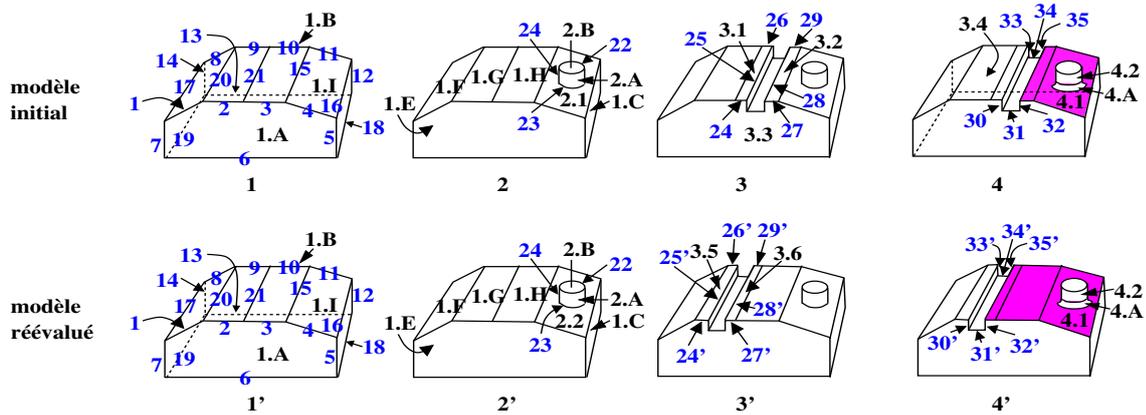


Figure 4.22: Appariement des agrégats contingents d'entités sur le modèle de la figure 4.19.

(c'est-à-dire, l'ensemble constitué de l'agrégat des faces 3.6, 1.H et 2.2, illustré en violet sur le modèle réévalué de la figure 4.22) parce que les arêtes 4, 16, 11, 15, 27 et 29 du modèle initial correspondent aux arêtes 4, 16, 11, 15, 3 et 10 du modèle réévalué.

De façon similaire, l'agrégat de faces (1.F, 3.1, 1.G) caractérisé par $\langle 4, \langle 1, haut \rangle, \{24, 25, 26, 21, 9, 2, 20, 8, 1, 17\} \rangle$ (illustré en violet sur le modèle initial de la figure 4.23) génère l'ensemble

$$\Gamma^{réév} = \{ \langle 4, \langle 1, haut \rangle, \{1, 17, 8, 20, 24', 25', 26'\} \rangle, \langle 4, \langle 1, haut \rangle, \{27', 28', 29', 21, 3, 10, 15, 4, 11, 16\} \rangle \}$$

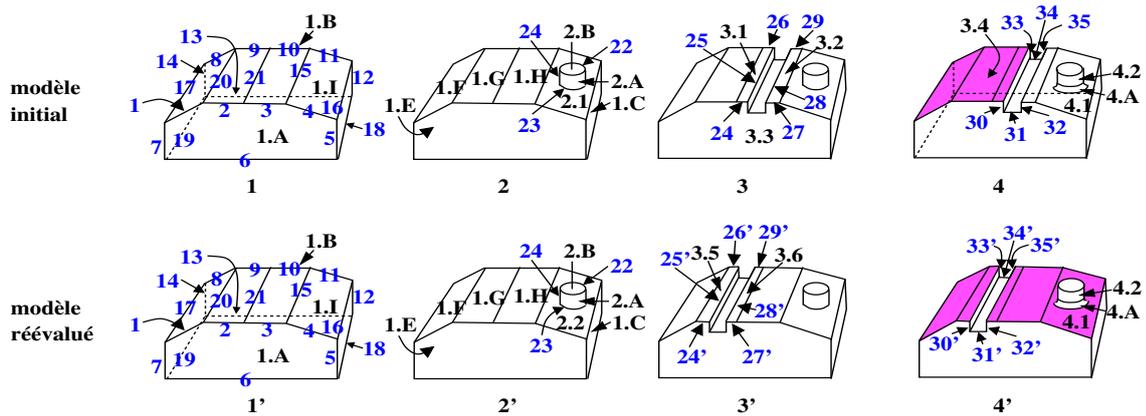


Figure 4.23: Appariement des agrégats contingents d'entités sur le modèle de la figure 4.19.

(c'est-à-dire, un ensemble de deux agrégats de faces constitués d'une part, des faces 1.F, 3.5, et, d'autre part des faces 3.6, 1.H, 2.2, illustrés en violet sur le modèle réévalué de la figure 4.23).

En effet, les arêtes 9 (resp. 2) du modèle initial sont respectivement appariées aux arêtes 26' et 29' (resp. les arêtes 24' et 27'), et le parcours incrémental à partir de ces deux couples d'arêtes génère deux agrégats connexes de faces.

4.4 Conclusion

Dans ce chapitre, nous avons présenté les principes et l'architecture de notre système hiérarchique de nomination persistante. Ainsi, nous avons défini une structure de nom et un mécanisme d'appariement pour les arêtes. Nous avons ensuite fait de même pour les entités et les agrégats d'entités de dimension supérieure.

Dans notre analyse des systèmes de nomination existant, nous avons montré les difficultés à généraliser les mécanismes de nomination en se basant sur les faces. Nous avons également montré que le caractère relativement stable des faces en dimension 3 ne reste pas un fait établi en dimension supérieure. De ce fait, nous avons mis en œuvre un système de nomination avec des mécanismes de caractérisation et d'appariement spécifiques aux arêtes. Puis, nous avons utilisé les arêtes et les relations d'appariements qui les lient pour caractériser et appairer les entités et les agrégats d'entités de dimensions supérieures.

