

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M.Ing.

PAR
BRUNO LAMARCHE

MISE EN OEUVRE DES LIENS TECHNOLOGIQUES COMME OUTIL DE
CAPTURE DE L'INTENTION DE CONCEPTION EN AÉRONAUTIQUE

MONTRÉAL, LE 22 MAI 2007

© droits réservés de Bruno Lamarche

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSE DE :

M. Louis Rivest, ing., Ph.D, directeur de recherche
Département de génie de la production automatisée
École de technologie supérieure

M. Roland Maranzana, ing., Dr., président du jury
Département de génie de la production automatisée
École de technologie supérieure

M. Benoit Côté, ing., M.Sc.A., Conseiller technique senior
Centre de compétences PLM
Bombardier aéronautique, Ville Saint-Laurent

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 3 AVRIL 2007

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

MISE EN OEUVRE DES LIENS TECHNOLOGIQUES COMME OUTIL DE CAPTURE DE L'INTENTION DE CONCEPTION EN AÉRONAUTIQUE

Bruno Lamarche

SOMMAIRE

Les outils CAO modernes sont mis à contribution en développement de produits. Ils permettent d'automatiser des tâches de modélisation fastidieuses qui consomment le temps précieux des concepteurs spécialisés. Les gains de temps ainsi obtenus améliorent le processus de conception en diminuant les cycles de développement, en permettant la propagation automatique des changements d'ingénierie effectués dans un assemblage aux modèles tridimensionnels des pièces impactés et en maintenant la cohérence des données au sein des modèles.

Les concepts de typages d'entités, de liens technologiques établis entre les empreintes et les pseudo-empreintes des objets associés, ainsi que des caractéristiques contextuelles ont démontré qu'ils pouvaient contribuer à structurer efficacement l'exploitation des outils CAO et qu'ils permettaient de capturer une partie de l'intention de conception au sein des modèles. Leur application au cas de soyages et de découpes ont déjà permis des gains de temps substantiels (de l'ordre de 90%).

Tout d'abord, ce travail est un pas de plus dans la généralisation des concepts énumérés précédemment par le développement d'une nouvelle application tirée d'un cas de la réalité industrielle : la modélisation des poches d'allègement contenues dans les panneaux de revêtement des fuselages d'un avion. Cette application exploite le contexte de conception de l'assemblage du panneau pour déterminer la géométrie de la poche à l'aide des éléments de structures (des lisses et des cadres), et du savoir-faire spécifique à ce domaine. Elle insère ensuite automatiquement une nouvelle instance de la caractéristique poche à chaque quadrilatère formé par deux lisses et deux cadres. Cette instance de poche est également liée au contexte à l'aide d'associations permettant la propagation automatique des changements ce qui garantit la cohérence des données au sein des modèles. Les gains de temps obtenus lors de l'utilisation de cette application sont de l'ordre de 98,5%.

En deuxième lieu, nous nous penchons sur l'implication économique et sur la capacité de déploiement à grande échelle des outils de capture du savoir-faire de la CAO moderne (UDFs, programmes VB, etc.) permettant l'automatisation de tâches de modélisation. À défaut de fournir des réponses définitives, cette discussion jette un éclairage nouveau sur le degré d'automatisme souhaitable en CAO appliquée.

IMPLEMENTATION OF TECHNOLOGICAL LINKS AS A TOOL TO CAPTURE DESIGN INTENT IN AEROSPACE ENGINEERING

Bruno Lamarche

ABSTRACT

Modern CAD tools are used in product development to automate tedious and repetitive modeling tasks which consume the invaluable time of specialized designers. The time saved aids the design process by decreasing development cycles time, allowing automatic propagation of engineering changes to the impacted parts of the assembly, and by maintaining data consistency within models.

The concepts of entity typification and technological links, which are established between imprints and pseudo-imprints of associated objects, as well as the concept of contextual features, have all demonstrated that they can contribute in effectively structuring the exploitation of CAD tools and allow a part of the design intent to be captured within models. In the instance of joggles and cut-outs, CAD tool application already exhibited substantial time savings (about 90%).

This paper is foremost a step in the generalization of concepts enumerated previously by the development of a new application for CAD tools drawn from industry: the modeling of lightening pockets found in the skin panel assembly of airplane fuselages. This application uses the design assembly of certain structural elements of the skin panel (stringers and frames), and the know-how of this specific field, to determine the geometry of the pocket. The application then automatically inserts this new pocket feature into each quadrilateral formed by two stringers and two frames. This pocket feature is also contextual linked using persistent associations which allow the automatic propagation of changes that guarantee the coherence of data within the models. The time savings of the use of this application are about 98,5%.

We also consider in this paper the economic implication and the capacity of implementing, on a large scale, the know-how capturing tools of modern CAD (UDFs, VB programs, etc.), allowing for the automation of modeling tasks. Although definitive answers have yet to be found, this discussion sheds new light on the desirable degree of automatism CAD tools can provide.

REMERCIEMENTS

Je voudrais tout d'abord remercier spécialement le Dr. Marc F. Giroux, neurochirurgien à l'Hôpital du Sacré-Coeur de Montréal, sans qui je n'aurais jamais pu accomplir tout ce travail.

Je tiens à remercier mon directeur, M. Louis Rivest, qui a su m'encadrer et me diriger dans la réalisation de ce travail, mais qui m'a aussi supporté dans les épreuves que j'ai vécues en étant disponible, compréhensif et surtout très positif à mon égard.

Je remercie également Bombardier Aéronautique, ainsi que leur personnel, pour m'avoir accueilli à l'été 2003 et pour avoir mis diverses ressources à ma disposition.

Je remercie mes collègues du LIPPS, Omar Msaaf, Thomas Guillaume Tremblay et Michel Michaud, pour leur précieux support et judicieux conseils.

Ce travail fut financé par le programme d'incitation pour diplômés de l'ÉTS à poursuivre des études de 2^e et 3^e cycles à l'ÉTS ainsi que par le CRSNG.

Enfin, je tiens à remercier ma compagne, Vivianne, qui m'a permis de réaliser ce travail grâce à ses encouragements, sa compréhension et son aide au quotidien.

TABLE DES MATIÈRES

	Pages
SOMMAIRE.....	i
ABSTRACT.....	ii
REMERCIEMENTS.....	iii
TABLE DES MATIÈRES.....	iv
LISTE DES TABLEAUX.....	viii
LISTES DES FIGURES.....	ix
LISTE DES ABRÉVIATIONS ET DES SIGLES.....	xiii
INTRODUCTION.....	1
CHAPITRE 1 MISE EN CONTEXTE DU CAS D'APPLICATION.....	5
1.1 Description du contexte.....	5
1.2 Les différents APRs d'un avion.....	6
1.3 Positionnement des éléments de structure.....	10
1.4 Éléments de structures des APRs.....	11
1.5 La tôle de revêtement.....	13
1.5.1 Usinage chimique.....	14
1.5.2 Les poches d'allègement.....	15
1.6 Règles de conception des poches d'allègement.....	19
1.6.1 Méthodologie de conception actuelle des poches d'allègement.....	20
1.7 Gestion des changements et types de liens dans Catia V4.....	26
1.8 Synthèse et objectifs de recherche.....	27
CHAPITRE 2 LA REVUE DE LITTÉRATURE.....	32
2.1 Processus de conception.....	32
2.1.1 L'importance du processus de conception.....	32
2.1.2 Modèles du processus de conception.....	33
2.1.3 Processus de conception en aéronautique.....	36
2.1.4 La propagation des changements d'ingénierie.....	39
2.2 Conception par caractéristiques.....	46
2.2.1 Bref historique de la CAO.....	46

2.2.2	Définition d'une caractéristique.....	47
2.2.2.1	Attributs des caractéristiques	49
2.2.2.2	Types de caractéristiques	50
2.2.2.3	Propriétés des caractéristiques	50
2.3	La représentation du savoir-faire d'ingénierie et de la connaissance	52
2.4	Les UDFs (User-Defined Features)	57
2.5	Synthèse de la revue de littérature	66
CHAPITRE 3 LES CONCEPTS FONDAMENTAUX À VALIDER		70
3.1	Terminologie.....	70
3.1.1	Termes génériques de la CAO	70
3.1.2	Savoir-faire et associations	72
3.2	Temporalité des associations	75
3.2.1	Association persistante.....	76
3.2.2	Association éphémère	76
3.3	Cardinalité des associations	77
3.4	Agrégation et décomposition des associations	78
3.5	Lien de dérivation	80
3.6	Synthèse sur les associations et les liens.....	81
3.7	Le typage des entités	83
3.8	Concept d'empreinte et pseudo-empreinte	83
3.8.1	Concept d'empreinte et pseudo-empreinte dans le cas des poches.....	89
3.8.2	Différents contextes d'application du concept d'empreinte et pseudo- empreinte.....	92
3.9	Synthèse générale.....	94
CHAPITRE 4 PROTOTYPE DE VALIDATION.....		97
4.1	Choix du contexte et objectifs spécifiques du prototype	97
4.2	Terminologie liée au prototype	98
4.3	Granularité des objets considérés par l'application	100
4.4	Règles de conception de la caractéristique « poche » du prototype	101
4.5	Le choix d'une plateforme CAO.....	103
4.6	Quelques différences entre Catia V5 et Catia V4	104
4.7	Stratégie d'exploitation de Catia V5 pour le développement de l'application	106
4.8	Éléments définissant le cadre, la lisse et la tôle de revêtement	107
4.9	Structure et associations d'éléments de l'APR sous Catia V5.....	108
4.10	Utilisation des UDFs (User-Defined Features).....	111
4.11	Représentation d'une instance d'UDF dans l'arbre de spécifications	114
4.12	Méthodologie de conception de l'assemblage type sous Catia V5.....	115
4.13	Différents essais de méthodologies de modélisation de poches	117
4.14	Résumé des conditions préalables et choix méthodologiques retenus.....	121

4.15	Formalisation de la méthodologie de conception des poches d'allègement sous Catia V5	122
4.16	Publications des éléments des UDFs cadre, lisse et tôle.....	126
4.17	Assemblage type et fonctionnement des associations persistantes.....	128
4.18	Création de la caractéristique poche à l'aide d'un UDF	132
4.18.1	Définition de l'UDF poche	132
4.18.2	Test d'instanciation de l'UDF poche	137
4.19	Programmation de l'insertion automatique des poches dans l'APR.....	141
4.20	Synthèse et conclusion relative au développement de l'application.....	147
CHAPITRE 5 ANALYSE DES RÉSULTATS ET DISCUSSION.....		149
5.1	Analyse des résultats obtenus à l'aide du prototype d'application d'insertion automatique des poches d'allègement dans un APR	149
5.1.1	Clarifications sur les temps calculés	149
5.1.2	Adaptation de l'assemblage type	151
5.1.3	Temps de création des UDFs et de l'APR type	152
5.1.4	Comparaisons des temps de création des poches.....	153
5.2	Temps requis pour apporter certaines modification à l'APR.....	156
5.3	Les UDFs et la programmation VB comme outil de mise en œuvre des concepts.....	159
5.4	Capacité de déploiement à l'échelle industrielle d'outils CAO permettant d'automatiser des processus de conception	165
5.4.1	Est-il rentable pour une entreprise de modifier ses méthodologies de conception lorsqu'elle migre d'un outil CAO vers un autre?	166
5.4.2	Est-il rentable pour une entreprise d'investir dans la personnalisation d'outils CAO permettant d'encapsuler et de réutiliser les connaissances ?	167
5.4.2.1	Solutions apportées aux contextes de Giguère et Michaud à l'aide d'UDFs.....	170
5.4.3	Est-il rentable pour une entreprise d'investir dans la programmation d'application orientée métier afin d'automatiser certaines tâches de modélisation ?	173
5.4.4	Existe-t-il d'autres activités de modélisation du domaine aéronautique qui pourraient potentiellement bénéficier de la personnalisation des outils CAO et des applications automatiques ?	175
5.4.5	Peut-on assurer la pérennité des outils CAO de personnalisation et d'automatisation à mesure que la plateforme évolue?	178
5.5	Synthèse de l'analyse des résultats	182
CONCLUSION.....		183
RECOMMANDATIONS		186

BIBLIOGRAPHIE..... 188

LISTE DES TABLEAUX

	Pages
Tableau I	Classification des tôles de revêtement (adapté de Michaud [6]) 18
Tableau II	Règles industrielles de conception pour les poches d'allègement [8]..... 20
Tableau III	Méthodologie de conception des poches d'allègement sous Catia V4 [8]. 21
Tableau IV	Synthèse du concept d'association (adapté de Tremblay [53-55]) 81
Tableau V	Méthodologie de modélisation des poches d'allègement en mode interactif sous Catia V5 123
Tableau VI	Étapes de création de la caractéristique poche à l'aide d'un UDF 133
Tableau VII	Nombre de pièces composant l'assemblage type avant et après l'adaptation 151
Tableau VIII	Temps de création des modèles et des UDFs de l'APR type..... 152
Tableau IX	Comparaison des résultats de Bombardier, Michaud et Lamarche..... 155
Tableau X	Temps requis pour propager certaines modifications dans l'APR type sous Catia V5 154
Tableau XI	Temps requis pour créer et instancier des UDFs des contextes de Giguère et Michaud 172
Tableau XII	Gains apportés par l'application de soyages et de découpe de Giguère [3-5] 174
Tableau XIII	Pérennité de l'application à travers l'évolution de Catia V5 180

LISTES DES FIGURES

	Pages
Figure 1	Exemple d'un assemblage de panneau de revêtement (APR) du fuselage central [9] 6
Figure 2	Exemple de subdivision d'un avion [9] 7
Figure 3	Exemple d'APRs du fuselage avant [9] 8
Figure 4	Exemple d'APRs du fuselage central [9] 9
Figure 5	Exemple d'APRs du fuselage arrière [9] 9
Figure 6	Exemple de "Fuselages Station" pour les cadres 10
Figure 7	Exemple de plans de positionnement des lisses..... 11
Figure 8	Différentes formes d'extrusion utilisées pour les lisses [8]..... 12
Figure 9	Exemple de lisses et de cadres [10] 12
Figure 10	Exemple de cadres usinés 12
Figure 11	Différentes caractéristiques composant un cadre [9] 13
Figure 12	Exemple de tôles dont la profondeur des poches est uniforme [6]..... 16
Figure 13	Tôle à une seule poche dont le contour est complexe [6]..... 16
Figure 14	Exemples de poches des familles 3 et 4 [6] 17
Figure 15	Nomenclature des paramètres de conception des poches d'allègement [8]19
Figure 16	Principaux champs d'expertises impliqués dans le développement d'un avion..... 37
Figure 17	Processus de conception appliqué au domaine aéronautique (adaptée de Jenkinson [7] et de Raymer [22])..... 38
Figure 18	Architecture synthétisée du CASDAT [23] 39

Figure 19	Exemple de procédures de gestion des modifications [24].....	41
Figure 20	Représentation linguistique du concept de porte [40].....	53
Figure 21	Exemple de relations statiques entre les concepts [40].....	54
Figure 22	Exemple de relations dynamiques entre les concepts [40]	54
Figure 23	Cycle de vie de la connaissance [41]	55
Figure 24	Exemple d'UDF: Cylindres de lubrification dans une bielle [45].....	60
Figure 25	Exemple de structure d'une classe de caractéristiques [35;47]	61
Figure 26	Architecture du modeleur de caractéristiques Spiff [35]	61
Figure 27	Structure d'un modèle dynamique faisant varier le nombre de cadres et de lisses [48]	62
Figure 28	Représentation schématique de la procédure d'optimisation [48]	63
Figure 29	Exemple de vues multiples de caractéristiques [49]	64
Figure 30	Associativité entre différentes « vues » d'une pièce [49].....	64
Figure 31	Types de classification des faces F_2 en fonction de F_1 [49].....	65
Figure 32	Quatre cas de cardinalité [53]	77
Figure 33	Exemple d'agrégation du lien rainure-clavette [53]	78
Figure 34	Représentation bas niveau du lien rainure-clavette [53].....	79
Figure 35	Exemple d'agrégation et de décomposition maximale au niveau des associations [53].....	79
Figure 36	Agrégation du lien de dérivation de l'empreinte et de la pseudo- empreinte.....	84
Figure 37	Extraction de l'empreinte dans le cas de la découpe et du soyage (adapté de Giguère [3]).....	85
Figure 38	Création de la pseudo-empreinte et application de la caractéristique cible (adapté de Giguère [3])	86

Figure 39	Relation et liens associant la lisse et le cadre	88
Figure 40	Liens et contraintes associant la lisse et le cadre	89
Figure 41	Exemple de caractéristique contextuelle menant à la définition d'une poche d'allègement.....	90
Figure 42	Lien de dérivation de la poche et sa décomposition maximale en contraintes	92
Figure 43	Granularité considérée par l'application orientée métier des poches d'allègement	101
Figure 44	Paramètres et liens contextuels de la caractéristique poche.....	103
Figure 45	Épaisseur du métal utilisé pour modéliser les lisses et les cadres.....	108
Figure 46	Graphique d'interdépendance des pièces, des caractéristiques, des paramètres et des éléments géométriques impliqués dans l'assemblage.....	109
Figure 47	Structure d'un prototype d'UDF (Users-Defined Features) dans Catia V5	112
Figure 48	Représentation générale d'une instance d'UDF dans l'arbre des spécifications.....	114
Figure 49	Exemple d'APR du fuselage central	115
Figure 50	Arborescence de l'assemblage	116
Figure 51	Arborescence et nomenclature du produit « Cadre en C » et ses UDFs..	117
Figure 52	Choix de quatre congés possibles sous Catia V5.....	119
Figure 53	Plans correspondant au dos et au ventre du cadre.....	120
Figure 54	Plans correspondant aux faces	120
Figure 55	Paramètres et éléments géométriques publiés dans l'UDF cadre.....	126
Figure 56	Paramètres et éléments géométriques publiés dans l'UDF lisse	127
Figure 57	Paramètre publié dans l'UDF de la tôle.....	127

Figure 58	Différents rayons de courbure et angles de la tôle.....	129
Figure 59	Exemple de surfaces à simple et double courbure.....	130
Figure 60	Processus de changement de la surface reliée à la publication « OML ».....	131
Figure 61	APRs basés sur des surfaces à simple et double courbure.....	131
Figure 62	Test d'instanciation de l'UDF à l'intérieur de son modèle de création....	137
Figure 63	Instanciation de la poche dans l'APR type (surface simple courbure)	139
Figure 64	Instanciation des poches dans l'APR type (surface double courbure).....	141
Figure 65	Organigramme général de fonctionnement de l'application	143
Figure 66	Boîte de dialogue et messages guidant l'utilisateur lors de l'exécution de l'application	144
Figure 67	Interface de recherche d'entités dans Catia V5	145
Figure 68	Boîtes de dialogue informant l'utilisateur sur le nombre de poches.....	146
Figure 69	Tôles de revêtement avec les poches d'allègement.....	147
Figure 70	Les deux niveaux d'abstraction de la relation «Contexte d'assemblage - Tôle»	160
Figure 71	Lien de dérivation «Contexte d'assemblage – Poche»	161
Figure 72	L'UDF comme outil de mise en œuvre du lien de dérivation	162
Figure 73	Décomposition du lien «Contexte d'assemblage – Caractéristique Poche» en contraintes	163
Figure 74	Processus complet d'instanciation de l'UDF poche.....	165

LISTE DES ABRÉVIATIONS ET DES SIGLES

ABET	Accreditation Board for Engineering and Technology
API	Application Programming Interface
APR	Assemblage de panneau de revêtement
CAA	Component Application Architecture
CAO	Conception assistée par ordinateur
F.S.	Fuselage Station
IML	Inside Mold lines
LIPPS	Laboratoire d'ingénierie des produits, procédés et systèmes
OML	Outside Mold lines
RADE	Rapid Application Development
UDF	User-Defined Features

INTRODUCTION

Les objectifs globaux des entreprises œuvrant en développement de produits sont généralement de réduire les cycles de conception, de diminuer les coûts et d'augmenter la qualité de leurs produits. Les marchés imposent que le développement de produits se fasse à grande vitesse et les processus qu'adoptent les entreprises pour développer leurs produits sont donc d'une importance capitale dans l'atteinte de ces objectifs.