L'architecture de notre système de nomination comporte plusieurs niveaux : le niveau des arêtes et les niveaux comportant chacun les entités et les agrégats d'entités de dimension donnée. Le niveau des arêtes est le niveau sur lequel repose, en grande partie, la pertinence des caractérisations et des appariements à effectuer aux niveaux supérieurs. C'est pourquoi nous avons porté une attention toute particulière :

- à la structure de nom utilisée pour caractériser les arêtes : à l'instar des noms d'arêtes proposées par les différentes approches de la littérature, une arête est caractérisée par un voisinage topologique et des informations géométriques utilisées pour lever les ambiguïtés qui peuvent apparaître dans le cas non-planaire. Cependant, dans notre approche on ne se contente pas uniquement des faces invariantes et des informations géométriques calculées à des points particuliers des arêtes (gradients, ordre dans un espace paramétrique, etc.). En effet, nous avons utilisé toutes les entités invariantes et les agrégats invariants d'entités incidents à l'arête et aux sommets constituant les bords de cette arête. Nous avons ainsi renforcé la robustesse et la sémantique de la caractérisation en utilisant le maximum de données invariantes constituant le voisinage de l'arête considérée. De plus, nous avons pris en compte la forme générale des objets comme élément principale qui permet à l'utilisateur de déterminer ce qu'il voit. Les informations géométriques qui en découlent permettent d'augmenter la pertinence des appariements jusqu'ici calculés lorsque des ambiguïtés apparaissent.
- au mécanisme d'appariement des arêtes contingentes : après la réévaluation d'un modèle paramétrique, la comparaison entre les caractérisations des arêtes ne permet pas d'obtenir des appariements correctes. Nous avons alors proposé un mécanisme qui s'appuie sur cette caractérisation pour isoler en premier les arêtes ayant les mêmes ensembles d'entités incidentes. Ensuite, nous avons utilisé les bords de ces arêtes pour déterminer le degré d'inclusion des arêtes du modèle initial dans celles du modèle réévalué, et inversement. C'est ce que nous avons appelé les relations de recouvrements. Enfin, nous avons recoupé les recouvrements établis dans les deux sens pour en extraire les appariements finaux. Avec ce procédé, nous avons souhaité prendre en compte le contexte topologique global entou-

rant les arêtes à apparier. En revanche, celui-ci n'est pas capable de trouver les relations de correspondance lorsque les arêtes ont le même voisinage topologique. Dès lors, nous avons mis en place une seconde phase qui consiste à comparer les informations géométriques des arêtes qui posent de telles ambiguïtés.

Enfin, le système hiérarchique de nomination que nous avons mis en œuvre répond positivement aux différentes exigences des systèmes de nomination. Il permet ainsi :

- la caractérisation de façon unique et non ambiguë les entités et les agrégats d'entités de toute dimension ;
- l'appariement des entités et des agrégats d'entités de toute dimension en utilisant les liens de correspondance établis entre les arêtes ;
- de prendre d'avantage en compte les intentions de conception en proposant la possibilité de nommer des agrégats d'entités de différentes dimensions.

Chapitre 5

Conclusion générale

Les modèles paramétriques sont structurés autour d'une couche de spécification paramétrique, un noyau géométrique et une couche de nomination. La spécification paramétrique enregistre tout le processus de modélisation paramétrique à travers un ensemble d'opérations de modélisation. Le noyau géométrique contient le modèle géométrique, alors que la couche de nomination permet la gestion des références utilisées par les opérations de modélisation pour désigner les entités topologiques d'un modèle géométrique. Cette architecture cause un certain nombre de problèmes. Nous pouvons citer :

- les problèmes liés à l'échange de modèles paramétriques entre systèmes paramétriques différents ;
- l'édition de la spécification paramétrique en modifiant l'ordre ou la nature des opérations de modélisation qui la constituent ;
- le problème de la nomination persistante.

Nous nous sommes intéressés dans le cadre de ce travail au problème de la nomination persistante, c'est-à-dire au problème de la désignation des entités topologiques lorsque les paramètres des opérations de modélisation changent et avec elles la topologie du modèle géométrique. Dès lors, la transformation de la topologie du modèle géométrique rend certaines références caduques et engendre en conséquence des réévaluations incorrectes. En outre, la résolution de ce problème passe par la satisfaction d'un certain nombre d'exigences, à savoir :

1. la nécessité d'un nom unique et non ambigu pour chaque entité référencée (1^{er} problème) ;
2. la nécessité d'un mécanisme capable de mettre en correspondance les entités *semblables* après la réévaluation du processus de modélisation initiale (2^{ème} problème) ;
3. la capture des intentions de conception en permettant la désignation d'un autre type d'entités, les agrégats (3^{ème} problème).

Dans ce contexte, plusieurs solutions ont été proposées pour résoudre le problème de la nomination persistante. Chacune d'entre elles propose des mécanismes de caractérisation et d'appariement différents pour dénouer les difficultés posées par différentes configurations (nomination des arêtes, cas des objets planaires ou non planaires, traitement des agrégats de faces, etc.). Cependant, toutes s'accordent sur un certain nombre de points communs comme par exemple le fait que les mécanismes de caractérisation soient basés sur les entités invariantes, que ces entités invariantes soient des faces à cause de leurs caractères relativement stables en $3D$, etc.

L'analyse que nous avons faite des points communs et des points différents de ces solutions, nous a permis de mettre en exergue un certain nombre de limites. En effet, nous avons montré à travers cette analyse les lacunes de ces systèmes pour traiter de façon homogène les entités et

les agrégats d'entités de dimension supérieure à 2. Nous avons également fait ressortir les limitations constatées dans la nomination des arêtes résultant de l'interaction d'objets non planaires. La résolution de ces problèmes et la sauvegarde des résultats obtenus jusqu'à présent dans le domaine de la nomination persistante ont constitué l'objet de notre étude.

Dans la première section de cette conclusion, nous présentons le travail que nous avons réalisé en montrant notre apport, les résultats obtenus ainsi que les limites de notre modèle. L'existence de ces limites nous amène à présenter, dans la deuxième section, les prolongements intéressants et les perspectives de recherche.

5.1 Travail réalisé

Dans le deuxième chapitre, nous montrons les difficultés toujours présentes dans la résolution du problème de la nomination persistante. Pour ce faire, nous avons commencé, dans un bref historique, par aborder la genèse des systèmes paramétriques. Nous avons, en outre, mis l'accent sur le passage d'une représentation géométrique figée vers une autre pilotée par une spécification paramétrique. Ensuite, nous avons évoqué plus en détails la problématique de la nomination persistante. Nous avons alors discuté de trois fonctions essentielles que doit fournir un système de nomination persistante : apporter un nom, apparier les entités correspondantes et capturer les intentions de conception. Nous avons mis en exergue les concepts qui lient ou différencient les solutions présentées dans la littérature. Ainsi, nous avons mis en exergue les difficultés liées au concept de caractérisation basée sur les faces invariantes en $3D$. Les mécanismes de caractérisation et d'appariement qui s'appuient sur cette notion deviennent hétérogènes du fait qu'il existe un mécanisme spécifique par dimension d'entités. De plus, nous avons remarqué l'apport des informations géométriques pour résoudre les ambiguïtés qui peuvent apparaître lors de l'interaction d'objets non planaires. Cet apport n'est pas efficace dans tous les cas de figures. Il constitue, par conséquent un autre point à améliorer dans notre étude. La nécessité de mettre en place des mécanismes de nomination remédiant aux difficultés précédentes, nous ont amenés, dans le dernier chapitre, à définir un système hiérarchique de nomination persistante.

Mais avant d'y parvenir, notre système hiérarchique de nomination doit être au courant des évolutions topologiques se déroulant au niveau de noyau géométrique. Cette préoccupation est traitée dans le troisième chapitre. En effet, nous avons proposé un ensemble de mécanismes permettant le suivi des événements de création, scission, fusion, modification et suppression d'entités topologiques. Ces mécanismes définis de façon indépendante de tout modèle topologique ont fait l'objet d'une implantation dans le modèle des cartes généralisées. L'étude des cartes généralisées a permis de mettre en évidence un ensemble de quatre opérations (suppression, contraction, insertion et éclatement) à la base desquelles peuvent être définies toutes les autres opérations. Par ailleurs, l'étude des opérations de base nous a permis de mieux appréhender les mécanismes de suivi d'événements implantés en utilisant la structure de « tag ».

Dans le quatrième chapitre, nous avons défini notre système hiérarchique de nomination persistante. Pour ce faire, nous avons structuré les entités référencées par un processus de modélisation dans un graphe. Ce graphe comprend des entités de toute dimension (sommets, arêtes, faces, volumes, hypervolumes, etc.) et des agrégats de ces entités afin d'appréhender différents niveaux de granularité. Le système se base sur les arêtes et non sur les faces comme le font les autres solutions présentées dans la littérature. Ceci implique de pouvoir mettre en œuvre d'abord des mécanismes de caractérisation et d'appariement robustes au niveau des arêtes. Nous

5.1. TRAVAIL RÉALISÉ

avons cherché à préserver cette robustesse pour *toutes* les entités et tous les agrégats d'entités exprimant la propriété d'invariance dans le voisinage topologique d'une arête. De plus, lorsque ces informations d'ordre topologique ne suffisent pas, d'autres informations d'ordre géométriques sont utilisées pour lever les ambiguïtés qui peuvent survenir. Ensuite, en se basant sur la structure précédente de graphe et les résultats de ces appariements, le système que nous avons développé caractérise et apparie les entités et les agrégats d'entités de dimension supérieure.

5.1.1 Apports et résultats

Pour établir une nomination persistante de tout type d'entité (atomique ou agrégat), en toute dimension, dans un contexte où les objets planaires ou non interagissent, nous avons proposé un système hiérarchique basé sur les arêtes. Le système que nous avons proposé s'appuie sur un graphe reflétant la structure hiérarchique de la topologie du modèle. Il est structuré sous formes d'agrégats et de sous-agrégats d'entités dont le niveau le plus bas est le niveau des arêtes. Nous avons été amenés à définir d'abord une structure de nom et un mécanisme d'appariement d'arêtes robustes pour faire face aux différentes transformations topologiques. Ensuite, le nom est l'appariement des entités et des agrégats d'entités de dimension supérieur se base sur les données contenues au niveau des arêtes. Au-delà de cette architecture, notre contribution est la suivante :

- la caractérisation des arêtes : nous avons expliqué que le nom d'une arête est constitué d'informations issues de son voisinage topologique et d'autres informations d'ordre géométriques. Le premier type d'informations est utilisé de façon systématique alors que le second intervient lorsque plusieurs arêtes ont le même voisinage topologique. Dans ce contexte, nous avons considéré le voisinage topologique comme un ensemble d'entités invariantes de dimension $(n - 1)$ ¹ incidentes à l'arête considérée, et deux ensembles d'entités invariantes et/ou d'agrégats invariants d'entités de toute dimension incidents à ses bords. Par ailleurs, les informations géométriques sont calculées après avoir décomposé les faces non-planaires en fonction du critère de courbures. Elles sont constituées du couple de sous-surfaces incidentes à l'arête considérée et résultant du processus de décomposition précédent et de l'orientation des arêtes. De plus, l'ancêtre invariant de chaque entité contingente est calculé à partir d'une structure qui trace son évolution au cours du processus de modélisation, le journal de bord ;
- la caractérisation des agrégats : nous avons proposé un graphe qui reflète la structure hiérarchique du modèle topologique utilisé. Nous avons également proposé une caractérisation des agrégats contingents en utilisant leurs ancêtres invariants qui découle de ce graphe et de l'ensemble des arêtes qui constituent leurs bords ;
- l'appariement des arêtes : nous avons proposé un mécanisme d'appariement qui permet de faire correspondre des arêtes caractérisées par la seule donnée de leurs voisinages topologiques, puis les arêtes caractérisées avec des informations géométriques (c'est-à-dire, celles qui présentent des ambiguïtés). Le principe que nous avons adopté dans chaque cas permet de mettre en correspondances des arêtes ayant des caractérisations similaires. En effet, le mécanisme de nomination calcul, en fonction des noms d'arêtes, des relations de recouvrement des arêtes du modèle initial sur les arêtes du modèle réévalué, et inversement. Ensuite, un recouplement des liens de recouvrement établis dans les deux sens, permet le

1. n correspond à la dimension dans laquelle est plongée le modèle géométrique.

calcul effectif des liens d'appariement.

- l'appariement d'agrégats : nous avons proposé un mécanisme d'appariement des agrégats contingents d'entités qui se base sur celui des arêtes. En effet, ces agrégats sont caractérisés par un ensemble d'arêtes et leur appariement passe par l'appariement des arêtes qui les constituent. De plus, à ce niveau, l'appariement des entités contingentes de dimension supérieure ou égale à 2 (dites entités atomiques) et les autres types d'agrégats est pris en compte de la même façon.