Afin de supporter et de maîtriser leurs processus de développement, les entreprises utilisent une panoplie de dispositifs informatiques. Parmi eux, nous retrouvons les outils de conception assistée par ordinateurs (CAO). Bien que ceux-ci confèrent plusieurs avantages, comme la réduction du nombre de prototypes physiques nécessaires à la validation d'un concept ou d'un produit, certains aspects liés à leur utilisation peuvent encore être améliorés afin d'en maximiser l'efficacité. Entre autre, lors de la définition du modèle produit, au cours de la phase de conception détaillée, il a été estimé que 80% du temps était consacré à de la définition géométrique [1]. Ce travail routinier consomme le temps d'une main-d'œuvre qualifiée qui pourrait être consacré à des tâches plus créatives et à plus grande valeur ajoutée.

Un autre aspect important et complexe intervenant dans le processus de développement de produits est la gestion des évolutions d'ingénierie. En effet, le processus de conception est itératif et doit par conséquent supporter la gestion et la propagation des évolutions. Lorsqu'une évolution est requise, plusieurs documents sont impliqués et exigent d'être modifiés : modèles de pièces, modèles d'assemblages, gammes d'usinages, liste de composants, etc. De plus, comme les changements sont généralement propagés à la main, il arrive fréquemment que des erreurs surviennent, provoquant des incohérences au sein des données. Lorsqu'elles sont détectées une fois la phase de production amorcée, ces incohérences entraînent des délais et font augmenter les coûts. En effet, il faut soit retravailler les pièces ou les envoyer aux rebuts et en fabriquer de nouvelles.

Dans le domaine aéronautique le produit, tel un avion de transport régional, se compose de plusieurs dizaines de milliers de pièces et ce nombre dépasse 5 millions pour un avion de type Boeing 747 [2]. Chez Bombardier Aéronautique, le cycle de développement moyen pour un programme d'avion est de quelques dizaines de mois et les demandes de changements ont dépassé le nombre de 13 000 en 2001 [2]. Dans ce domaine, comme dans bien d'autres, le modèle produit, construit à l'aide d'outils CAO, est au centre du processus de conception. L'utilisation efficace des outils CAO lors de la création de la définition du produit, et lors de sa modification, contribue à diminuer la durée du cycle de développement.

Depuis quelques années, différents tests réalisés en laboratoire sur certains concepts, tel que le typage d'entités, le lien technologique de dérivation, la caractéristique contextuelle, l'empreinte et la pseudo-empreinte, suggèrent qu'il y a là une avenue à développer afin de supporter les entreprises dans l'atteinte de leurs objectifs. Ces concepts visent essentiellement à structurer le savoir-faire d'ingénierie et la connaissance pour mieux les réutiliser dans certaines tâches de modélisation.

Giguère [3-5] a démontré, à l'aide d'une application CAO exploitant ces concepts afin d'automatiser la définition de soyages et de découpes entre un cadre et une lisse, qu'il était possible d'obtenir des gains de productivité supérieur à 90% au niveau du temps de modélisation géométrique et de l'ordre de 80% sur le temps requis pour propager les changements au sein d'un assemblage tout en maintenant la cohérence des modèles.

Les applications CAO modernes offrent maintenant des outils tel que les liens persistants, les «Users Defined Features» (UDF) et la possibilité d'interroger et de manipuler les modèles tridimensionnels à l'aide de programmes Visual Basic et C++ qui vont dans le sens préconisé par les concepts mentionnés ci-haut.

Bien que les gains obtenus par les applications orientées métier soient substantiels, la faisabilité de leur mise en œuvre à l'échelle industrielle est questionnable entre autre à cause des coûts engendrés par leur développement et leur maintien. En effet, il faut une expertise spécialisée en CAO et en programmation afin de mettre sur pied et entretenir de tels outils. De plus, leur pérennité n'est pas assurée à travers les mises à jour régulières faites sur les plateformes CAO par les éditeurs, ce qui implique parfois la révision des applications.

Ce mémoire vise deux objectifs. Tout d'abord, nous voulons franchir une étape de plus dans la démonstration du caractère générique du concept de typage d'entités et de celui d'associations établies entre ces entités et leurs empreintes et pseudo-empreintes au sein d'une caractéristique contextuelle.

Pour atteindre cet objectif, nous avons tout d'abord identifié, au sein de la réalité industriel, le cas de la définition des poches d'allègement dans la tôle des panneaux de revêtement des fuselages d'un avion qui se distingue de celui des soyages et des découpes utilisé par Giguère [3-5] en 2001. Nous avons ensuite appliqué les concepts mentionnés ci-haut au cas des poches et nous les avons implantés dans un outil de nouvelle génération se matérialisant en une application orientée métier qui automatise, à différents degrés, l'insertion des poches d'allègement au sein de la tôle de revêtement. Cette application exploite le contexte d'assemblage des éléments de structure d'un avion, permettant d'extraire l'information nécessaire à la définition des poches ainsi que le mécanisme des UDFs, permettant d'encapsuler, à l'intérieur même de l'outil CAO, une partie du savoir-faire d'ingénierie propre à cette tâche afin de réutiliser les connaissances.

Le deuxièmement objectif de ce travail est d'alimenter la réflexion sur la valeur en milieu industriel des gains obtenus par des applications développées dans l'environnement contrôlé des laboratoires de recherche. Nous nous penchons sur certains

obstacles techniques et économiques au déploiement à l'échelle industrielle des outils de nouvelle génération.

Pour atteindre ce deuxième objectif, nous avons tout d'abord comparé les gains obtenus lors de l'utilisation d'un outil CAO de nouvelle génération avec et sans adaptation des méthodologies de conception à l'aide du cas des poches d'allègement. Nous avons également repris les contextes de Giguère [3-5] et Michaud [6] afin de mesurer les gains obtenus par l'utilisation d'outils permettant d'encapsuler les connaissances et facilitant l'instanciation. Nous comparons ensuite les gains obtenus par l'utilisation de notre application automatisant l'insertion des poches à ceux obtenus par l'application de Giguère. Finalement, nous avons relevé plusieurs autres situations de modélisation du domaine aéronautique où le potentiel de gains de temps est similaire aux différents cas déjà étudiés.

Le premier chapitre de ce travail est consacré à la description détaillée du contexte du cas d'application tiré de la réalité industrielle. Une revue bibliographique suit afin de situer ce projet par rapport à l'existant. Le troisième chapitre présente les concepts que nous voulons valider à l'aide du cas d'application. Le quatrième chapitre décrit le prototype d'application que nous avons développé et finalement, le dernier chapitre portera sur les résultats obtenus, leur analyse et les conclusions à en tirer en termes de potentiel de déploiement industriel des concepts et des outils.

CHAPITRE 1

MISE EN CONTEXTE DU CAS D'APPLICATION

Ce chapitre décrit le contexte de la conception des poches d'allègement contenues dans la tôle d'un *assemblage de panneau de revêtement* (que nous désignerons par l'acronyme APR) du fuselage d'un avion. Nous débutons par cette description parce qu'elle permet, dans un premier temps, d'introduire le vocabulaire du domaine et, dans un deuxième temps, de situer le contexte par rapport auquel sera menée la revue de littérature.

1.1 Description du contexte

Comme nous l'avons mentionné, un avion de transport régional est composé de plusieurs milliers de pièces. Parmi celles-ci, nous retrouvons, entre autres, des lisses et des cadres. Ces éléments de structures nous intéressent particulièrement, car en plus de constituer le squelette de l'avion, ils servent de référence dans la définition de la géométrie des poches d'allègement contenus dans la tôle de l'APR.

L'une des principales caractéristiques guidant la conception d'un avion est son poids. Plus l'avion sera lourd et moins il sera possible d'y loger des passagers et/ou de la marchandise pour une même distance à parcourir [7]. Une façon de diminuer le poids de l'appareil est de faire varier l'épaisseur de la tôle des APRs. Par exemple, à proximité des éléments de structure, l'épaisseur doit être plus grande, car les contraintes mécaniques subies par l'APR sont plus fortes. En contrepartie, les endroits de l'APR moins sollicités requièrent une épaisseur moindre, comme par exemple à mi-chemin entre les éléments de structure, voir la figure 1, ci-dessous [8].

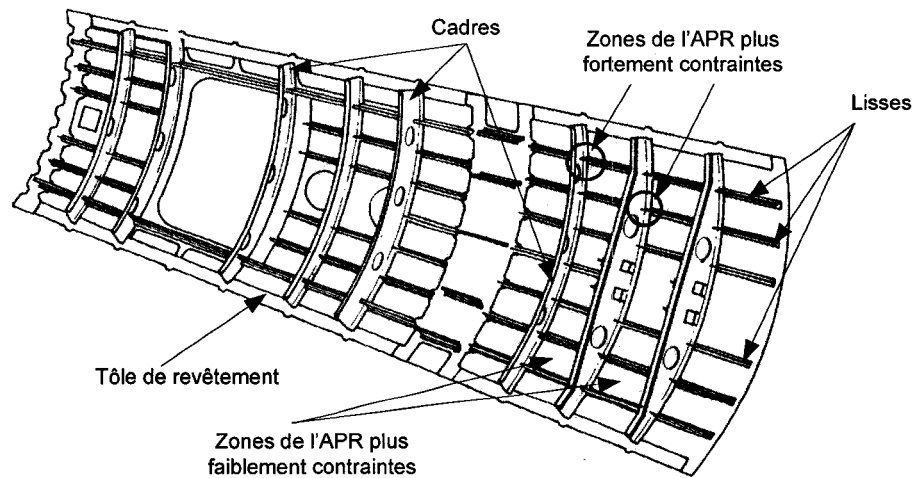


Figure 1 Exemple d'un assemblage de panneau de revêtement (APR) du fuselage central [9]

Des amincissements sont pratiqués dans la tôle (appelés poches d'allègement) dans les zones plus faiblement contraintes. L'usinage chimique est employé afin de « creuser » le métal à l'épaisseur désirée. Pour ce faire, on recouvre complètement la tôle d'un masque dans lequel on pratique des ouvertures, à l'aide d'un gabarit de découpe, pour que l'acide retire le métal et amincisse la tôle aux endroits voulus.

Le travail effectué par les concepteurs pour obtenir la géométrie de chaque poche d'allègement est fastidieux et pourrait être automatisé.

1.2 Les différents APRs d'un avion

L'une des phases du processus de conception en aéronautique consiste à obtenir la forme extérieure de l'avion en termes de surfaces maîtresses, aussi désignées dans le jargon par « OML » pour « Outside Mold Lines ». C'est à partir de ces surfaces que sont élaborés, entre autre, la forme des éléments structuraux de l'avion, les lisses et les cadres, ainsi que celle des tôles de revêtement qui sont assemblées pour donner les assemblages de panneaux de revêtement (APRs).

Un avion peut être subdivisé de plusieurs façons selon le point de vue adopté. Par exemple, le concepteur, le dessinateur d'outillage et l'agent des méthodes n'auront pas du tout la même vue de l'avion et conséquemment, ne le diviseront pas de la même façon. Dans le cas qui nous occupe, nous subdivisons l'avion en trois parties, soit : le fuselage avant, le fuselage central et le fuselage arrière, voir la figure 2.

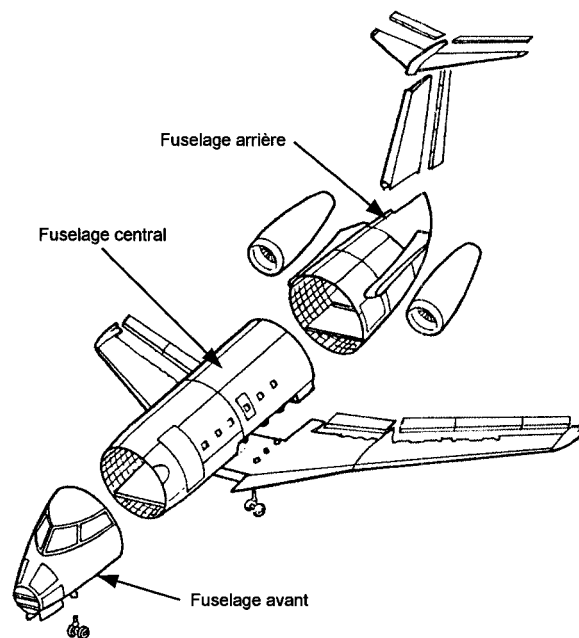


Figure 2 Exemple de subdivision d'un avion [9]

Chaque portion de fuselage est à son tour subdivisée en différents APRs. Le fuselage avant est constitué des APRs latéral droit et gauche, supérieur et inférieur ventral. Ces APRs sont dits à «double courbure» car leur forme épouse une surface variant selon deux axes, voir la figure 3.

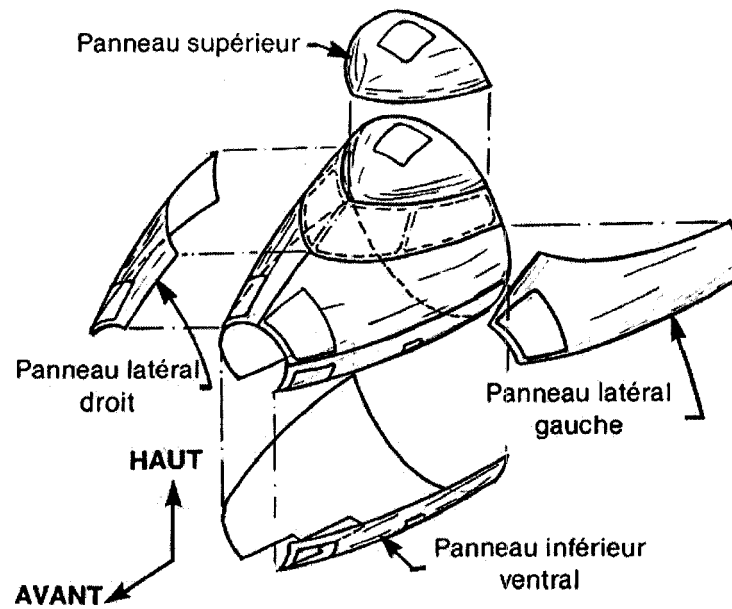


Figure 3 Exemple d'APRs du fuselage avant [9]

Dans l'exemple de l'avion illustrée à la figure 2, le fuselage central comprend douze APRs ; ils sont dits à «simple courbure» car leur forme épouse une surface variant selon un seul axe, voir la figure 4.

Finalement, le fuselage arrière, dans ce cas-ci, se compose de quatre APRs à «double courbure», voir la figure 5.