5.1.2 Domaine de validité des résultats

La solution proposée pour le problème de la nomination persistante dans un modèle paramétrique, s'applique à des modèles géométriques de toute dimension, supportant des plongements planaires et non planaires. Cependant, quelques limites doivent être évoquées. Ainsi, nous pouvons citer l'exemple de surfaces non planaires qui présentent peu de courbure. Dans ce cas, les sous-surfaces résultant de l'opération de décomposition pour le calcul des informations géométriques sont étroites. Par conséquent, avec des variations numériques mineures, les arêtes résultant de l'intersection de faces non planaires peuvent basculer facilement d'une sous-surface à une autre, faussant ainsi les résultats d'appariements. Par ailleurs, le fait de monter en dimension supérieure ou égale à 4 rend plus complexe l'appréhension de la géométrie. La question qui se pose alors est : que devient la perception qui guide l'intention de l'utilisateur ?

Notre proposition de réponse au problème de la nomination persistante dans les modèles paramétriques amène donc d'autres questionnements, et laisse ouvert plusieurs axes de recherche.

5.2 Perspectives

Nous pensons que les perspectives de recherche et les prolongements intéressants de notre travail sont nombreux.

Tout d'abord, il est important de compléter l'implantation de notre système hiérarchique de nomination dans le modèleur Moka. En effet, la majorité des composantes de notre système de nomination ont été implantées. Il manque uniquement la partie prenant en compte la nomination des entités et des agrégats d'entités de dimension supérieure à 1.

Deuxièmement, il serait important également de tester notre système dans le cadre d'une application spécifique. Dans le cadre du projet VISA², nous envisageons, très prochainement, de pouvoir utiliser ce système de nomination dans une application liée à la reconstruction de bâtiments anciens à partir de fouilles archéologiques.

Troisièmement, il serait intéressant d'étudier le problème de la nomination persistante lorsqu'il y a édition de la spécification paramétrique (par exemple, lorsqu'il y a permutation des opérations de modélisation). Cette étude est importante pour valider notre modèle de nomination dans le cadre d'utilisations plus large.

Quatrièmement, il serait intéressant de structurer la spécification paramétrique sous forme d'arbre (comme le modèle CSG) et d'étudier la réévaluation partielle ou totale du modèle paramétrique. L'objectif est de permettre une plus grande flexibilité dans le processus de modélisation et d'optimiser l'opération de réévaluation.

2. VISA : VISualisation de données Archéologiques et architecturales.

5.3 Synthèse

Établir des appariements entre les entités et les agrégats d'entités est un problème difficile. Il devient encore plus compliqué lorsque des ambiguïtés apparaissent ou lorsque le nombre d'éléments invariants n'est pas assez conséquent pour dénouer l'appariement de $m > 1$ entités ou agrégats d'entités du modèle initial avec $n > 1$ entités ou agrégats d'entités du modèle réévalué. Il n'est ainsi pas possible de poser formellement et de façon non ambiguë les problèmes de caractérisation et d'appariement, en particulier lorsque le concepteur lui-même ne saurait exprimer l'appariement conforme à ses attentes. La solution que nous avons proposée pour résoudre le problème de la nomination persistante est partiellement fondée sur des heuristiques. Néanmoins, elle permet de réaliser des appariements « raisonnables » là où les autres méthodes de la littérature sont prises en défaut.

Bibliographie

- [ABC⁺00] C. Armstrong, A. Bowyer, S. Cameron, J. Corney, G. Jared, R. R. Martin, A. Middleditch, M. Sabin, and J. Salmon. *A Geometric Interface for Solid Modelling Specification and Report*. Information Geometers Ltd, 2000.
- [Agb02] D. Agbodan. *Nomination persistante dans un modèle paramétrique. Identification non-ambiguë et appariement générique d'entités topologiques*. PhD thesis, Université de Poitiers, Poitiers, 2002.
- [AM88] G. Aldefeld and G. Mayer. Variations of geometries based on a geometric-reasoning method. *Computer Aided-Design*, 20(3) :117–126, 1988.
- [AM95] R. Anderl and R. Mendgen. Parametric design and its impact on solid modeling applications. In *SMA '95 : Proceedings of the third ACM symposium on Solid modeling and applications*, pages 1–12, New York, NY, USA, 1995. ACM.
- [AMP99] D. Agbodan, D. Marcheix, and G. Pierra. A data model architecture for parametrics. *Journal for Geometry and Graphics*, 3(1) :17–38, 1999.
- [AMP00] D. Agbodan, D. Marcheix, and G. Pierra. Persistent naming for parametric models. In *in WSCG'2000, Vol*, pages 17–38, 2000.
- [AMPT03] D. Agbodan, D. Marcheix, G. Pierra, and C. Thabaud. A topological entity matching technique for geometric parametric models. In *SMI '03 : Proceedings of the Shape Modeling International 2003*, pages 235–245, Washington, DC, USA, 2003. IEEE Computer Society.
- [Arb82] P. Arbab. *Requirements and Architecture of CAM oriented CAD Systems for Design and Manufacture of Mechanical Parts*. PhD thesis, University of California, Los Angeles, California, USA, 1982.
- [BADSM08] M. Baba-Ali, G. Damiand, X. Skapin, and D. Marcheix. Insertion and expansion operations for n -dimensional generalized maps. In *Proceedings of 14th IAPR International Conference on Discrete Geometry for Computer Imagery*, volume 4992 of *LNCS*, pages 141–152. Springer-Verlag Berlin Heidelberg, April 2008.
- [BaMS09] M. Baba-ali, D. Marcheix, and X. Skapin. A method to improve matching process by shape characteristics in parametric systems. *Computer Aided Design & Applications*, 6(3) :341–350, June 2009.

-
- [Bar88] B.A. Barsky. *Computer graphics and geometric modeling using Beta-splines*. Springer-Verlag New York, Inc., New York, NY, USA, 1988.
- [Bau75] B.G. Baumgart. A polyhedron representation for computer vision. *Proceeding of AFIPS, National Computer Conference*, 44 :589–596, 1975.
- [BB00] R. Bidarra and W.F. Bronsvort. Semantic feature modeling. *Computer Aided Design*, 32(3) :201–225, 2000.
- [BD94] Y. Bertrand and J.F. Dufour. Algebraic specification of a 3d-modeler based on hypermaps. *CVGIP : Graphical Models and Image Processing*, 56(1) :29–60, 1994.
- [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.
- [BHS80] I.C. Braid, R.C. Hillyard, and I.A. Stroud. Stepwise construction of polyhedra in geometric modeling. In *Mathematical Methods in Computer Graphics and Design Academic Press*, pages 123–141, 1980.
- [BNB05] R. Bidarra, P.J. Nyirenda, and W.F. Bronsvort. A feature-based solution to the persistent naming problem. *Computer Aided Design & Applications*, 2(1-4) :5176–526, 2005.
- [BP07] D. Bechmann and B. Péroche. *Informatique graphique, modélisation géométrique et animation*. Paris, France, editions lavoisier edition, 2007. ISBN 978-2-7462-1514-6.
- [Bra75] I.C. Braid. the synthesis of solids bounded by many faces. *Communication of the ACM*, 18(4) :209–216, 1975.
- [Bri93] E. Brisson. Representing geometric structures in d dimensions : topology and order. *Discrete and Computational Geometry*, (1) :327–426, 1993.
- [BS85] R. Bryant and D. Singerman. Foundations of the theory of maps on surfaces with boundaries. *Quart. Journal of Math. Oxford*, 2(36) :17–41, 1985.
- [BS91] J. Bloomenthal and K. Shoemake. Convolution surfaces. *Computer Graphics*, 25(4) :251–256, 1991. Siggraph proceedings.
- [BS01] B.P. Bettig and J.J. Shah. Derivation of a standard set of geometric constraints for parametric modeling and data exchange. *Computer-Aided Design*, 33(1) :17–33, 2001.
- [But79] M. Buthion. Un programme qui résoud formellement des problèmes de construction géométriques. *RAIRO*, 13(1) :73–106, 1979.
- [BW90] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2) :190–116, mars 1990. Proceedings of Symposium on Interactive 3D Graphics.

- [Béz86] P. Bézier. *Courbes and surfaces*. Hermès, Paris, 1986.
- [CAM88] T.C. Chang, D.C. Anderson, and O.R. Mitchell. Qtc - an integrated design/manufacturing/inspection system for prismatic parts. In *ASME Computers in Engineering Conference*, pages 417–426, 1988.
- [Cas85] P.D. Casteljaou. *Formes à pôles*. Hermès, Paris, 1985.
- [CBC⁺01] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans. Reconstruction and representation of 3d objects with radial basis functions. *Computer Graphics Proceedings*, pages 67–76, 2001.
- [CD88] J. Cunningham and J.R. Dixon. Designing with features : The origin of failures. In *ASME Computers in Engineering Conference*, pages 237–243, San Francisco, California, USA, 1988.
- [CFM⁺94] P. Cigogni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution modeling and visualization of volume data based on simplicial complexes. In *Symposium on Volume Visualization*, pages 19–26, 1994.
- [Che95] X. Chen. *Representation, Evaluation and Editing of Feature-Based and Constraint-Based design*. PhD thesis, Purdue University, West Lafayette, Indiana, 1995.
- [CK88] H. Chiyokura and F. Kimura. A new surface interpolation method for irregular curve models. *Computer Graphics Forum*, 3 :209–218, 1988.
- [Coo87] S. Coons. *Méthodes matricielles*. Hermès, Paris, 1987.
- [CT92] M.R. Cutkosky and J.M. Tenenbaum. Toward a framework for concurrent design. *International Journal of Systems Automation : Research and Application*, 1(3) :239–361, 1992.
- [dB78] C. de Boor. *A practical guide to splines*. Springer Verlag, Berlin, 1978.
- [DL89] D. Dobkin and M. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, (4) :3–32, 1989.
- [DL03] G. Damiani and P. Lienhardt. Removal and contraction for n-dimensional generalized maps. In *Proceedings of 11th Discrete Geometry for Computer Imagery*, volume 2886 of *LNCS*, pages 408–419, Naples, Italy, November 2003.
- [DMS98] J.F. Dufourd, P. Mathis, and P. Schreck. Geometric construction by assembling solved subfigures. *Artificial Intelligence*, 99(1) :73–119, 1998.
- [EL93] H. Elter and P. Lienhardt. Different combinatorial models based on the map concept for the representation of different types of cellular complexes. In *Proceedings of IFIP TC 5/WG II Working Conference on Geometric Modeling in Computer Graphics*. Falcidieno B., Kunii T.L. Ltd, 1993.

- [EL94] H. Elter and P. Lienhardt. Cellular complexes as structured semi-simplicial sets. *International Journal of Shape Modeling*, 1(2) :191–217, 1994.
- [FB00] J. Françon and Y. Bertrand. Topological 3d-manifolds : a statistical study of the cells. *Theor. Comput. Sci.*, 234(1-2) :233–254, 2000.
- [FCG00] E. Ferley, M.P. Cani, and J.D. Gascuel. Practical volumetric sculpting. *The Visual Computer*, 16(7) :469–480, 2000.
- [FDF⁺95] J.D. Foley, A. Van Dam, S.K. Feiner, J.F. Hughes, and R.L. Phillips. *Introduction à l'infographie*. France, editions addison-wesley edition, 1995. ISBN 2-879-08058-4.
- [FH97] I. Fudos and C. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics*, 16(2) :179–216, 1997.
- [Flo87] L. De Floriani. A graph-based approach to object feature recognition. In *Proceedings of the 3rd ACM symposium on Computational Geometry*, 1987.
- [FMJ05] S. Foufou, D. Michelucci, and J.P. Jurzak. Numerical decomposition of geometric constraints. In *Proc. of Solid and Physical Modeling, SPM*, pages 143–151, 2005.
- [FMP03] L. De Floriani, F. Morando, and E. Puppo. Representation of non-manifold objects through decomposition into nearly manifold parts. 8th *A.CM. Symposium on Solid modeling and Applications*, pages 304–309, 2003.
- [Gar91] Y. Gardan. *La CFAO introduction, techniques et mise en œuvre*. 3^{ème} édition, Hermès,, Paris, 1991.
- [GH91] T. Galyean and J. Hughes. Sculpting : An interactive volumetric modeling technique. *Computer Graphics*, 25(4) :267–274, 1991. Proceedings of SIGGRAPH'91.
- [GM93] Y. Gardan and C. Minich. *Feature-based models for CAD/CAM and their limits*. Elsevier Science Publishers, 1993.
- [GMP03] Y. Gardan, C. Minich, and E. Perrin. Boolean operations on feature-based models. *Journal of WSCG (Winter School of Computer Graphics)*, 1–3(11) :265–289, 2003.
- [GR74] W.J. Gordon and R.F. Riesenfeld. Bernstein-bézier methods for the computer-aided design of freeform curves and surfaces. *Communication of the ACM*, 21(2) :293–310, 1974.
- [Gui06] N. Guiard. *Construction de modèles géologiques 3D par co-raffinement de surfaces*. PhD thesis, École des Mines de Paris, Paris, 2006.