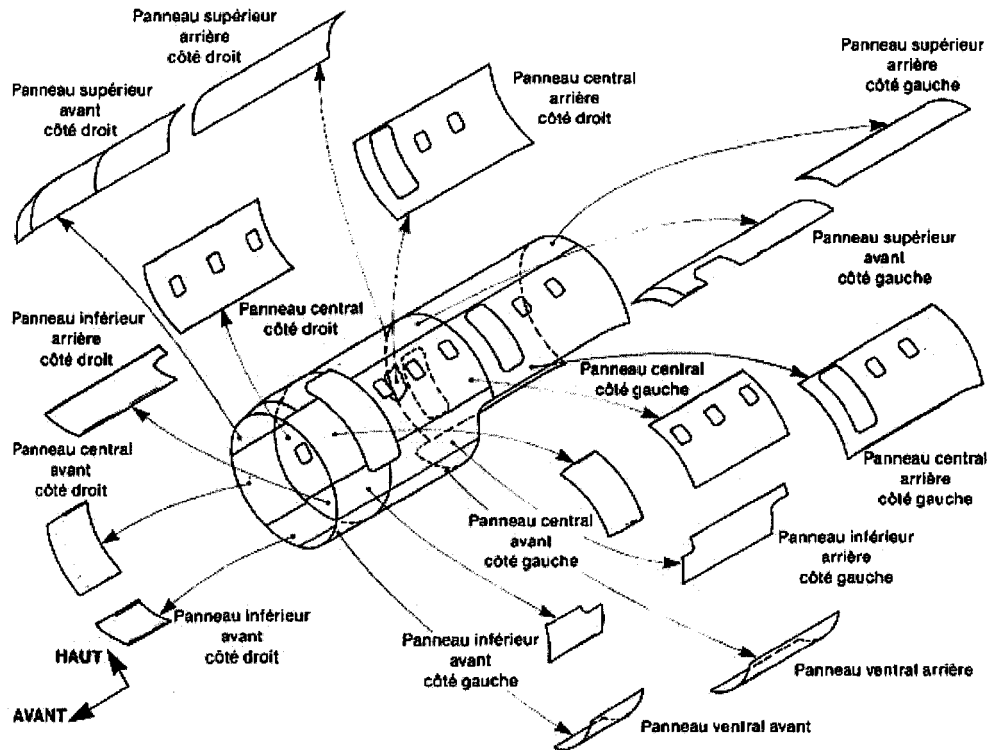


Figure 4 Exemple d'APRs du fuselage central [9]

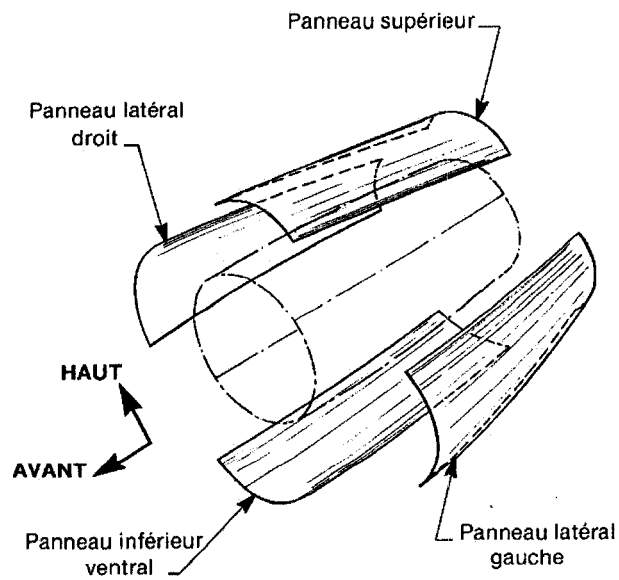


Figure 5 Exemple d'APRs du fuselage arrière [9]

1.3 Positionnement des éléments de structure

Chaque tôle de revêtement est fixée à des éléments de structure. Ceux-ci sont essentiellement de deux types : les lisses (stringers), sont des éléments de structure longitudinaux par rapport au fuselage de l'avion, alors que les cadres (frames), sont des éléments de structure transversaux par rapport au fuselage.

Afin de positionner ces éléments de structures dans la maquette numérique, deux systèmes de plans sont employés. Pour les cadres, les plans sont appelés « Fuselage Station » (F.S.) et sont positionnés du nez à la queue de l'avion à partir d'un système de référence établi à l'avant de celui-ci, voir la figure 6. Pour les lisses, les plans sont distribués radialement par rapport au fuselage de l'avion et sont, par conséquent, fonction d'un angle, voir la figure 7.

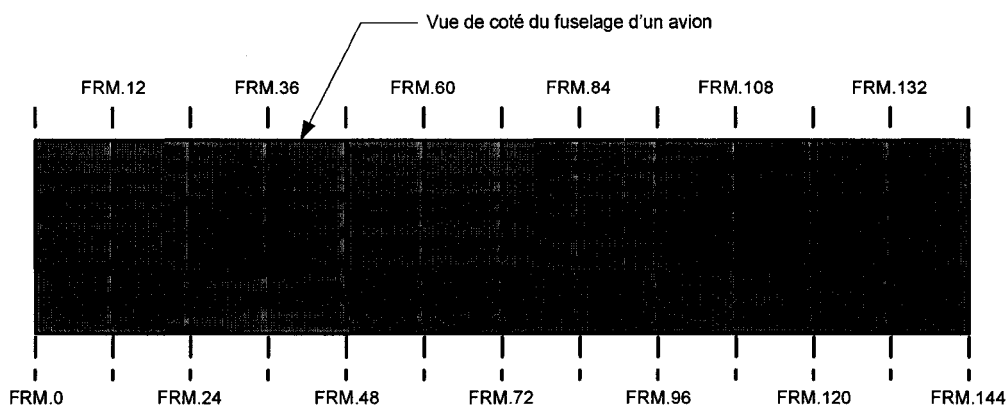


Figure 6 Exemple de "Fuselages Station" pour les cadres

Pour situer facilement ces plans, une nomenclature est adoptée. La forme « Type.Distance » en est un exemple, de telle sorte que « FRM.60 » désigne un cadre situé à 60" du référentiel situé à l'avant de l'avion. De même, pour désigner une lisse positionnée à l'angle 120°, on écrira « STR.120 ».

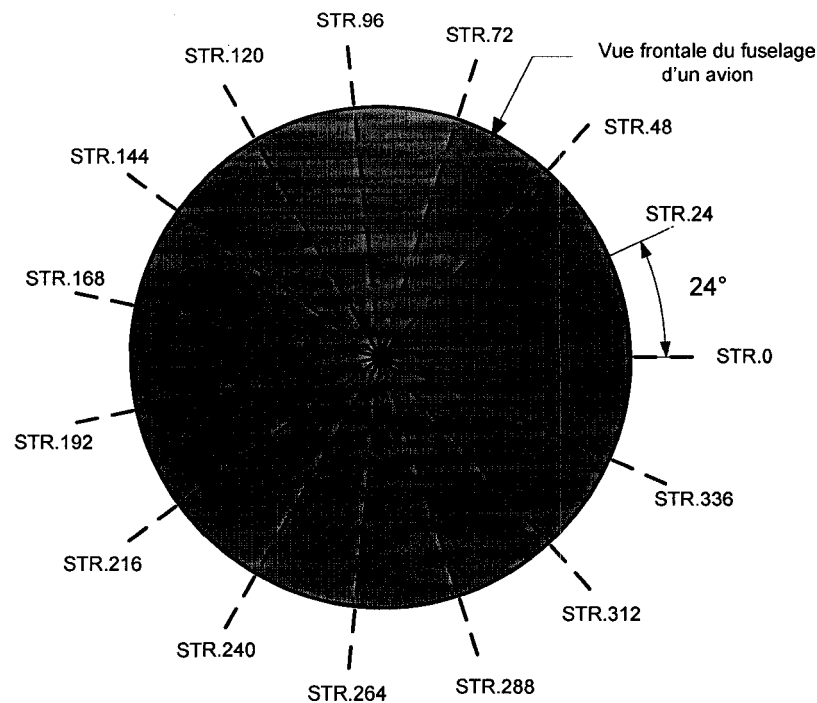


Figure 7 Exemple de plans de positionnement des lisses

1.4 Éléments de structures des APRs

Les éléments de structures qui nous intéressent pour ce travail sont les lisses et les cadres. Les lisses sont des éléments de structures fabriqués selon le procédé d'« extrusion », consistant à faire passer du métal à très haute température au travers d'une matrice ayant la forme du profil désiré [9], voir la figure 8. Ils peuvent également être fabriqués par formage et d'emboutissage.

L'anatomie d'une lisse est très simple et sa forme, sur sa longueur, sera définie selon l'endroit où elle se trouve dans l'avion. Par exemple, dans le fuselage central, elle sera droite, tandis que dans les fuselages avant et arrière elle sera courbée pour épouser la forme de la surface maîtresse interne, « Inside Mold Lines » ou « IML ». Cette dernière est obtenue en décalant l'OML de l'épaisseur de la tôle de revêtement vers l'intérieur du fuselage.

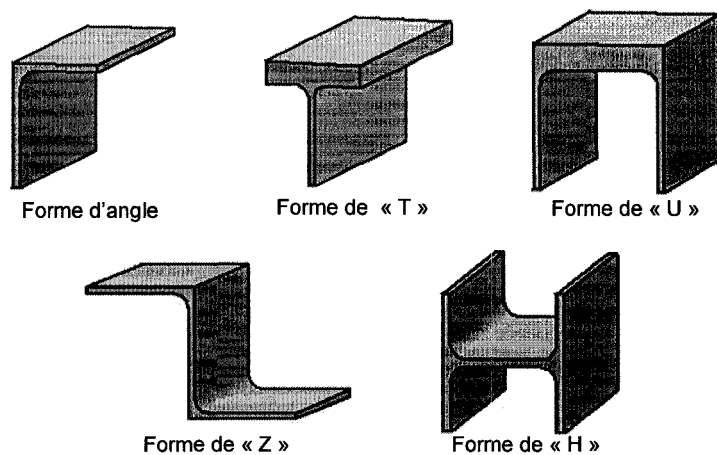


Figure 8 Différentes formes d'extrusion utilisées pour les lisses [8]

Le deuxième élément de structure qui nous intéresse est le cadre. Tout comme la lisse, il fait partie du squelette de l'avion et impose sa forme au fuselage. Les cadres sont soit des pièces usinées ou de métal en feuille et sont généralement en forme de « C », de « Z » ou composé de deux « C » dos à dos, voir la figure 9 et la figure 10 ci-dessous.

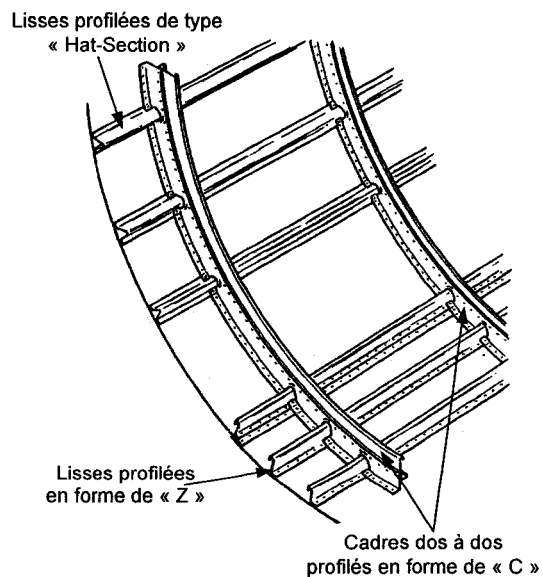


Figure 9 Exemple de lisses et de cadres [10]

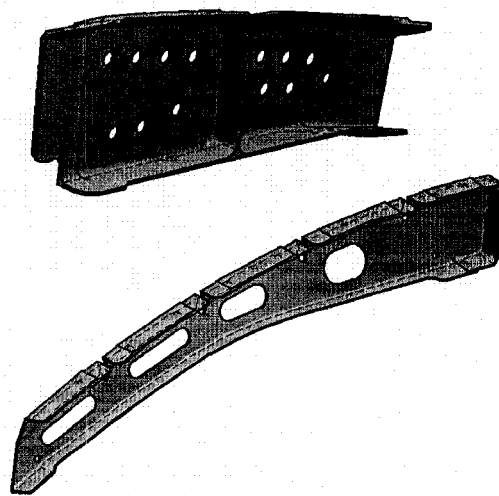


Figure 10 Exemple de cadres usinés

Lors de la préparation de la maquette numérique, la forme du cadre épouse celle de la surface « IML ». Une ouverture, appelée découpe, est pratiquée à l'intersection cadre-lisse. La définition du cadre doit prévoir des soyages sur l'un ou les deux cotés de celui-ci selon la géométrie de la lisse. Ces deux opérations sont dictées par des règles établies par le constructeur. De plus, ces ouvertures nécessitent parfois que l'on rigidifie le cadre à l'aide d'embossages que l'on nomme « beads ». Des trous d'allègement, que l'on nomme « Lightning Holes », peuvent également être pratiqués entre les embossages afin d'alléger le cadre tout en maintenant sa rigidité, voir la figure 11.

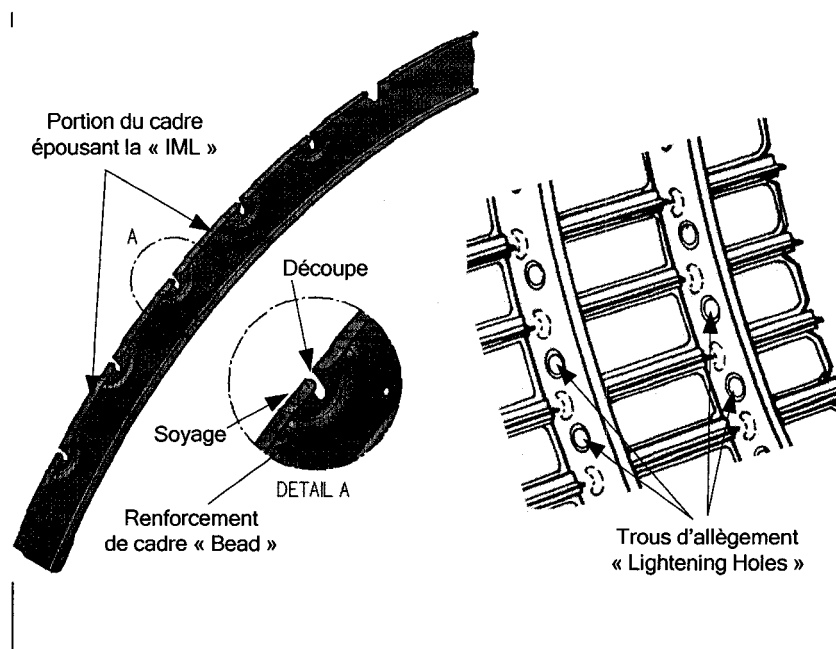


Figure 11 Différentes caractéristiques composant un cadre [9]

1.5 La tôle de revêtement

Le dernier élément qui nous intéresse, et qui est au centre de ce travail, est la tôle de revêtement (Skin). Chez Bombardier Aéronautique, elle est fabriquée d'une feuille de métal en alliage d'aluminium dont l'épaisseur peut varier entre 0.040" et 0.120" [8] et recouvre le squelette de l'avion. Comme nous l'avons mentionné précédemment, cette

épaisseur n'est pas uniforme et varie en fonction des contraintes mécaniques qui sont prévues aux différents endroits de l'APR. Aux intersections cadres-lisses, l'épaisseur nominale de la tôle est conservée à cause des contraintes plus fortes tandis qu'en d'autres endroits, là où les contraintes sont inférieures, des poches d'allègements sont pratiquées afin d'amincir la tôle et de diminuer le poids de l'APR.

À cause de l'épaisseur de la tôle et de sa forme, l'usinage conventionnel des poches d'allègement à l'aide de machines outils est difficile et par conséquent très coûteux [8]. C'est pourquoi la technique d'usinage chimique, abordée à la section suivante, est employée pour « graver » les poches dans la tôle.

1.5.1 Usinage chimique

Le principe de l'usinage chimique, consistant à attaquer un métal à l'aide d'une solution acide ou corrosive, est assez vieux en soi et remonte à 400 ans avant J.-C. À cette époque on utilisait du vinaigre pour graver des décorations faites à l'aide de plomb. C'est autour de 1500 après J.-C. que des solutions d'acide sulfurique et hydrochlorique ont été utilisées pour décorer des armures [11]. Ces techniques sont les précurseurs de l'usinage chimique.

Dans le domaine aéronautique, c'est un ingénieur de la North American Aviaton, Manuel C. Sanz, qui exploite en 1953 la technique d'usinage chimique afin d'alléger la tôle de revêtement d'un missile [11]. Toutefois, ce n'est qu'en 1965 que le Mirage-F1, conçu et fabriqué par Dassault Aviation, devient le premier avion dont les APRs sont allégés par ce procédé [12].

L'usinage chimique des tôles de revêtement se fait par étapes successives. Tout d'abord, on nettoie la pièce à l'aide de solvant afin d'éliminer toute les huiles et les graisses qui pourraient nuire à l'adhérence du masquant. Ce dernier, à base de néoprène, est ensuite

appliqué sur toute la pièce en l'immergeant dans un bassin. On suspend ensuite la pièce afin de l'égoutter et de la faire sécher. Ces étapes sont répétées jusqu'à ce que l'on obtienne une épaisseur uniforme de néoprène de l'ordre de 0.010". Ensuite on découpe, à l'aide d'un couteau à fine lame, les endroits où des poches d'allègement doivent être pratiquées. Cette étape se fait à l'aide d'un gabarit de découpe dont la forme est obtenue grâce à un outil CAO. Les endroits devant être usinés sont mis à découvert. Puis, on trempe la tôle dans une solution hautement corrosive qui attaque le métal mis à nu [8].

Bien que la quantité de matière enlevée soit fonction de la concentration de la solution acide et du temps de trempage, elle se fait typiquement au taux de 0.001" par minute pour l'aluminium et de 0.005" par minute pour l'acier trempé ainsi que pour les alliages d'acier et de titane. Généralement on utilise de l'hydroxyde de sodium comme acide. Une tôle, dont l'épaisseur nominale est de 0.060", sera par exemple trempée dans l'acide jusqu'à ce qu'elle atteigne 0.020" d'épaisseur dans les poches d'allègement [8].

Un séjour effectué chez Bombardier Aéronautique à l'été 2003 nous a permis d'établir que près de 30 000 pièces étaient fabriquées annuellement à l'aide de cette technique [8].