- [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.
- [HK01] C.M. Hoffmann and K. Kim. Towards valid parametric cad models. *Computer-Aided Design*, 33(1) :81–90, 2001.
- [HR98] J. Han and A.A.G. Requicha. Feature recognition from cad models. *IEEE Computer Graphics & Applications*, 18(2) :80–94, 1998.
- [JaSRMvVp04] R. Joan-arinyo, A. Soto-Riera, S. Marta-vila, and J. Vilaplana-pasto. Revisiting decomposition analysis of geometric constraint graphs. *Computer Aided-Design*, 36 :123–140, 2004.
- [JC88] S. Joshi and T.C. Chang. Graph-based heuristics for recognition of machined features from a 3d solid model. *Computer-Aided Design*, 20(2) :58–66, 1988.
- [JT80] C.L. Jackins and S.L. Tanimoto. Octrees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3) :249–270, 1980.
- [JTNM06] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraint systems : a survey. *Int. J. Comput. Geometry Appl.*, 16(5-6) :379–414, 2006.
- [Kri95] J. Kripac. A mechanism for persistently naming topological entities in history-based parametric solid models. In *SMA '95 : Proceedings of the third ACM symposium on Solid modeling and applications*, pages 21–30, New York, NY, USA, 1995. ACM.
- [KY92] K. Kitajima and M. Yamaguchi. A shell oriented boolean set operations algorithm suited for the b-reps based on boundary edge loops. *Systems and computers*, 23(6), 1992.
- [Kyp80] L. Kyprianou. *Shape classification in computer aided design*. PhD thesis, Cambridge University, Cambridge, Massachusetts, USA, 1980.
- [LDS86] S. Luby, J.R. Dixon, and M.K. Simmons. Creating and using feature databases. *Computers in Mechanical Engineering*, 5(3) :25–33, 1986.
- [LF87] Y.C. Lee and K.S. Fu. Machine understanding of csg : Extraction and unification of manufacturing features. *IEEE Computer Graphics & Applications*, 7(1) :20–32, 1987.
- [Lie91] P. Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer Aided Design*, 23(1) :59–82, 1991.

-
- [Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3) :275–324, 1994.
- [LL95] V. Lang and P. Lienhardt. geometric modeling with simplicial sets. In *Proceedings of Pacific Graphics'95*, 1995.
- [LL96] V. Lang and P. Lienhardt. Simplicial sets and triangular patches. In *Proceedings of CGI'96*, 1996.
- [Mar96] D. Marcheix. *Modélisation géométrique d'objets non-variétés : construction, représentation et manipulation*. PhD thesis, Bordeaux, France, 1996.
- [Mar06] D. Marcheix. A persistent naming of shells. *International Journal of CAD/CAM*, 6(1) :121–133, 2006.
- [MF06] D. Michelucci and S. Foufou. The witness configuration method. *Computer Aided Design*, 38(4) :284–299, 2006.
- [MH05] D. Mun and S. Han. Identification of topological entities and naming mapping for parametric cad model exchanges. *International Journal of CAD/CAM*, 5(1) :69–81, 2005.
- [Min85] R.H. Miner. *A method for the representation and manipulation of geomandric features in a solid model*. PhD thesis, Cambridge, Massachusandts, USA, 1985.
- [Män88] M. Mäntylä. *An introduction to Solide modelling*. Computer Science Press, 1988.
- [MOP89] M. Mäntylä, J. Opas, and J. Puhakka. Generative process planning of prismatic parts by feature relaxation. In B. Ravani, editor, *Proceedings of 15th ASME Design Automation Conference*, pages 49–60, New York, USA, 1989.
- [Mor97] M.E. Mortenson. *Geometric Modeling*. 2^{ème} édition. New York, USA, editions jhon wiley & sons edition, 1997. ISBN 0-471-12957-7.
- [MP02] D. Marcheix and G. Pierra. A survey of the persistent naming problem. In *SMA '02 : Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 13–22, New York, NY, USA, 2002. ACM.
- [MS82] M. Mäntylä and R. Sulonen. Gwb : A solide modeler with the euler operators. *IEEE Computer Graphics*, 2(7) :17–31, 1982.
- [MT83] M. Mantyla and M. Tamminen. Localized set operations for solid modeling. *Computer and Graphics*, 17(3), 1983.
- [MT88] D. Ma and R. Tang. Realizing the boolean operations in solid modeling technique via directed loops. *Computer and Graphics*, 12(3/4), 1988.
- [Owe91] J. Owen. Algebraic solution for geometry from dimensional constraints. In *Proc. of Solid Modeling and CAD/CAM Applications*, pages 397–407, 1991.

- [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.
- [PBCF94] A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci. Dimension independent modeling with simplicial complexes. *A.C.M. Transactions on Graphics*, 12(1) :56–102, 1994.
- [Per05] E. Perrin. Opérations booléennes : trente années d’un algorithme toujours au coeur des systèmes de cao. *Revue internationale d’ingénierie numérique (de la conception collaborative à la simulation)*, 1(3) :265–289, 2005.
- [PF01] R. Perry and S. Frisken. kizamu : a system for sculpting digital characters. In *Computer Graphics*, pages 47–56. siggraph proceedings, 2001.
- [PPG94] G. Pierra, J.C. Potier, and P. Girard. The ebp system : Exemple based programming. In *Workshop on Graphic and Modeling in Science and Technology*, pages 124–140, Coimbra, Portugal, 1994. Springer Verlag series.
- [PRS94] T. Peters, D.W. Rosen, and V. Shapiro. A topological model of limitations in design for manufacturing. *Research in Engineering Design*, 6(4) :223–233, 1994.
- [PT97] L. Piegl and W. Tiller. *The NURBS book*. second edition. Springer Verlag, Berlin, 1997.
- [PW87] M.J. Pratt and P.R. Wilson. Conceptual design of a feature-oriented solid modeler. In *Draft Document 3B, General Electric Corporate R&D*, 1987.
- [Reg95] W.C. Regli. *geometric Algorithms for Recognition of Features from Solid Models*. PhD thesis, Computer Science Department, University of Maryland, USA, 1995.
- [Req77] A.A.G. Requicha. *Mathematical Model of Rigid solid Objects*. PhD thesis, New York, USA, 1977.
- [Req80] A.A.G. Requicha. Representation for rigid solids : theory, methods ans systems. *Computing survey*, 22(4) :437–463, 1980.
- [Rol91] D. Roller. Advanced methods for parametric design. In *Geometric Modeling, Methods and Application*, 1991.
- [Ros90] J.R. Rossignac. Issues on feature-based editing and interrogating of solid models. *Computer Graphics*, 14(2) :149–172, 1990.
- [RR92] A.A.G Requicha and J.R. Rossignac. Solid modeling and beyond. *IEEE Computer Graphics & Applications*, 12(5) :31–44, 1992.
- [RS98] S. Raghathama, , and S. Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Transactions on Graphics*, 17(4) :259–286, 1998.

-
- [Sch02] P. Schreck. Résolution symbolique de contraintes géométriques. In *Habilitation à diriger des recherches, Université Louis Pasteur*, 2002.
- [SDL06] C. Simon, G. Damiand, and P. Lienhardt. nd generalized map pyramids : definition, representations and basic operations. *Pattern Recognition*, 39(4) :527–538, 2006.
- [SHk93] O.W. Salomons, F.J.A.M. Van Houten, and H.J.J kals. Review of research in feature-based design. *Journal of manufacturing Systems*, 12(2) :113–132, 1993.
- [Sim06] C. Simon. *Définition et étude des pyramides généralisées nD : application pour la segmentation multi-échelle d'images 3D*. PhD thesis, Université de Poitiers, Poitiers, 2006.
- [SL01] X. Skapin and P. Lienhardt. Using cartesian product for animation. *Journal of Visualization and Computer Animation*, 12(3) :131–144, 2001.
- [SM95] J.J. Shah and M. Mäntylä. *Parametric and feature-based CAD/CAM : Concepts, Techniques, Applications*. John Wiley & Sons Inc., USA, 1995.
- [SR88a] J.J. Shah and M.T. Rogers. Expert form feature modeling shell. *Computer Aided Design*, 20(9) :515–524, 1988.
- [SR88b] J.J. Shah and M.T. Rogers. Functional requirements and conceptual design of the feature-based modeling system. *Computer-Aided Engineering*, 5(1) :9–15, 1988.
- [SR04] S. Spitz and A. Rappoport. Integrated feature-based and geometric cad data exchange. In *SM '04 : Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 183–190, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [SSR⁺88] J.J. Shah, P. Sreevalsan, M.T. Rogers, R. Billo, and A. Mathew. Current status of features technology. In *Report task 0, Technical Report R-88-GM-04.4, CAM-I*, Arlington, Texas, USA, 1988.
- [SSS94] J.J. Shah, Y. Shen, and A. Shirur. Determination of machining volumes from extensible sets of design features. In J.J. Shah, D.S. Nau, and M. Mäntylä, editors, *Advances in feature based manufacturing*, pages 129–157. Elsevier Science Publishers B.V., 1994.
- [Sut63] I.E. Sutherland. a man-machine graphical communication system. In *AFIPS Spring Joint Computer Conference*, pages 329–346, 1963.
- [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.

- [SW97] Y.S. Suh and M.J. Wozny. Interactive feature extraction for a form feature conversion system. In C.M. Hoffmann and W. Bronsvort, editors, *Solid Modeling'97, Proceedings of the 4th ACM symposium on Solid Modeling and Applications*, pages 111–122. ACM Press, 1997.
- [TK94] S. Trika and L. Kashyap. Geometric reasoning for extraction of manufacturing features in iso-oriented polyhedrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11) :1087–1100, 1994.
- [TO99] G. Turk and J. O'Brien. Shape transformation using variational implicit functions. *Computer Graphics (siggraph proceedings)*, pages 335–342, 1999.
- [Wei86] K. Weiler. Boundary graph operators for non-manifold geometric modeling topology representations. In *Proceedings IFIP WG 5.2 Working Conference*, Rensselaerville, USA, 1986.
- [WK93] D. Waco and Y.S. Kim. Geometric reasoning for machining features using convex decomposition. In J. Rossignac, J. Turner, and G. Allen, editors, *Solid Modeling'93, Proceedings of the 2nd ACM symposium on Solid Modeling and Applications*, pages 323–332. ACM Press, 1993.
- [WMW86] G. Wyvill, C. McPheanders, and B. Wyvill. Data structures for soft objects. *The visual computer*, 2(4) :227–234, 1986.
- [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.
- [Woo82] T. Woo. Feature extraction by volume decomposition. In *Proc. of Conference on CAD/CAM Technology in Mechanical Engineering*, pages 76–94, 1982.
- [WSMF06] J. Wintz, P. Schreck, P. Mathis, and A. Fabre. A framework for geomandric constraint satisfaction problem. In *Proceedings of the ACM SAC2006 Conference*, pages 974–978, 2006.
- [WZZZ01] J. Wu, T. Zhang, X. Zhang, and J. Zhou. A face based mechanism for naming, recording and retrieving topological entities. *Computer Aided Design*, 33(10) :687–698, 2001.