1.5.2 Les poches d'allègement

Les poches d'allègements sont des amincissements pratiqués dans la tôle de revêtement dont le but est d'en diminuer le poids. Michaud [6] classe les tôles en quatre familles distinctes selon l'usage ou leur position dans l'avion et selon le type de poches qui y sont logées.

La première famille est composée des tôles dont les poches sont de profondeur uniforme. Ces tôles sont à simple courbure (fuselage central) et à double courbure (fuselage avant et arrière). Cette famille représente 45% des pièces usinées chimiquement chez Bombardier Aéronautique [8], voir la figure 12.

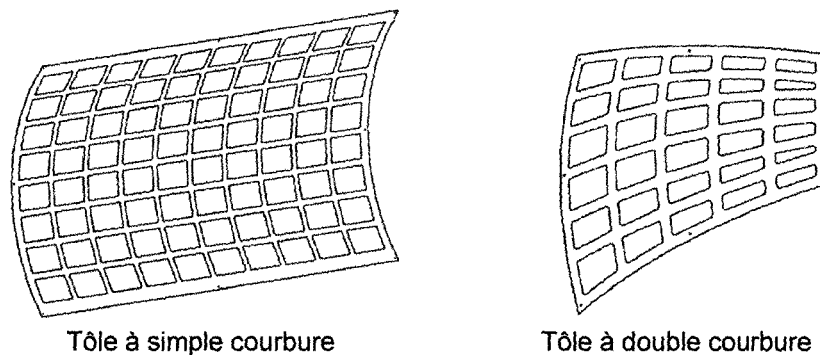


Figure 12 Exemple de tôles dont la profondeur des poches est uniforme [6]

Comme le montre la figure 12, le contour des poches est rectangulaire pour les tôles des APRs du fuselage central et trapézoïdal pour les tôles des APRs des fuselages avant et arrière.

La deuxième famille se compose de pièces comportant peu de poches, mais dont le contour est plus complexe; elle représente 30% du volume de pièces usinées chimiquement, voir la figure 13.

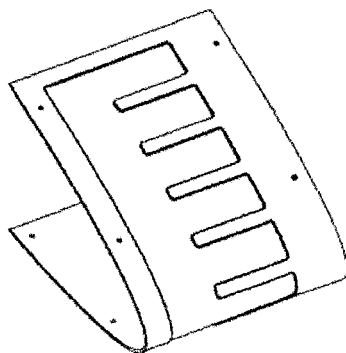


Figure 13 Tôle à une seule poche dont le contour est complexe [6]

La troisième famille regroupe des pièces dont les poches d'allègement sont à profondeurs multiples et représente 20% du volume total. Finalement, la quatrième

famille renferme des pièces dont les poches comportent un îlot et représente seulement 5% du volume total, voir la figure 14, ci-dessous.

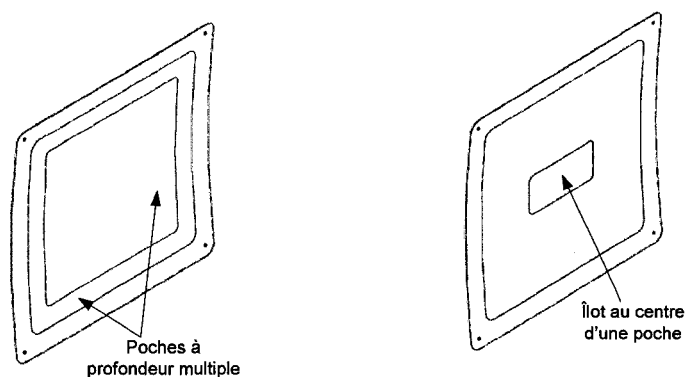


Figure 14 Exemples de poches des familles 3 et 4 [6]

Cette classification permet de distinguer les différents types de poches ainsi que leur importance dans le bassin de production des pièces usinées chimiquement chez Bombardier Aéronautique.

Le tableau I de la page suivante résume les différentes caractéristiques de chacune des familles classifiées par Michaud [6].

Tableau I

Classification des tôles de revêtement (adapté de Michaud [6])

Familles	Volume de tôles	Courbures		Nombre de poches		Profondeur des poches		Profile des poches		Îlots dans les poches
		Simple	Double	Multiple	Peu	Uniforme	Multiple	Simple	Complex	
Première	45%	√	√	√		√		√		
Deuxième	30%	√	√		√	√		√	√	
Troisième	20%	√	√	√			√	√	√	
Quatrième	5%	√	√		√		√	√	√	√

Notre cas d'application s'adresse à la première famille de poches d'allègement, car elle représente le plus grand nombre de pièces dans les quatre familles décrites précédemment. Nous nous pencherons donc sur les poches à profondeur uniforme logées dans les APRs à simple et à double courbure.

1.6 Règles de conception des poches d'allègement

Les règles régissant la conception des poches sont assez simples et varient d'un constructeur à l'autre. Notre séjour chez Bombardier Aéronautique nous a permis de documenter les règles de conception générales pour les poches d'allègement. Toutefois, avant de les exposer, nous utiliserons la nomenclature décrite à la figure 15 afin de situer les différents éléments géométriques constituant la poche.

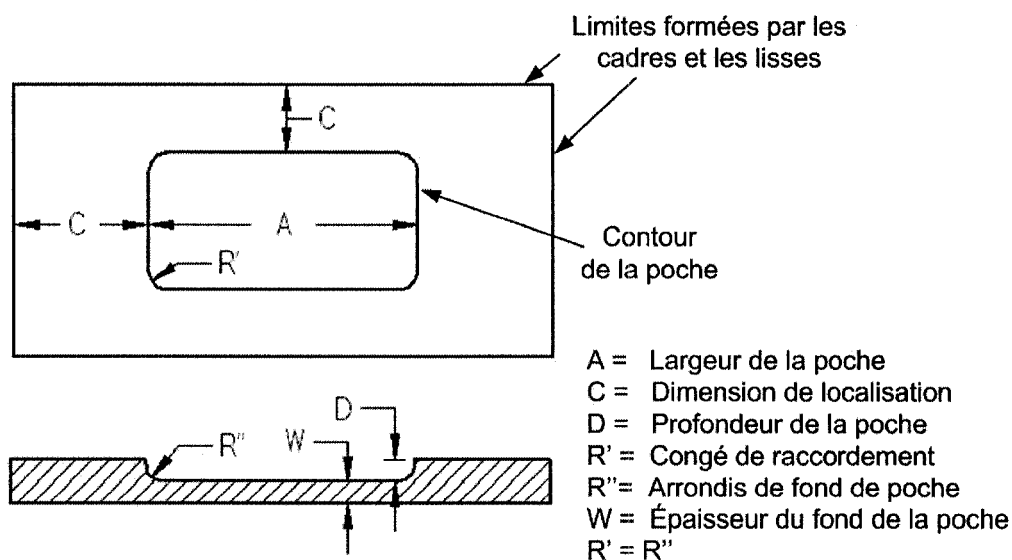


Figure 15 Nomenclature des paramètres de conception des poches d'allègement [8]

Dans la partie supérieure de la figure 15, les lignes formant le rectangle entourant les deux poches représentent les limites formées par les lisses et les cadres. La cote C, « *Dimension de localisation* », est une distance nécessaire entre un élément de structure et le début de la poche. La cote D, « *Profondeur de la poche* », représente l'épaisseur de métal enlevé lors de l'usinage chimique de la tôle. La cote W, « *Épaisseur du fond de la*

poche », correspond à l'épaisseur de métal restant après l'usinage. Les « *Congés de raccordement* » et les « *Arrondis de fond de poche* » sont généralement standardisés à un rayon de 0.500". La cote *A*, « *Largeur de la poche* », est fonction des limites établies par la position des objets environnement.

Le tableau II [8], ci-dessous, résume les règles utilisées par les concepteurs chez Bombardier Aéronautique. Celles-ci fournissent un cadre général aux concepteurs et certains détails sont laissés à leur discrétion, comme par exemple les « *Arrondis de fond de poche* », qui peuvent ne pas être modélisés.

Tableau II

Règles industrielles de conception
pour les poches d'allègement [8]

Règles	Valeurs (pouces)
<i>Dimension de localisation</i>	0.10"
<i>Congés de raccordement</i>	0.50"
<i>Arrondis de fond de poche</i>	0.50"

1.6.1 Méthodologie de conception actuelle des poches d'allègement

Au moment de notre séjour chez Bombardier Aéronautique en 2003, le travail de conception se faisait à l'aide de l'outil CAO Catia V4 afin de construire une maquette numérique tridimensionnelle de l'avion [8]. Chez les industriels qui utilisent ce type d'outil, des méthodologies de conception sont mises sur pied par des spécialistes dans le but de standardiser la façon dont les pièces sont conçues, structurées et positionnées dans chaque modèle numérique. Il en va de même pour la création des poches d'allègement des APRs.

Le processus de conception des poches d'allègement laisse le concepteur effectuer lui-même toutes les opérations de bas niveau afin d'obtenir un modèle numérique représentant l'état final de la tôle, i.e. avec ses poches d'allègement.

Le principe consiste à créer une extrusion, dont le contour correspond à la géométrie de la poche, pour être soustraite ensuite au volume représentant la tôle. Nous avons décortiqué le processus en quatorze étapes énumérées et illustrées ci-dessous :

Tableau III

Méthodologie de conception des poches d'allègement sous Catia V4 [8]

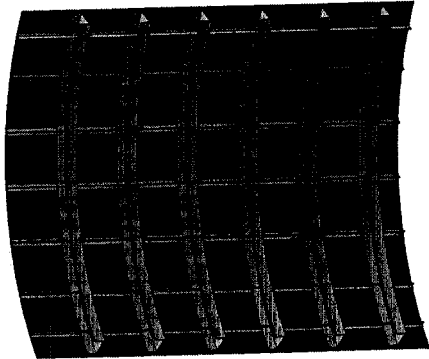
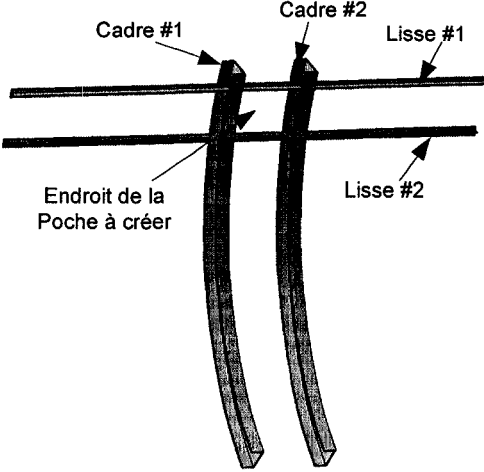
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #1 :</p> <p>Le concepteur charge d'abord l'APR dans la mémoire de l'ordinateur. Il contient les lisses, les cadres et la tôle de revêtement.</p>	
<p>Étape #2 :</p> <p>Il identifie les éléments de structures encadrant la poche à créer, soit une paire de lisses et une paire de cadres.</p>	

Tableau III (suite)

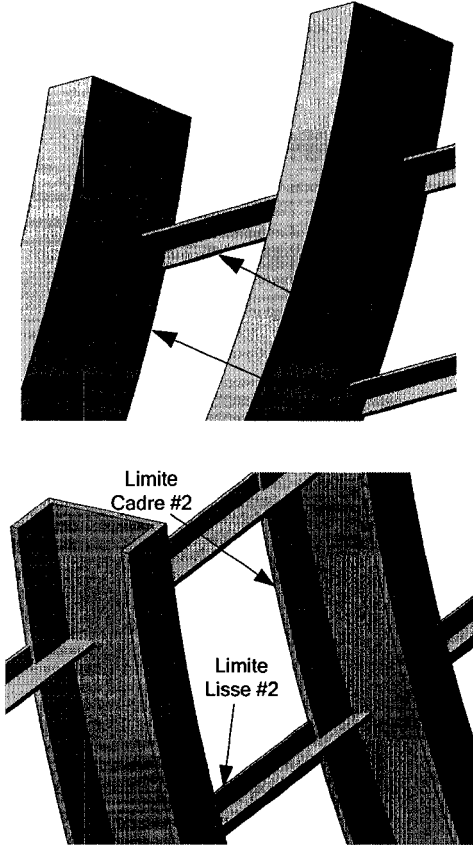
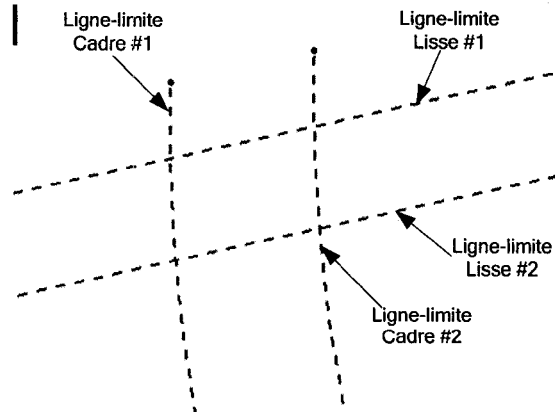
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #3 :</p> <p>Il extrait les lignes aux limites des lisses et des cadres reposant sur la surface de la tôle de revêtement du côté où la poche doit se trouver. C'est à partir de ces lignes limites que la géométrie de la poche sera créée.</p>	
<p>Étape #4 :</p> <p>Il transfère ensuite les lignes-limites dans le modèle de la tôle de revêtement.</p>	

Tableau III (suite)

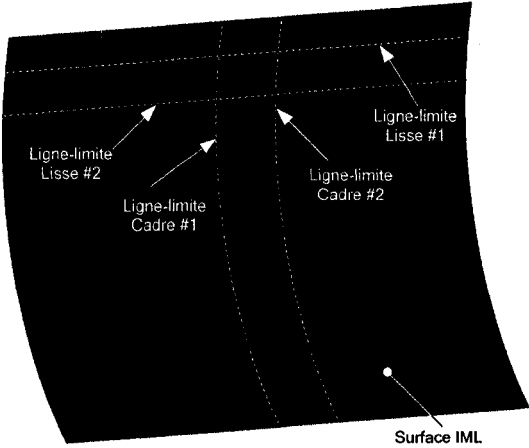
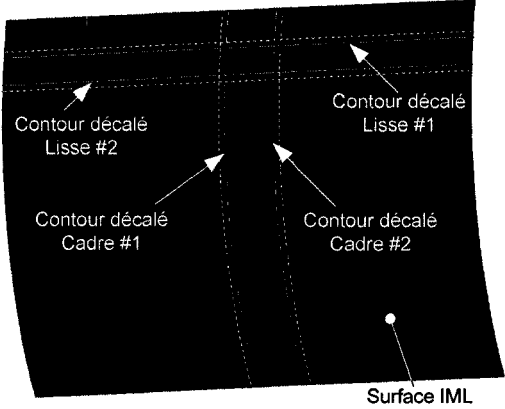
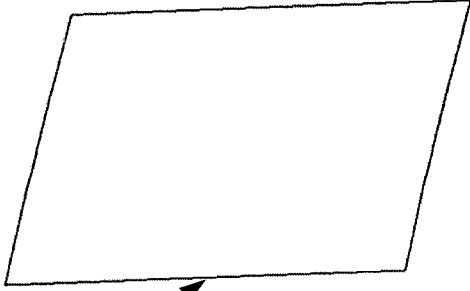
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #5 :</p> <p>Il projette ensuite les lignes-limites sur la surface IML « Inside Mold Line » présente dans le modèle de la tôle.</p> <p><i>Note. On suppose ici que la surface IML est déjà dans le modèle de la tôle.</i></p>	 <p>Le diagramme illustre une surface IML (Inside Mold Line) représentée par une forme quadrilatère déformée. Trois lignes-limites sont projetées sur cette surface : une ligne-limité Lisse #1 (à droite), une ligne-limité Cadre #1 (au centre) et une ligne-limité Lisse #2 (à gauche). Une légende à droite indique 'Surface IML' avec un point blanc.</p>
<p>Étape #6 :</p> <p>Il effectue ensuite un décalage « géodésique » des lignes-limites de 0.100", la « Dimension de localisation » standard, sur la surface IML afin de construire la géométrie du contour de la poche.</p> <p><i>Note : L'échelle n'est pas respectée ici afin de bien illustrer la méthode.</i></p>	 <p>Le diagramme illustre le décalage géodésique des lignes-limites. Les nouvelles lignes sont étiquetées comme 'Contour décalé Lisse #2', 'Contour décalé Cadre #1' et 'Contour décalé Lisse #1'. Une légende à droite indique 'Surface IML' avec un point blanc.</p>
<p>Étape #7 :</p> <p>Il procède ensuite à une « relimitation » des lignes-limites afin d'obtenir la géométrie du contour de la poche.</p> <p><i>Note : Le contour de la poche donne l'illusion d'être trapézoïdal à cause de la vue qui n'est pas parfaitement face à l'APR.</i></p>	 <p>Le diagramme illustre la géométrie rectangulaire du contour de la poche, montrant un rectangle déformé. Une légende en bas indique 'Géométrie rectangulaire du contour de la poche' avec une flèche pointant vers le contour.</p>

Tableau III (suite)

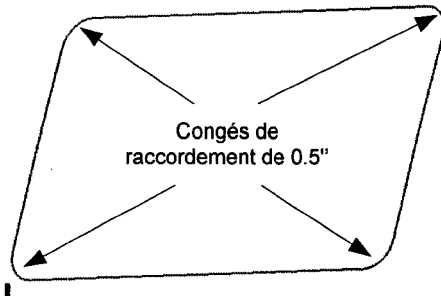
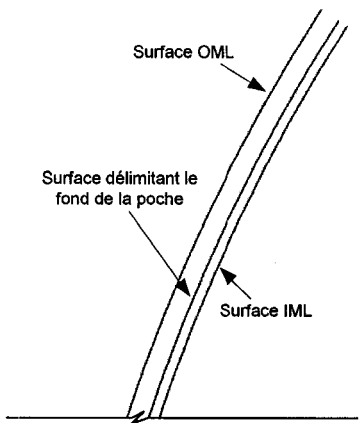
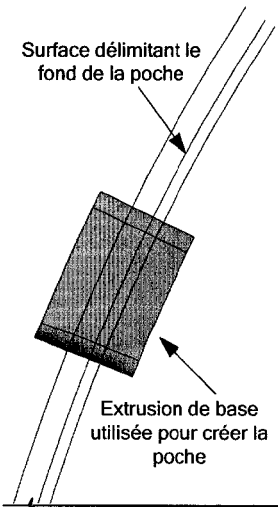
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #8 :</p> <p>Il définit ensuite les congés de raccordement aux quatre coins du rectangle de la poche en utilisant le paramètre de « <i>Congé de raccordement</i> » standard de 0.500".</p>	
<p>Étape #9 :</p> <p>Le concepteur utilise la surface IML pour créer une autre surface délimitant le fond de la poche. Elle est décalée de l'épaisseur du matériel enlevé lors de l'usinage chimique.</p>	 <p style="text-align: center;">Vue de profil des surfaces</p>
<p>Étape #10 :</p> <p>À l'aide du contour défini précédemment, le concepteur crée une extrusion dépassant la surface IML d'un côté et la surface OML de l'autre côté. Celle-ci est définie selon une droite passant par le centre du cylindre représentant le fuselage.</p>	 <p style="text-align: center;">Vue de profil des surfaces et de l'extrusion</p>

Tableau III (suite)

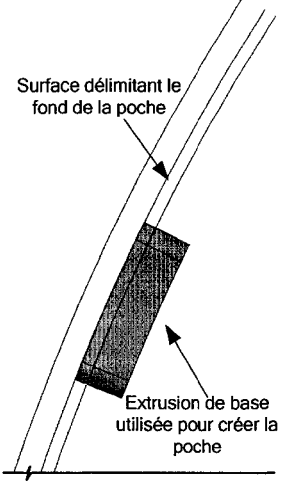
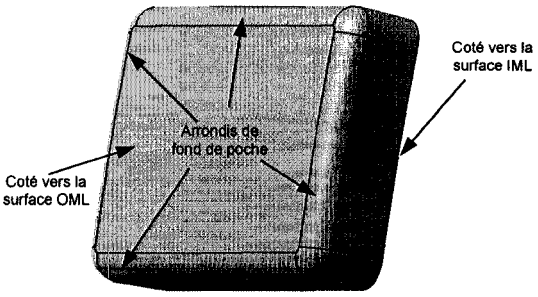
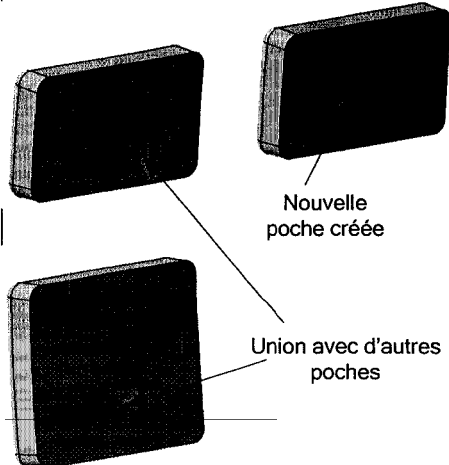
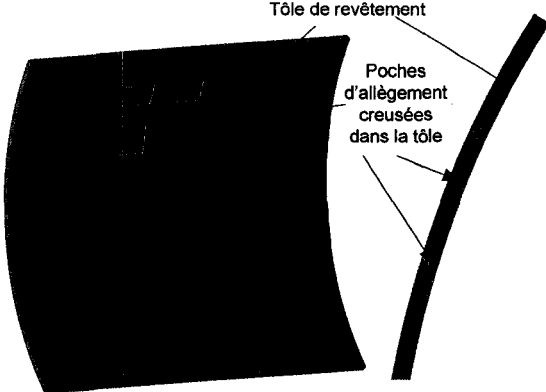
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #11 :</p> <p>Le concepteur délimite maintenant l'extrusion créé précédemment à l'aide de la surface délimitant le fond de la poche.</p>	
<p>Étape #12 :</p> <p>Il crée ensuite les arrondis sur le contour du bloc du côté de la surface OML en utilisant le paramètre d'« Arrondis de fond de poche » standard de 0.500".</p>	
<p>Étape #13 :</p> <p>Une union aux autres poches d'allègement de la tôle est ensuite effectuée.</p>	

Tableau III (suite)

Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #14 :</p> <p>Finallement, le concepteur soustrait toutes les extrusions de la tôle de revêtement. On obtient donc une maquette numérique de la tôle de revêtement à son état final.</p>	 <p>Tôle de revêtement</p> <p>Poches d'allègement creusées dans la tôle</p>

Ces étapes nous permettent de bien comprendre les opérations que doit effectuer le concepteur sous Catia V4 afin d'obtenir la tôle de revêtement avec ses poches d'allègement. Pour chacune d'elles, il doit répéter ces quatorze étapes composées d'opérations de bas niveaux, répétitives et fastidieuses qui pourraient être déléguées à la machine par une application les automatisant.

1.7 Gestion des changements et types de liens dans Catia V4

Catia V4 fonctionne sur une base de liens « éphémères » (voir la définition dans le chapitre 3 traitant des concepts). En effet, dans de nombreux cas, lorsqu'un élément est créé, il est presque automatiquement orphelin. Prenons l'exemple illustré à la section précédente où les lignes-limites extraites des lisses et des cadres sont utilisées pour définir la géométrie des poches. Dès que ces lignes sont créées, elles sont isolées de l'entité dont elles sont issues. Le logiciel ne garde aucune trace de l'historique de dépendance entre les entités de telle sorte que l'on pourrait effacer la lisse de laquelle une ligne-limite fut extraite sans affecter cette ligne.

Une des conséquences de cet état de fait est que dans bien des cas le coût associé à une modification de la géométrie dans un modèle Catia V4 est du même ordre que celui de la création initiale de celle-ci. Prenons à nouveau l'exemple des poches d'allègement. En l'absence de liens persistants, un changement, aussi simple qu'une modification de la position d'un cadre, oblige le concepteur à refaire plusieurs des étapes vues précédemment ou de modifier le contour des prismes des poches, ce qui est loin d'être trivial sous Catia V4. D'ailleurs, chez Bombardier Aéronautique, les concepteurs optent fréquemment pour refaire complètement les éléments impactés par le(s) changement(s). Cette option est globalement moins couteuse et les risques d'erreurs et d'introduction d'incohérences dans le modèle sont également moins élevés dans ces circonstances [8].

La révolution informatique que nous connaissons depuis les dix dernières années a contribué à l'émergence de nouveaux outils CAO très puissants tel que Catia V5 basés sur des liens persistants. C'est en partie grâce à la puissance de calcul des ordinateurs d'aujourd'hui qu'il est possible à Catia V5 d'emmagasiner et de gérer une grande quantité de liens.

1.8 Synthèse et objectifs de recherche

Les plateformes CAO évoluent et mûrissent constamment. Elles fournissent de plus en plus d'outils permettant d'accélérer la modélisation et d'enrichir les modèles d'information permettant de mieux manipuler la géométrie. Parmi ces outils de nouvelle génération, notons, entre autre, les liens persistants, les UDFs (User-Defined Features) et l'automatisation de processus de modélisation à l'aide de programmes Visual Basic ou C++.

Les concepts tels que le typage d'entités, l'empreinte et la pseudo-empreinte, les liens technologiques et les caractéristiques sont de mieux en mieux maîtrisés et s'intègrent plus naturellement aux plateformes CAO. Leur exploitation devient plus facile et promet des gains majeurs.

Toutefois, d'importantes questions subsistent quant au potentiel de déploiement à l'échelle industrielle de ces concepts et outils, à la complexité de la personnalisation et aux coûts qu'elle implique, mais aussi au choix à faire entre personnaliser des processus de modélisation, i.e. utiliser des UDFs, par exemple, et/ou automatiser complètement certains processus de modélisation, i.e. recourir à des programmes codés en VB ou en C++. La pérennité de l'effort requis par la personnalisation, ainsi que le niveau de maîtrise du processus de conception et des outils par les concepteurs sont également des aspects importants à considérer, car ils influencent directement la rentabilité de la personnalisation ou de l'automatisation.

Ce travail vise deux objectifs : le premier consiste à contribuer à valider le caractère générique des concepts de typage d'entités, de liens technologiques, d'empreinte et de pseudo-empreintes en les appliquant à un nouveau cas. Le second objectif consiste à analyser et à statuer sur le potentiel de déploiement industriel de la personnalisation et/ou de l'automatisation des processus de modélisation.

Giguère [3-5] a démontré que l'utilisation des concepts mentionnés ci-haut permettaient d'obtenir des gains en réduisant d'environ de 90% le temps de modélisation en les appliquant au cas de soyages et de découpes dans un cadre à l'intersection d'une lisse. Son cas d'application utilisait une pièce de référence et s'appliquait à une pièce cible. Nous verrons plus loin que la cardinalité du lien technologique qu'il a utilisé est de 1X1 (au niveau d'abstraction des pièces).

Le cas d'application sur lequel nous nous penchons dans ce travail est un peu plus complexe que celui utilisé par Giguère [3-5]. Comme nous venons de le voir, la modélisation d'une poche d'allègement exploite un quadrilatère formé par deux cadres et deux lisses afin d'en déterminer la géométrie. Elle pourra ensuite être usinée chimiquement dans la tôle de revêtement. L'application d'insertion automatique des poches dans les APRs que nous avons développée utilise un lien technologique reposant

sur quatre empreintes pour chaque poche ce qui implique une cardinalité de 4X1. En appliquant les concepts à ce deuxième cas, dont la complexité est plus élevée, on vise à démontrer leur potentiel de généralisation.

Le développement du prototype d'application nécessite plusieurs étapes et la méthodologie suivante est adoptée :

- maîtrise et compréhension des règles de conception des poches d'allègement chez Bombardier Aéronautique ;
- maîtrise des outils de nouvelle génération de la plateforme CAO utilisée ;
- développement d'une méthodologie de modélisation exploitant les outils de la plateforme CAO ;
- maîtrise et application des concepts de typage d'entités, d'empreinte et de pseudo-empreinte, de liens technologiques et de caractéristiques au contexte de modélisation des poches d'allègement ;
- développement d'une méthode et d'une application automatisant le processus de modélisation des poches d'allègement.

Le deuxième point que nous abordons dans ce travail consiste à alimenter la réflexion sur la capacité et la faisabilité de déployer les outils disponibles et les méthodes développées à une échelle industrielle. Plus précisément, les aspects suivants seront abordés et discutés :

- L'adaptation des méthodologies de conception aux nouveaux outils implantés en industrie est-elle nécessaire? Nous verrons à l'aide de données chiffrées que ce processus est inévitable afin de bénéficier pleinement du potentiel de gains de ces outils.

- Est-il rentable pour une entreprise d'investir dans la personnalisation d'outils CAO permettant d'encapsuler et de réutiliser les connaissances ? Nous verrons que le temps consacré à personnaliser ces outils, tel que les UDFs, est largement récupéré et ce de plusieurs façons.
- Est-il rentable pour une entreprise d'investir dans la programmation d'application orientée métier afin d'automatiser certaines tâches de modélisation ? Nous verrons à l'aide des résultats obtenus lors du développement et de l'utilisation de notre application de définition de poches que les gains sont substantiels.
- Existe-t-il d'autres activités de modélisation du domaine aéronautique qui pourraient potentiellement bénéficier de la personnalisation des outils CAO et des applications automatiques ? Nous dressons un inventaire des cas d'application où la « modélisation automatique » pourrait s'appliquer.
- Peut-on assurer la pérennité des outils CAO de personnalisation et d'automatisation à mesure que la plateforme évolue ? Nous voulons vérifier que l'effort de personnalisation et d'automatisation de certaines tâches de modélisation est conservé lors des différentes mises à jour appliquées à l'outil CAO (dans notre cas Catia V5).

Jusqu'à maintenant, nous avons établi la problématique et décrit le contexte du cas d'application que nous utiliserons. Le chapitre 2 est consacré à la revue de la littérature dans laquelle nous situons nos travaux par rapport à ce qui existe déjà. Le chapitre 3 est dédié à l'établissement de la terminologie que nous utiliserons tout au long de ce travail et à la définition de différents concepts que nous employons. Le chapitre 4 est dédié à la description du développement du prototype d'application permettant de valider les concepts. Dans le chapitre 5, nous nous penchons d'abord sur les résultats obtenus à

l'aide du prototype. Par la suite, nous nous basons sur ce travail pour discuter du potentiel de déploiement à l'échelle industrielle des outils CAO de personnalisation et d'automatisation de processus de modélisation. Finalement, nous terminons ce travail par une conclusion et des recommandations pour le futur.

CHAPITRE 2

LA REVUE DE LITTÉRATURE

Ce chapitre est consacré à l'état de la recherche dans les différents domaines reliés à ce travail. Il permet, entre autre, de situer le projet actuel par rapport à l'ensemble existant. Les sujets suivants sont abordés: le processus de conception, la conception par caractéristiques, la capture du savoir-faire d'ingénierie et de la connaissance, ainsi que les UDFs (User-Defined Features).

2.1 Processus de conception

2.1.1 L'importance du processus de conception

Les entreprises tendent à développer de nouveaux produits et à les lancer de plus en plus rapidement sur le marché pour assurer leur survie. Les étapes permettant de passer de la définition initiale d'un produit à sa fabrication sont la pierre angulaire d'une entreprise [13].

Le développement des nouvelles technologies a fait en sorte que les produits fabriqués par les entreprises sont de plus en plus complexes. Ce phénomène affecte directement la façon dont ils sont conçus et nécessite une approche multidisciplinaire. Cette augmentation du nombre de spécialistes au sein de l'équipe de développement exige une structure organisationnelle efficace afin qu'aucun aspect ne soit oublié lors de la conception. La complexification des produits se répercute donc sur le processus de conception [13;14].

Il a été estimé que 85% des nouveaux produits qui ne fonctionnaient pas comme ils le devaient, prenaient trop de temps à atteindre le marché ou coûtaient trop cher étaient le résultat d'un mauvais processus de conception [13].

Par conséquent, il est impératif que les entreprises maîtrisent les différents outils et techniques permettant d'établir, de suivre et d'améliorer leur processus de conception de produits pour maintenir leur compétitivité et assurer leur place sur le marché.

2.1.2 Modèles du processus de conception

Le processus de conception est la période de gestation d'un nouveau produit de sa définition à sa production. Selon Ullman [13], ce processus consiste à l'organisation et la gestion des gens et de l'information développée au cours de l'évolution d'un produit.

L'un des premiers et des plus vieux modèle du processus de conception se nomme «*Over-the-Wall*». Linéaire et sans rétroaction, il consiste simplement à «envoyer» l'information d'une étape «par-dessus » le mur à l'étape suivante [13].

Au début des années 80, le modèle du processus de conception d'ingénierie simultanée fait son apparition et propose une méthode basée sur le parallélisme des activités. Ce modèle permet d'assurer la communication et la coopération entre les différentes équipes de travail au lieu de transférer un lot de données hermétiques à l'équipe suivante.

Cette méthode procure des avantages tel que la diminution des délais, une meilleur réactivité aux changements, etc., mais exige, en contrepartie, une gestion accrue des différents couloirs de communication (brouillons, dessins détaillés, concepts, notes de réunions, etc.). Le processus doit donc être établi de façon à s'assurer que la bonne information soit transmise à la bonne personne au bon moment [13].

Hyman [14] considère le processus de conception comme une approche méthodique de résolution de classes particulières de problèmes complexes. Sa définition est basée sur un modèle d'application du processus de conception divisé en neuf étapes:

1. la reconnaissance des besoins ;
2. la définition du problème ;
3. la planification du projet ;
4. l'acquisition d'informations ;
5. l'élaboration d'approches alternatives ;
6. l'évaluation des alternatives ;
7. la sélection de l'alternative préférée ;
8. la communication de la conception finale ;
9. l'implantation de la conception finale.

Il adhère également à la définition du «*Accreditation Board for Engineering and Technology*»¹, stipulant que le processus de conception divise un système, un composant ou un processus afin qu'il remplisse les exigences émises au départ. C'est un processus décisionnel, souvent itératif, dans lequel on applique les sciences de base, les mathématiques et la technologie à la transformation efficace des ressources disponibles dans le but d'atteindre les objectifs fixés au départ du processus.

Selon Suh [15], le processus de conception est la création de solutions synthétisées sous formes de produits, de logiciels, de processus ou de systèmes par une sélection appropriée des paramètres de conception satisfaisant les besoins perçus à travers une correspondance entre les exigences fonctionnelles (dans le domaine fonctionnel) et les paramètres de conception (dans le domaine structurel). Cette correspondance est basée sur le savoir-faire de conception.

Finger [16;17] identifie trois grandes familles de modèles de base du processus de conception dans sa revue sur les recherches en conception mécanique, effectuée en 1989. Bien qu'elle ne soit pas récente, cette revue demeure actuelle et pertinente comme le démontre Reymen [18;19] dans ses travaux sur l'utilisation du modèle descriptif de

¹ ABET, organisme accréditant les programmes d'études en ingénierie aux Etats-Unis.

conception afin de développer un modèle prescriptif supportant la réflexion pendant le processus de conception. Il a étudié l'applicabilité de son modèle de façon générale, dans les domaines de l'architecture, de l'ingénierie logiciel et de l'ingénierie mécanique.