Index

- Acis, 56
- agrégat
- contingent d'entités, 110, 128–132
 - invariant d'entités, 110–113, 127–129, 131, 133
- agrégat d'entités, 106, 113, 125, 126, 128, 130, 131, 133, 134
- ancêtre invariant, 36, 37
- ancêtre invariant, 36, 37, 50, 51, 57, 103, 105, 106, 108, 112–114, 119, 120, 128, 130, 131
- appariement, 30–32, 35–40, 44, 46–52, 55, 105–109, 114, 118–120, 122–125, 131–134
- Brep, 8, 13, 14, 16, 25, 31, 38, 56
- brin, 19, 20, 70, 72–79, 81–88, 90–102
- CAO, 8, 12, 48, 56, 110, 127
- caractéristique, 9, 25–28, 38, 41, 52
- d'assemblage, 26
 - d'usinage, 25, 28
 - de fabrication, 26, 29
 - de forme, 8, 9, 12, 25, 26, 37, 40–42, 44, 49
 - de matériau, 26
 - de tolérance, 26
 - fonctionnelle, 26
- caractéristique d'usinage, 27
- caractéristique déclarative, 41
- caractéristique procédurale, 41
- cartes généralisées, 16, 17, 19, 20, 31, 53, 56, 58, 69, 70, 72, 73, 76, 87, 102, 128
- CasCade, 8, 55
- CATIA, 8, 28, 48, 55
- cellule, 14–17, 19–21, 36
- co-raffinement, 88, 89, 102
- conception par caractéristiques, 26
- coque, 14, 31–33, 48–51, 108, 112, 121, 125–130
- création de caractéristique, 27
- CSG, 8, 13, 14, 23, 25, 56
- éclatement, 17, 79, 80, 84–87, 102
- édition, 15, 30
- entité
- avec source, 34, 36
 - avec source, 34
 - contingente, 33–37, 39, 50, 57, 103, 105, 108, 110
 - invariante, 33–37, 42, 49, 50, 105, 108, 111–113, 127, 133
 - sans source, 34, 36
- espace paramétrique, 38, 42–44, 133
- événement, 53, 56–61, 64, 66, 69, 87, 89, 90, 94, 95, 101, 102
- de création, 53, 57, 58, 60, 68, 69, 102, 113
 - de fusion, 53, 57, 58, 60, 63, 65, 67–69, 99, 102, 113, 128
 - de modification, 53, 57, 68, 69, 102, 113, 128
 - de scission, 53, 57, 58, 60, 61, 63, 65, 67–69, 102, 113, 128
 - de suppression, 53, 57, 58, 60, 67–69, 102, 113, 128
- FAO, 56
- feature, 25
- G-carte, 17–21, 58, 69, 70, 72–87, 89, 90
- graphe, 14, 16, 31, 36, 37, 49, 50, 53, 89, 125–130
- hiérarchique, 38, 48, 49, 51–53, 55, 57, 102, 105, 106, 125–129, 134
- IAO, 56
- journal de bord, 53, 57, 58, 64, 66, 68, 69, 76, 102, 113

- modélisation
 par contraintes, 8
 géométrique, 7–9, 12, 13, 23, 52, 58, 102
 par caractéristiques, 9, 24, 25, 28
 par contraintes, 10
 paramétrique, 9, 12, 33, 42, 52, 53, 56, 57, 61, 67, 105, 110, 125, 128
 équationnelle, 10
 fonctionnelle, 11
- modèle
 géométrique, 12, 14, 16, 26, 27, 34, 50, 52, 53, 56, 58, 66, 102, 110, 114, 120, 128, 129, 131
 initial, 30–32, 37, 39, 43, 46, 47, 50, 60, 105–108, 118–124, 128, 131–133
 paramétrique, 7, 9, 12, 14, 30, 38, 40–42, 49, 62, 63, 66, 114, 127, 128, 131, 133
 équationnel, 10–12
 fonctionnel, 11, 12
 hybride, 12
 réévalué, 30–32, 37, 39, 43, 44, 46, 47, 50, 105, 107, 108, 118–124, 131–133
- Moka, 56, 128
- nomination persistante, 30–33, 36, 38–40, 48–53, 55–57, 101, 102, 105–107, 113, 114, 125, 127, 133
- noyau géométrique, 9, 17, 26, 53, 55–58, 87, 90, 96, 101, 128
- opération
 éclatement, 53, 58, 69, 70, 79, 85, 86
 élémentaire, 58, 61, 63, 67, 88, 89, 93, 95–99
 booléenne, 8, 11, 13, 14, 17, 34, 36, 38, 42, 45, 56, 57, 61, 62, 69, 88, 89
 contraction, 53, 58, 69–73, 76–79, 84–86, 102
 globale, 58–60, 62, 64, 66, 68, 88, 102
 insertion, 53, 58, 69, 70, 79–87, 89–91, 93, 95, 102
 suppression, 53, 58, 64, 69–86, 97, 102
- paramétrique, 9, 13, 15, 22, 25, 33
- plate-forme, 38, 57, 102
- plongement, 12, 16, 17, 20, 21, 34, 44, 56, 97, 99, 114
- Pro/ENGINEER, 8, 28, 55
- protrusion, 32, 49, 127, 130
- quasi-variété, 16, 17, 19, 53, 73, 81, 82, 86
- réévaluation, 11, 25, 30–32, 39, 43, 44, 46–49, 52, 60, 106–108, 118, 120, 124, 127, 131–133
- reconnaissance de caractéristique, 27
- reconnaissance de caractéristiques, 26, 28
- solide, 8, 13–17
- spécification paramétrique, 9, 26, 30–32, 35, 38, 39, 44, 46, 50
- structure, 9, 12, 14, 16, 17, 24, 26, 30–38, 49, 50, 52, 53, 56, 58, 68, 87, 88, 90, 93–95, 98, 99, 101, 102, 106, 109, 110, 112, 113, 116, 118, 121, 125–127, 129, 130, 133
- suppression, 123
- système
 paramétrique, 7, 9, 12, 15, 17, 22, 26, 28, 30–32, 34, 35, 37, 38, 43, 48, 49, 52, 53, 55, 56, 101, 105, 127, 128
 équationnel, 12
 fonctionnel, 12
- tag, 87, 88, 90–102
 actuel, 87, 88, 90, 92–98, 100–102
 ancêtre, 87, 88, 90, 92–96, 98, 100–102
- TopSolid, 28, 48
- voisinage topologique, 31, 36–41, 50, 51, 96, 106, 108, 109, 111, 112, 117, 120, 131, 133