La première famille identifiée par Finger est le modèle dit descriptif et s'adresse aux processus cognitifs à la base de la conception, i.e. à la façon dont les concepteurs réfléchissent sur un problème et aux processus mentaux leur permettant d'élaborer des solutions. Plusieurs écoles basées sur différentes philosophies de conception ont été formées, notons, entre autre, l'approche proposée par Warfield [20] nommée «*The General Design Theory*», ou encore celle de Fitzhorn [21] nommée «*The Turning Machine*».

La deuxième famille de modèles est dite prescriptive; elle est plus formelle et plus structurée que la première. Selon ces modèles, le processus de conception est itératif et devrait se dérouler selon les six étapes génériques suivantes :

1. l'établissement des besoins ;
2. la spécification des exigences ;
3. la formulation des concepts ;
4. la sélection des concepts ;
5. l'intégration des détails ;
6. la production et la vente.

Notons les modèles suivants : «*The canonical design process*», «*The morphological analysis*» et «*The prescriptive models of the design artifact* » [16;17].

La troisième famille de modèles est basée sur les ordinateurs (Computer-Based Models). Ces modèles utilisent des outils informatiques pour accomplir différentes tâches. On distingue essentiellement deux types de processus informatiques : ceux qui

accomplissent des tâches de conception et ceux qui fournissent l'information nécessaire sur lesquelles des évaluations conceptuelles et des décisions sont prises. Notons, entre autre, dans cette famille, les modèles de conception paramétrique et celui de conception structurée. Notre projet fait partie de cette famille et se situe dans la catégorie des outils informatiques qui accomplissent des tâches de modélisation.

2.1.3 Processus de conception en aéronautique

Le cas d'application que nous utiliserons afin de valider les concepts présentés dans ce travail est tiré du domaine aéronautique. Nous abordons donc, dans cette section, le processus de conception de produit adopté en général dans le domaine aéronautique.

L'une des principales caractéristique des aéronefs est leur complexité. Tout d'abord, au niveau du nombre de pièces qui les composent. En effet, un avion de transport de type régional peut compter plusieurs dizaines de milliers de pièces différentes, tandis qu'un avion de type «Boeing 747» en compte plus de 1 million [7;13]. En deuxième lieu, la production d'un avion fait intervenir de multiples champs d'expertises spécialisées.

Comme l'illustre la figure 16, chacune des spécialités impliquées dans le développement d'un aéronef possède sa propre vision de l'avion idéal. De manière caricaturale, le groupe responsable du poids voudrait le minimiser, tandis que le groupe de propulsion voudrait doter l'avion du moteur le plus puissant qui soit.

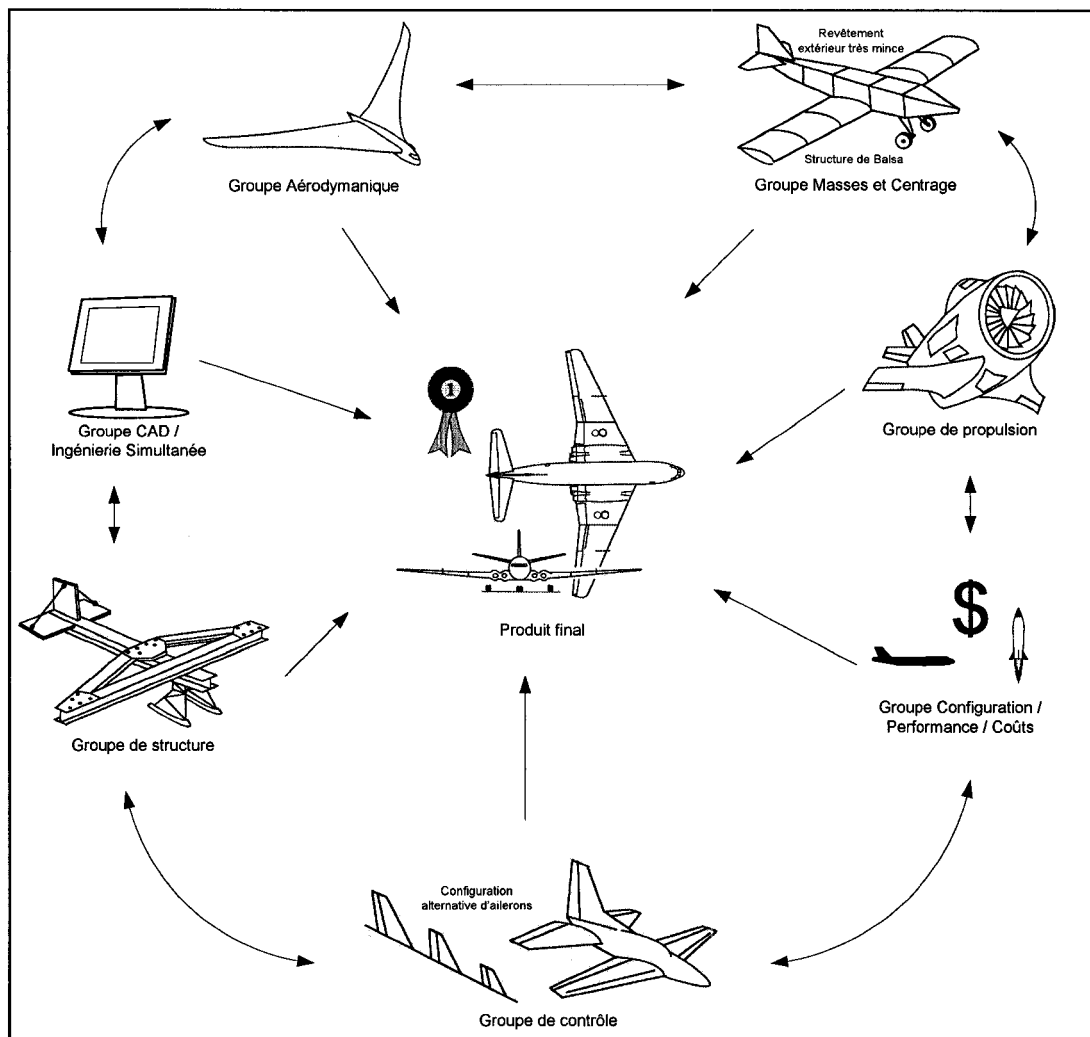


Figure 16 Principaux champs d'expertises impliqués dans le développement d'un avion
(adapté de [7])

Dans le domaine aéronautique, le processus de conception généralement employé est celui décrit par Raymer [22]. Ce modèle souscrit aux principes d'ingénierie simultanée et comprend, entre autre, les cinq phases suivantes :

- spécification des exigences et des contraintes de conception ;
- phase de design conceptuelle ;
- phase de design préliminaire ;

- phase de design détaillé ;
- fabrication.

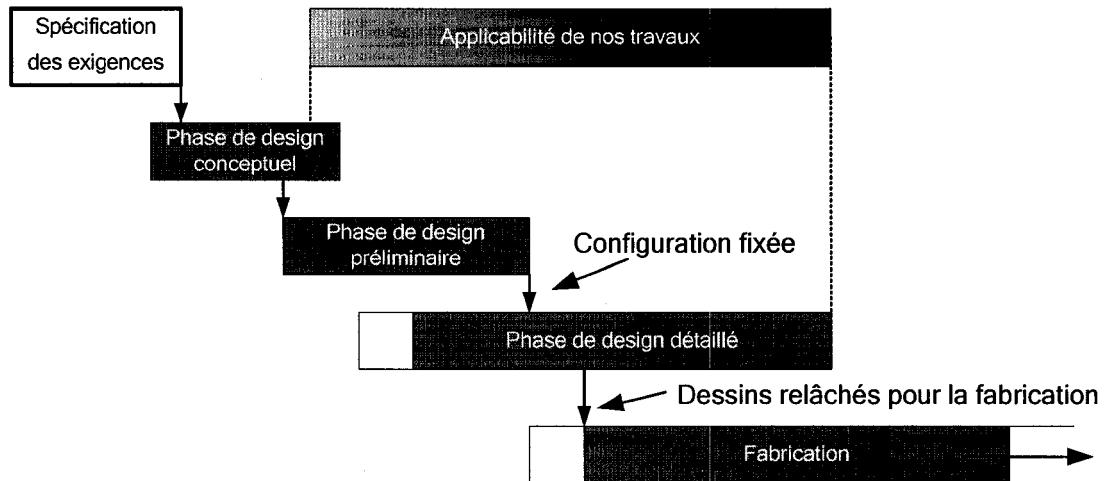


Figure 17 Processus de conception appliqué au domaine aéronautique (adaptée de Jenkinson [7] et de Raymer [22])

Comme on peut le voir à la figure 17 les différentes phases de conception se chevauchent représentant l'aspect simultané du processus. On remarquera que la phase de design conceptuel débute lorsque la spécification des exigences est terminée. Il est également à noter que la configuration de l'avion est fixée lorsque la phase de design préliminaire se termine.

Finalement, on peut voir sur la figure 17 le champ d'application de nos travaux. Nous nous intéressons particulièrement aux phases préliminaire et détaillée. Nous pensons que le développement d'outils orientés métier, tel que l'application automatique de soyages et de découpes conçue par Giguère, permet de réduire le temps consacré à ces phases de conception, donc de réduire la durée des cycles de développement et les coûts.

Il existe également d'autres modèles et outils s'adressant aux différentes phases du processus de conception dans le domaine aéronautique. Notons, entre autre, « *The*

Conceptual Aerospace Systems Design and Analysis » (CASDAT) développé par Hale [23]. Ces outils modulaires, développés pour et en collaboration avec les laboratoires de recherche de l'armée de l'air des États-Unis, supportent le processus de conception dès la phase conceptuelle. La figure 18 schématise l'architecture modulaire de l'outil CASDAT et démontre les différentes disciplines impliquées dans le processus de développement de l'avion.

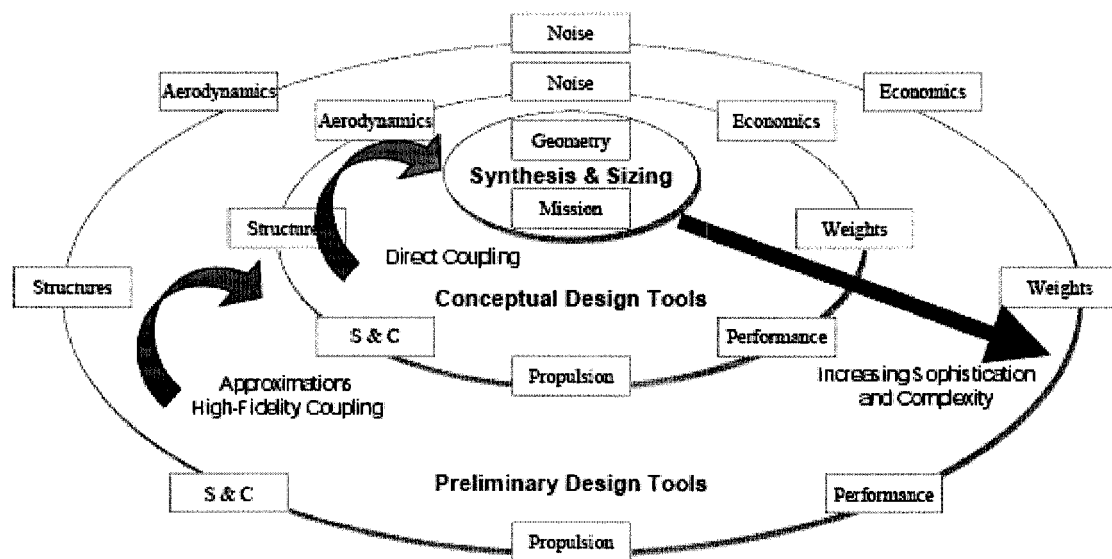


Figure 18 Architecture synthétisée du CASDAT [23]

2.1.4 La propagation des changements d'ingénierie

La propagation des changements d'ingénierie est l'une des étapes qui s'inscrit dans le processus global de la gestion des changements au sein d'une entreprise. Ce domaine est vaste et il ne fait pas parti des objectifs de ce mémoire de traiter ce sujet en particulier. Toutefois, nous l'abordons dans cette section afin de situer la contribution des applications orientées métier dans le processus de gestions des changements. Bien que les documents liés à la maquette numérique du produit (modèles de pièces, d'assemblages, etc.) soient généralement au centre du processus de conception, il y a

bien d'autres documents qui lui sont rattachés (gammes d'usinages, liste de matériel, liste d'exigences, etc.) et qui peuvent nécessiter une mise à jour lorsqu'une demande de changements est produite.

Maurino [24] définit une modification comme étant une évolution apportée à un document ou à un dossier validé. Une distinction importante est faite entre les évolutions apportées aux produits pendant le processus de conception, que Maurino nomme « corrections » de celles apportées à des données dites « validées » et « libérées », qu'il nomme « modifications ». Nous définissons ici les termes validé et libéré :

Données libérées :

Correspond à un « item » qui est considéré comme complet et approprié pour son usage par les autorités compétentes [25].

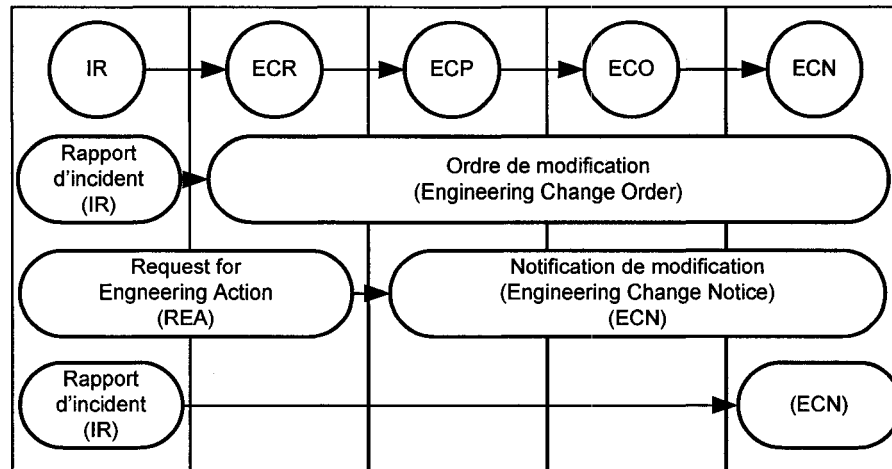
Données validées :

Correspond à une version de document ou de logiciel approuvé par les autorités compétentes et sur qui des procédures de changement de configuration peuvent s'appliquer [25].

L'évolution d'un produit peut être de différents types et n'implique pas nécessairement l'ingénierie. Lorsqu'il s'agit d'une évolution s'appliquant à des données validées et libérées, celle-ci fera intervenir le processus de gestion des modifications. Maurino [24] distingue quatre phases dans le processus de modification.

- Une phase de demande de modification
- Une phase d'instruction de la modification
- Une phase d'exécution de la modification
- Une phase d'application de la modification

Chacune de ces phases nécessite certaines procédures au bout desquelles une décision sera prise de passer à l'étape suivante ou non. La figure 19 montre un exemple de ces procédures.



Légende:

IR: Incident Report	ECR: Engineering Change Request
ECP: Engineering Change Protocol	ECO: Engineering Change Order
ECN: Engineering Change Notice	REA: Request for Engineering Action

Figure 19 Exemple de procédures de gestion des modifications [24]

Les demandes de modification, qu'elles soient issues d'une évolution du besoin du produit, d'une amélioration ou d'une non-conformité, débute par l'identification d'un « problème » (IR). Par la suite, une demande de changement (ECR) est produite et examinée par une commission de modification formée de représentants des différentes fonctions de l'entreprise (finance, achat, ingénierie, fabrication, etc.) qui évalue la pertinence et la faisabilité du changement. Lorsque la demande est acceptée, une proposition de modification (ECP) est créée afin d'analyser l'impact de la modification sur chacun des objets techniques associés. Une fois la proposition de modification acceptée, un ordre de modification (ECO) est créé pour permettre la mise à jour des documents impactés. Durant cette procédure un certain nombre de contrôles sont effectués. Lorsque les dossiers et modèles impactés par la modification sont mis à jour,

l'ordre de modification est transmis au service de gestion de la production qui étudiera la situation des fournitures, des ordres de fabrication, etc. Sur cette base, le service proposera une applicabilité de la modification. Finalement, l'ordre de modification est transmis à la commission de modification pour être validé et si elle est acceptée, une notification de modification (ECN) est envoyée aux différents services concernés par la modification, dans le cas contraire, la modification est annulée [24].

Kidd [26] décrit la gestion des changements d'ingénierie comme étant une activité pendant laquelle les changements sont surveillés et contrôlés durant tout le cycle de vie du produit et que cette gestion joue un rôle essentiel dans le succès des entreprises fabricant des produits. Il mentionne qu'au Japon plus de 60% des corrections sont effectuées durant la définition du produit et 10% des modifications surviennent lors du pré-lancement d'une redéfinition du produit. Au Royaume Unis et aux États-Unis 17% des corrections surviennent durant le processus de conception et 50% des modifications lors du pré-lancement de la production.

Kidd [26] fait également une comparaison entre le contrôle de la gestion des changements d'ingénierie réalisé à l'aide d'un système basé sur des documents papiers et à l'aide d'un système informatique dédié. Les systèmes fonctionnant à l'aide de papiers se sont avérés inefficaces et surtout très lents. En effet, seulement une personne à la fois peut réviser les documents impactés pour chaque domaine et le cycle de révision est souvent interrompu à cause de goulots d'étranglement. De plus, les changements multiples pouvant être demandés pour le même composant et qui progressent dans le système, aggravent souvent le problème et peuvent même avoir un impact négatif sur les autres pièces impliquées, ce qui devient difficile à déceler.

Selon l'étude effectuée par Bhuiyan et al. [27], la procédure de demande de changement, « Engineering Change Request » (ECR), doit être bien contrôlé et implanté au sein d'une entreprise pour minimiser les coûts et les délais. Pour démontrer l'impact de deux méthodes de gestion des ECRs, les auteurs ont développés un modèle informatique basé

sur les probabilités d'apparition des ECRs pendant le processus de conception d'un nouveau produit, « New Product Development » (NPD). Durant ce processus, les entreprises traitent généralement les ECRs un à un lorsqu'ils surviennent. Le modèle de Bhuiyan démontre qu'il est plus rentable et efficace de les traiter par lot. La raison principale est que la charge totale des processus est moins élevée lorsque les ECRs sont traités en groupe. En effet, comme le travail requis pour accomplir chacun des changements est toujours le même, les gains sont réalisés sur l'optimisation du temps consacré par les spécialistes lors de communications, sur le temps nécessaire aux rencontres pour réviser la conception et pour la gestion globale des changements. Certaines entreprises du domaine automobile et aéronautique appliquent déjà cette méthode pour des nouvelles versions et/ou révisions de leur produit.

Depuis les 20 dernières années, les systèmes PDM « *Product Data Management* » sont apparus avec comme objectif de prendre en charge la gestion des changements. Unigraphics a développé un système nommé « *Change Control Manager* », intégré à certains de leurs outils CAO, qui permet, entre autre, de maintenir une traçabilité des changements d'ingénierie dans le système. Notons également un système unique, nommé BAe(MAD) « British Aerospace (Military Aircraft Division) », développé par la division aérospatiale militaire britannique, exploitant les procédures de contrôle des changements d'ingénierie. Il s'est avéré plus rentable pour eux de développer leur propre système de gestion des changements plutôt que d'en modifier un déjà existant sur le marché [26].

Dans sa revue des recherches effectuées dans le domaine de la gestion des changements d'ingénierie, Wright [28] définit le changement d'ingénierie, « *engineering change* » (EC), comme étant une modification apportée à un composant d'un produit après qu'il soit entré en production. Il identifie, entre autre, deux domaines de recherche des changements d'ingénierie : tout d'abord il y a les outils (généralement complètement informatisés) utilisés pour analyser le problème générant la demande de changement ou synthétisant les solutions potentielles aux problèmes. La seconde catégorie regroupe les

méthodes possibles permettant de diminuer l'impact d'un changement sur la fabrication et le contrôle de l'inventaire.

Un aspect important des changements d'ingénierie concerne le moment où ils surviennent. En effet, lorsqu'ils se manifestent au début du processus de développement, ils nécessitent du temps d'ingénierie mais ne requièrent aucune intervention au niveau du matériel et de la documentation du produit par exemple. De plus, les modifications survenant tôt dans le processus ne sont pas traitées par le processus de gestion des modifications. Ainsi, les changements qui sont effectués durant la phase conceptuelle du processus coûteront environ 1000\$ en temps d'ingénierie. Cependant, lorsqu'ils surviennent pendant la phase de conception détaillée du produit ils coûteront 10 000\$. Finalement, les changements d'ingénierie effectués lors de la phase de production et/ou d'utilisation du produit, changements effectués sur des données validées et libérées, sont ceux qui pénalisent le plus l'entreprise car ils font appel au processus de gestion de modifications qui implique l'ingénierie, la fabrication, l'outillages, les achats de fournitures, etc. [7;13;14;26].

Chez Bombardier Aéronautique les demandes de changements d'ingénierie ont dépassé le nombre de 13000 pour la seule année 2001 [2].

La gestion des changements et la façon dont ils sont propagés constitue un facteur déterminant sur les coûts totaux de production d'un produit. Les changements sont généralement propagés de façon manuelle et sont tributaires des erreurs humaines pouvant entraîner ainsi de l'incohérence au sein des données du produit avec comme conséquences d'augmenter les coûts.

Dans le domaine de la CAO, par exemple, un changement demandé peut être effectué dans le modèle tridimensionnel d'une pièce, mais les plans 2D servant à la production de celle-ci peuvent ne pas être mis à jour. La chaîne de propagation est ainsi interrompue provoquant de l'incohérence entre le modèle 3D de la pièce et les plans de production.

Si l'erreur est découverte avant la fabrication de la pièce, une partie des coûts sera évitée, toutefois, il faudra éventuellement établir le point de rupture et terminer la propagation du changement demandé. Si l'erreur est découverte après la fabrication de la pièce, au contrôle de qualité par exemple, il en résultera une pièce défectueuse pouvant être complètement rejetée et nécessitant une nouvelle fabrication. La pièce peut également être récupérée moyennant un «*rework*», mais dans les deux cas, l'erreur résulte en une augmentation des coûts.

Il serait pertinent de disposer d'un outil assurant la propagation des changements dans tous les documents impactés reliés au produit en question. En pratique cela est très difficile à cause du nombre et de la variété des documents impliqués. En effet, en plus du modèle 3D et des plans 2D, d'autres types de documents, comme par exemple, la nomenclature d'ingénierie « BOM », les dessins d'installations et d'outillages ainsi que les documents des méthodes doivent être modifiés.

Ce travail aborde le problème de la propagation des changements au sein de la maquette numérique seulement. Dans un assemblage, lorsqu'une pièce est modifiée, les autres pièces dont la définition est dépendante doivent également être modifiées. Une propagation automatique de ces changements fait gagner un temps précieux et assure la cohérence des données. Que ce soit durant les phases de développement du produit (conceptuelle, préliminaire, détaillée) ou lors de l'applicabilité des ECRs au sein du processus de modification, la propagation automatique des changements fait gagner un temps précieux.

Giguère [3-5], dans son travail sur les soyages et les découpes d'un cadre à l'intersection d'une lisse, utilise un lien de dérivation pour associer, dans un premier temps, les pièces interdépendantes, et un processus résident en mémoire propage automatiquement les changements aux pièces impactées lorsqu'un changement est détecté par l'application.

Nous verrons que dans le développement de notre application de définition automatique de poches, la tâche de propagation est assurée par les liens persistants établis dans la maquette.

2.2 Conception par caractéristiques

Cette section aborde l'approche de conception par caractéristiques de plus en plus utilisée dans les outils CAO modernes. Les recherches sur cette technique ont débuté il y a près de 20 ans, mais les résultats commencent à peine à se faire sentir [29;30]. La principale motivation pour le développement de cette approche vient du désir de structurer l'information afin de pouvoir la manipuler plus aisément et surtout plus rapidement.

L'utilisation des caractéristiques est pertinente à ce travail car, en plus d'accélérer l'accomplissement de certaines tâches de modélisation, les caractéristiques nous permettent d'encapsuler de la géométrie et des connaissances propres à une tâche spécifique dans une même « entité ». Il nous suffit par la suite d'instancier les caractéristiques désirées dans le contexte approprié.

Finalement, les caractéristiques permettent de capturer l'intention de conception directement dans l'outil CAO sous différentes formes.

2.2.1 Bref historique de la CAO

C'est à la fin des années 50 et au début des années 60 que la CAO fit son apparition. En fait, on parle plutôt d'outils de « dessin » assisté par ordinateur. C'est Ivan Sutherland, en 1963, qui a développé le premier outil informatique permettant de créer des dessins tant au niveau artistique que technique. Son outil, le « SketchPad » est considéré comme l'ancêtre de la CAO [31].

Au tout début, les systèmes fournissaient des fonctions essentiellement en 2D. Vers la fin des années 60 début 70, le désir de disposer d'un modèle virtuel en trois dimensions (3D) se fait sentir. Les concepteurs de cette époque étaient motivés par l'extraction automatique des différentes projections en 2D et pour ce faire, ils avaient besoin d'un modèle initialement conçu en 3D.

Deux approches de création de modèles tridimensionnels informatiques furent développées. La première, issue de Ian Braid et de ses collègues de l'Université de Cambridge, se nomme BREP, «Boundary Representation», et consiste à bâtir le modèle à l'aide de facettes (sous ensembles de surfaces planes, quadratiques ou toroïdales). La deuxième approche, issue de Voelcker et Requicha de l'Université de Rochester, est dite CSG, «Constructive Solid Geometry», et consiste plutôt à soustraire et additionner des volumes afin d'obtenir une représentation géométrique de la pièce. La plupart des modeleurs solides actuels utilisent ces deux approches et sont appelés modeleurs «hybrides» [29].

2.2.2 Définition d'une caractéristique

Avant d'aller plus loin, il convient de formaliser la définition d'une caractéristique. En effet, la définition d'une caractéristique est très variée dans la littérature et diffère d'un auteur à l'autre et d'un domaine à l'autre. C'est pourquoi nous prenons la peine de bien établir celle que nous utiliserons tout au long de ce travail.

Cunningham [32] définit une caractéristique comme étant n'importe(s) quelle(s) forme(s) ou entité(s) géométrique(s) utilisée(s) dans une ou plusieurs activités de conception ou de fabrication de produit.

Dixon [33] définit une caractéristique simplement comme étant une entité ayant une signification tant sur sa forme que sur sa fonction.

Bidarra [34;35], quant à lui, définit une caractéristique comme étant une représentation des aspects d'une forme d'un produit correspondant à une forme générique et ayant une signification fonctionnelle selon une phase du cycle de vie du produit.

Deneux [36] s'inspire de la définition proposée par le FEMEX, « Feature Modelling Experts Working Group » définissant une caractéristique comme étant une unité d'information qui décrit une agrégation de propriétés d'un modèle produit pertinente à une vue spécifique du produit.

Zimmermann [37;38] définit une caractéristique comme un objet, utilisé dans un modèle produit selon le point de vue adopté à une étape spécifique du développement, qui représente une composante d'une pièce contenant de l'information géométrique et non-géométrique. Une caractéristique est plus petite qu'une pièce, mais plus grande qu'une face.

Selon Hummel & Brown [39] une caractéristique est simplement une classification distinctive d'un objet ayant une signification dans un domaine.

Pour Shah & Mantyla [29], une caractéristique constitue une partie physique d'une pièce. Elle donne une signification à l'information (géométrie d'une pièce, élément(s) d'un assemblage, type de matériel, etc.) contenu dans le modèle produit, possède des propriétés prévisibles et attribuables à une forme générale en fonction du point de vue du domaine d'application adopté.

Une autre définition, moins formelle, est celle qu'utilisent implicitement les modelleurs modernes tels que Catia V5, SolidEdge, SolidWorks, etc. En effet, ces modelleurs ont tendances à désigner des fonctions spécifiques de l'outil comme étant des caractéristiques. Par exemple, un «Sweep» ou un «CutOut » dans SolidEdge sont vus et traités comme des caractéristiques dans l'application. Il est vrai que ces opérations

portent une information sur l'intention de conception, mais celle-ci demeure à un très bas niveau. Les caractéristiques, au sens exprimé dans les définitions précédentes, informent sur l'intention du concepteur à un plus haut niveau. La fonction «trou» que l'on retrouve intégré sous forme de caractéristique dans plusieurs modeleurs modernes est plus près de la définition scientifique d'une caractéristique.

Dans ce projet, nous utiliserons les caractéristiques en adoptant la définition proposée par Deneux stipulant qu'une caractéristique est une unité d'information qui décrit une agrégation de propriétés d'un modèle produit pertinente à une vue spécifique du produit.

2.2.2.1 Attributs des caractéristiques

Les attributs sont des spécifications ou des qualités distinctives pouvant être utilisés au niveau d'une caractéristique, d'un ensemble de caractéristiques, d'une pièce complète ou d'un assemblage. Elles peuvent également enregistrer de l'information relative à une relation entre caractéristiques ou un ensemble de caractéristiques. Shah & Mantyla [29] distinguent trois types d'attributs :

<i>Les attributs d'assemblage :</i>	Accouplement de surfaces, dégagement, profondeur d'insertion, vecteur d'orientation, etc.
<i>Les attributs des pièces :</i>	Spécifications du matériel, numéro de pièce ou données administratives.
<i>Les attributs de caractéristiques :</i>	Position, orientation, dimension, forme, tolérance, etc.

2.2.2.2 Types de caractéristiques

Shah & Mantyla [29] regroupent les caractéristiques selon leur fonction et les classent de la façon suivant :

<i>Caractéristiques de forme :</i>	Portion nominale de la géométrie.
<i>Caractéristiques de tolérance :</i>	Déviations par rapport au nominal : forme, grandeur, position, etc.
<i>Caractéristiques d'assemblage :</i>	Regroupement permettant de définir les relations entre les caractéristiques d'un assemblage.
<i>Caractéristiques fonctionnelles :</i>	Ensemble relatif à des fonctions spécifiques : intention de conception, paramètre non géométrique relatif à la fonction, à la performance, etc.
<i>Caractéristiques matérielles :</i>	Composition du matériel, traitement, conditions, etc.

Les trois premières classes de caractéristiques se rapportent à la portion géométrique d'une pièce, c'est pourquoi Shah & Mantyla les regroupent sous une méta-classe appelée «caractéristiques géométriques». À leur tour, les caractéristiques géométriques peuvent être classifiées selon l'application dans laquelle elles sont utilisées : caractéristiques de conception, de fabrication, d'inspection, etc.

2.2.2.3 Propriétés des caractéristiques

L'une des fonctions majeures des caractéristiques est de créer des associations entre les entités définissant le produit. Shah & Mantyla [29] distinguent différents types de

propriétés des caractéristiques: génériques, spécifiques, intrinsèques, extrinsèques et autres.

La définition des propriétés génériques et spécifiques des caractéristiques est fondamentale en regard aux bénéfices attendus de cette approche de conception. Les propriétés génériques enregistrent le sens du point de vue adopté (ingénierie, fabrication, etc.) ainsi que la forme générale de la caractéristique. Elles fournissent également les éléments de bases afin d'appliquer le comportement prévisible de la caractéristique. Les propriétés spécifiques, quand à elles, sont plus microscopiques ou locales à la caractéristique comme par exemple la position ou le diamètre d'un trou.

Les propriétés intrinsèques ou «intra-caractéristiques» sont des attributs spécifiques à une caractéristique et indépendantes des autres caractéristiques du modèle. La forme géométrique de base, des paramètres d'identification, les tolérances de forme en sont des exemples.

Les propriétés extrinsèques ou « inter-caractéristiques » sont des attributs impliquant plusieurs caractéristiques entre elles, notons, entre autre, les paramètres et les dimensions d'une caractéristique, les contraintes de grandeur, de localisation ou d'orientation d'une caractéristique, etc.

Finalement, les caractéristiques permettent d'enregistrer d'autres types d'informations, qui ne sont pas essentielles à la définition géométrique du modèle ou à sa manipulation, mais qui peuvent être vitales pour une autre application. Shah & Mantyla [29] les classes sous « autres attributs » :

<i>Attributs non géométriques :</i>	nom, code de composition, propriété du matériel, traitement thermique, de surface, revêtement, etc.
-------------------------------------	---

Attributs symbolique axe central, point central, section
ou *représentation squelettique* : transversale, plan médian, profils 2D,
enveloppe 2D ou 3D, symbole ou icône
ne correspondant pas à la géométrie
réaliste.

2.3 La représentation du savoir-faire d'ingénierie et de la connaissance

L'un des objectifs de ce travail consiste à intégrer une partie du savoir-faire d'ingénierie et de la connaissance spécifique à une tâche de modélisation au sein de l'outil CAO.

La capture des connaissances en milieu industriel est un domaine en soi et beaucoup de recherches sont en cours pour développer et établir des systèmes, généralement informatiques, capturant le savoir des employés spécialisés de multiples façons. En effet, dans le domaine industriel, tel qu'en aéronautique, les employés impliqués dans les différents aspects de la conception d'un aéronef acquièrent et développent beaucoup de connaissances spécialisées. Ces connaissances partent avec les spécialistes lorsqu'ils quittent l'entreprise. De plus, tant que la connaissance fait partie intégrante du concepteur, elle ne profite directement qu'à lui-même. Il serait profitable pour une entreprise de pouvoir capturer le plus de connaissance possible de ses employés pour la conserver, mais aussi pour la rendre accessible à l'ensemble de l'entreprise.

Les concepteurs se réfèrent et réutilisent, consciemment ou inconsciemment, le savoir acquis lors de leurs expériences de conception précédentes. C'est pourquoi une grande partie de l'activité de conception est en fait ce que l'on appelle du « re-design ».

Deneux [40], dans son travail sur le développement d'un modèle de connaissance pour le « re-design » fonctionnel, le définit comme étant une conception non-innovatrice d'un produit à partir d'éléments standards ou maîtrisés par le concepteur. Il décrit

également la conception fonctionnelle d'un produit comme un processus itératif en constante interaction entre le «QUOI» et le «COMMENT».

Dans son travail, Deneux [40] identifie deux approches principales de la représentation du savoir. La représentation symbolique qui utilise des règles de conception et des actions-conditions systématiques, «IF conditions THEN actions» par exemple, et la représentation connexionniste de la connaissance qui repose sur le paradigme des réseaux de neurones artificielles. Le modèle qu'il propose incorpore également la représentation de la connaissance imprécise à l'aide de « Fuzzy systems ».

Le modèle proposé par Deneux [40] se compose de concepts et de relations entres-eux. Les concepts sont de deux natures : il s'agit de problèmes ou de solutions. Les concepts sont divisés en deux niveaux : le niveau conceptuel, représenté par un symbole, et le niveau des paramètres, représenté par des valeurs pouvant être linguistiques ou numériques. La représentation formelle du modèle est la suivante : $C = (A, V)$. Si le concept est un problème (A), alors (V) représente un ensemble de solutions potentielles. La figure 20 donne l'exemple du concept de porte ayant deux paramètres fonctionnels possibles, ouvert ou fermé. Dans le même ordre d'idée, si le concept est une contrainte (A), alors (V) représente les valeurs possibles que la contrainte peut prendre.

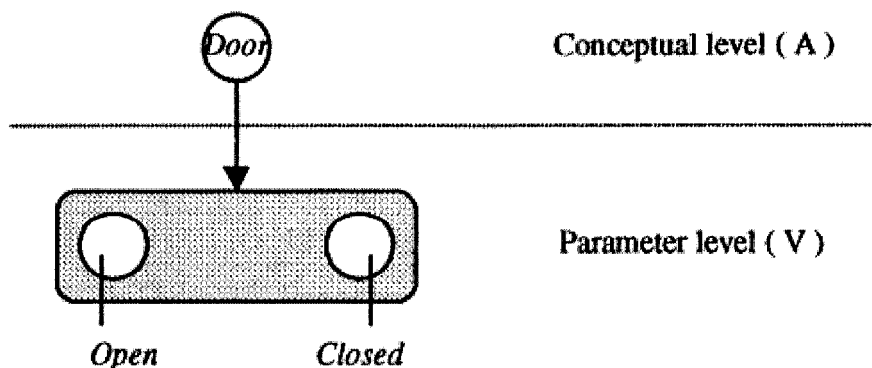


Figure 20 Représentation linguistique du concept de porte [40]

Les concepts sont reliés entres-eux par des relations tant au niveau conceptuel qu'au niveau des paramètres. Des relations établies entre un problème et des contraintes expriment le fait que les contraintes affectent directement la solution.

Les relations sont de deux types, soit statiques, modélisant des connaissances arrêtées dans le temps tel que des lois physiques ou mathématiques, voir la figure 21. Les relations peuvent être également dynamiques, modélisant les connaissances d'experts et ne pouvant être définitivement établies dans le temps, voir la figure 22.

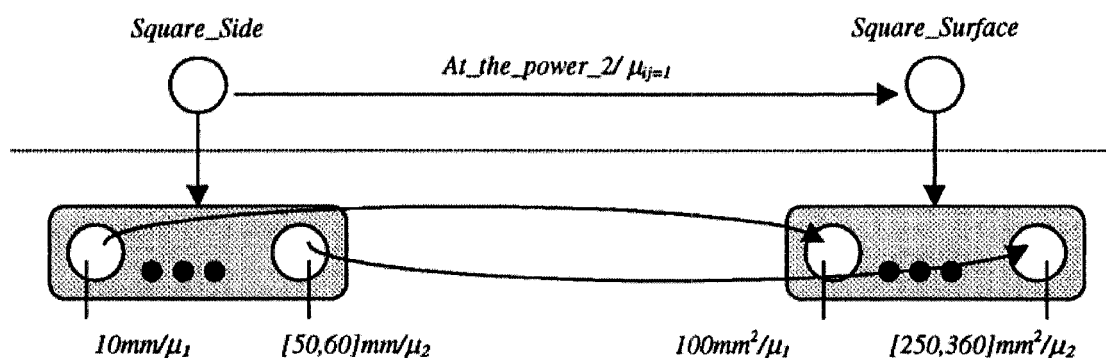


Figure 21 Exemple de relations statiques entre les concepts [40]

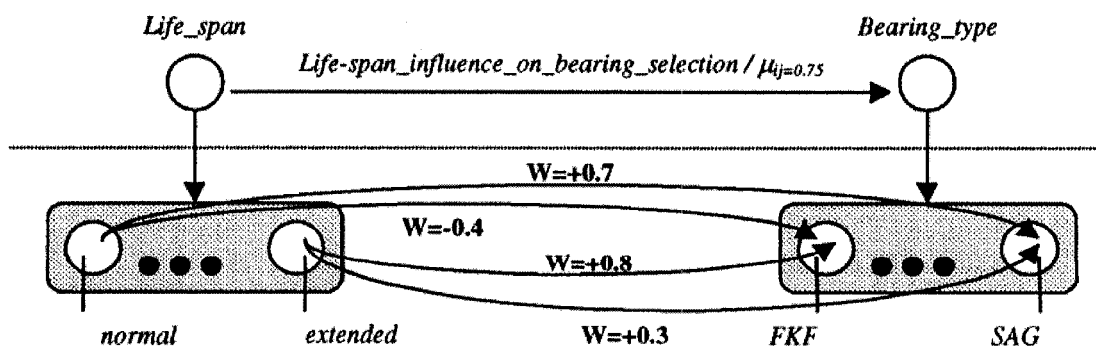


Figure 22 Exemple de relations dynamiques entre les concepts [40]

Deneux [40] a développé un prototype d'application de son modèle dans un contexte réel de « re-design » lors d'un projet de recherche appelé SODA, « Système d'Outillage D'Assemblage » en collaboration avec Dassault Aviation. Ce projet consistait à établir les spécifications de l'architecture d'un système informatique dédié à la conception fonctionnelle de systèmes d'outillages d'assemblage de pièces aéronautiques. Il a été démontré que ce modèle a la capacité de représenter les connaissances des experts et supporte le processus de décision basé sur une approche fonctionnelle du « re-design ».

Fruchter [41] a également développé un système permettant la réutilisation des connaissances contextuelles d'une corporation. La figure 23 représente le cycle de vie de la connaissance dans une entreprise au cours duquel elle est créée, capturée, indexée et stockée dans un système d'archivage. Plus tard dans le processus, la connaissance devra être extraite du système d'archivage pour être réutilisée.

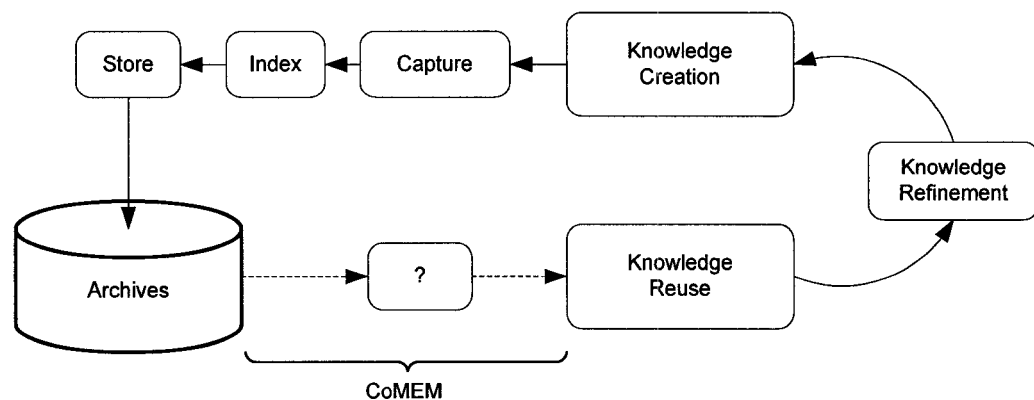


Figure 23 Cycle de vie de la connaissance [41]

Fruchter [41] distingue deux types de réutilisations de la connaissance : la réutilisation interne, qui correspond à la connaissance contenue dans la mémoire des concepteurs et la réutilisation externe, qui correspond à la connaissance stockée dans un système de gestion de la connaissance. Il introduit la notion de « connaissances contextuelles » qui

représente le savoir de conception tel qu'il survient dans la mémoire d'un concepteur : riche, détaillée et contextuelle. Il définit également la «mémoire corporative» comme étant l'ensemble centralisé des «connaissances contextuelles» de l'entreprise. Cet ensemble représente en fait les connaissances externes reliées aux projets réalisés par l'entreprise dans le passé et stockées dans une base de données accessible aux employés.

À l'aide d'observations empiriques, Fruchter a identifié trois étapes dans le processus interne de réutilisation du savoir : le retrouver, explorer l'historique de son évolution et explorer son contexte. Il affirme qu'un système doit émuler ce processus pour réutiliser la connaissance de façon externe. Il propose un prototype nommé CoMem, « *A Corporate MEMory computer environment* », développé dans le cadre d'un projet de recherche sur la gestion de la connaissance mené au « *Project-Based Learning Laboratory* » à l'université de Stanford. Ce prototype supporte les trois activités mentionnées ci-haut à l'aide de trois modules distincts : un module de vue d'ensemble, un module d'exploration de projets contextuels et un explorateur de l'évolution de l'historique.

Zimmermann [37;38] introduit une nouvelle approche de liens qu'il nomme ULEO, « Universal Linking of Engineering Objects » visant à permettre la circulation d'informations de haute qualité entre les différentes applications du processus de développement du modèle produit, qu'il nomme «ProSAp», « Process Step Application ». Dans un premier temps, il définit les EOs, « Engineering Objects » qui sont à la base du modèle et correspondent à tous les objets ayant un rapport à l'ingénierie tel que des modèles CAO d'assemblage, des caractéristiques, des pièces, des surfaces, des tolérances, des matériels, etc. Les EOs et les interrelations correspondantes sont modélisés de deux façons. D'abord, les EOs sont divisés en différentes classes à l'intérieur d'une taxonomie centrale nommée UMEO, « Unified Model of Engineering Objects » afin que les différents types d'applications spécifiques, les EOs standardisés et non-standardisés, soit disposés selon une taxonomie unique et globale, assurant ainsi que

toutes les applications qui accèdent le modèle obtiennent la même information et utilise les mêmes concepts. Ensuite, tous les types de relations entre les classes de EOs et/ou leurs instances sont modélisés à l'intérieur d'une MTRT, « Meta Taxonomy of *Relation Types* », laquelle est logiquement située à un niveau d'abstraction au dessus du UMEO.

Les relations inter-pièces et intra-pièces peuvent être de types héritages, agrégations ou EORs, « *engineering objects relations* », lesquelles sont des associations directes de spécialisations. La MTRT couvre les IEORs, « *informational EORs* », qui représentent les relations logiques entre les classes d'EOs, aussi bien que les GEORs, « *generative EORs* », qui encapsule le savoir-faire d'ingénierie en indiquant la façon d'instancier les EOs et les EORs automatiquement.

Zimmermann applique ses concepts à l'aide d'un prototype d'application dans le contexte de fabrication des têtes de cylindres des moteurs automobiles. Il utilise le ULEO et les EOs pour établir des liens entre les différents domaines d'applications impliquées dans la fabrication : la conception, l'outillage, la planification de la fabrication et le contrôle de la qualité. L'un des scénarios de l'application parcourt les caractéristiques et les pièces impliquées dans le processus de fabrication : les modèles des pièces des têtes de cylindres à l'état finis et à l'état brut, l'outillage de fabrication et de finition pour instancier automatiquement un système permettant de fixer la pièce brute à l'outillage pendant l'usinage.

2.4 Les UDFs (User-Defined Features)

De plus en plus de modeleurs CAO modernes sont basés sur la conception par caractéristiques, bien que le sens attribué à cette expression par les éditeurs d'outils CAO ne correspond pas toujours aux définitions établies dans la littérature scientifique. En effet, dans un prisme, par exemple, il n'y a aucune intention de conception ou d'intention orientée métier et pourtant les éditeurs utilisent le terme caractéristique (feature) pour désigner ce type d'objet.

Dans un système bâti selon le paradigme de conception par caractéristiques, celles-ci constituent l'unité de base permettant de modéliser une pièce qui s'insère dans un assemblage et représente finalement un produit. Il est impossible pour les éditeurs d'outils CAO de fournir une bibliothèque incorporant toutes les caractéristiques possibles pour tous les domaines. Généralement, les modeleurs possèdent un répertoire de base se limitant aux caractéristiques les plus souvent utilisées.

Un mécanisme procédural, les UDFs (User-Defined Features), est proposé pour combler le vide entre les caractéristiques de base et celles plus spécifiques à des domaines en particulier. Ce type d'outil permet à un usager de définir ses propres caractéristiques et de les instancier dans ses modèles au besoin.

SolidEdge est l'un des modeleurs possédant un outil se rapprochant de la définition de caractéristiques propre à l'utilisateur, le «*Systems libraries*» [42]. Il permet de regrouper des pièces et de les instancier dans un assemblage. Il ne permet toutefois pas de définir une seule pièce ou une seule caractéristique spécifique, comme un embossage ou un trou d'allègement par exemple. Il est probablement possible de contourner ce problème en utilisant la conception en contexte d'une seule pièce, mais déjà on constate des limitations quand à l'utilisation de ce module. Les modeleurs plus puissants, tel que NX 3.0 et Catia V5, possèdent un outil de création de caractéristiques plus sophistiqué.

Shah et al. [30;43;44] ont développé le «*ASU testbed*», un interpréteur de commande (Shell), composé d'une librairie de caractéristiques génériques organisées sous forme d'une liste de propriétés. Chacune des caractéristiques possède un type l'identifiant, un nom, une liste de caractéristiques compatibles et une représentation solide sous forme d'arbre CSG. Les primitives créées à l'aide de cette approche procédurale sont destinées à l'usage d'application basée sur les caractéristiques en vue d'un prototypage rapide.

Hoffman [45;46] distingue deux étapes du mécanisme des UDFs. La première concerne sa définition et correspond au processus de création du prototype de la caractéristique; la deuxième étape correspond à l'utilisation de l'UDF en l'instanciant dans un contexte adéquat.

Pour assurer le bon fonctionnement du mécanisme d'UDFs, il dresse également certaines exigences que l'outil CAO doit remplir :

- un UDF peut être défini en utilisant les caractéristiques standards contenues dans un outil CAO et/ou des UDFs définis précédemment ;
- l'utilisateur doit pouvoir placer et orienter l'UDF dans une pièce pendant sa modélisation à l'aide des procédures d'insertion standard ;
- les UDFs doivent fournir un mécanisme de définition pour assurer la validité de l'instanciation et de l'attachement des UDFs ;
- les UDFs doivent supporter les attributs non-géométriques et les contraintes.

Hoffman [46] a développé un format de langage neutre pour la représentation des caractéristiques de formes et des contraintes : « *Erep -- An Editable, high-level REPresentation for geometric design and analysis* ». Le terme « neutre » signifie que l'application est développée indépendamment de toute plateforme CAO (Catia, SolidEdge, etc.).

Il utilise ce langage pour construire un UDF de cylindres de lubrification insérés dans une bielle de moteur à combustion [45]. Il démontre certains mécanismes de base pour la réalisation d'UDFs permettant de construire des formes complexes, intégrant des attributs et des paramètres. La figure 24 ci-dessous, montre la bielle et l'UDF comprenant les cylindres de lubrification.

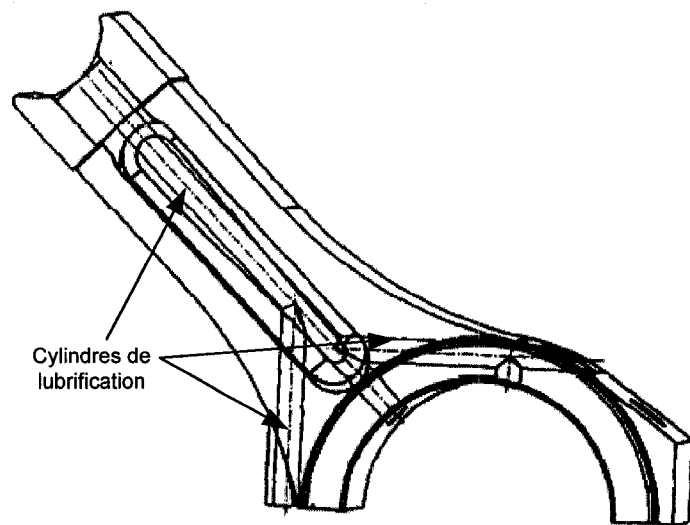


Figure 24 Exemple d'UDF: Cylindres de lubrification dans une bielle [45]

Bidarra [35;47], quant à lui, propose une structure pour agencer les différentes classes de caractéristiques définies par les usagers. Il établit d'abord ce qu'est une classe de caractéristiques spécifiées par l'utilisateur, il décrit ensuite les spécifications de formes et les paramètres de ces caractéristiques. Il aborde également les aspects géométriques, topologiques et fonctionnels permettant de valider des caractéristiques définies par les usagers. La figure 25 présente un exemple de la structure d'une classe de caractéristiques.

Bidarra, dans son travail intitulé « Semantic feature modelling » [35], implante cette approche de modélisation de caractéristiques dans un système nommé « Spiff » développé au « Delft University of Technology ». La figure 26 démontre l'architecture de ce modèleur.

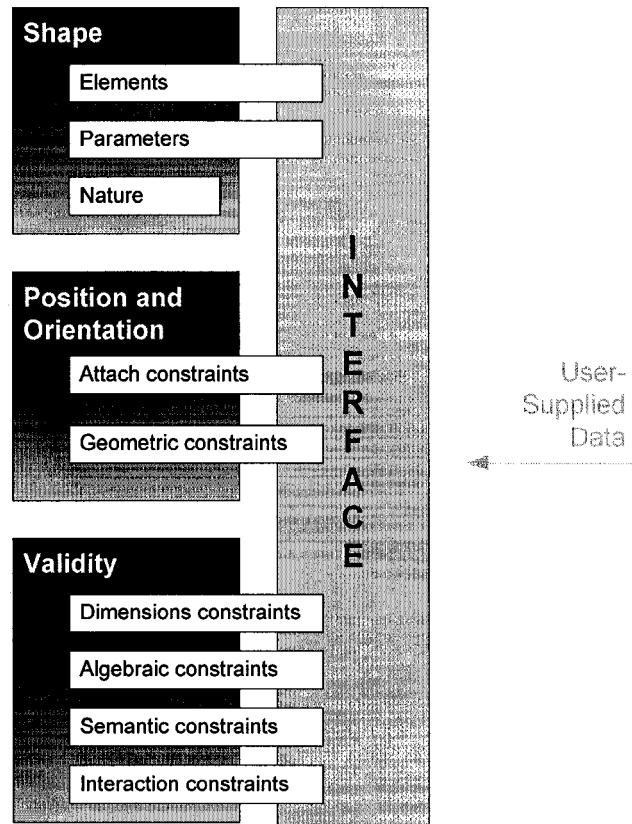


Figure 25 Exemple de structure d'une classe de caractéristiques [35;47]

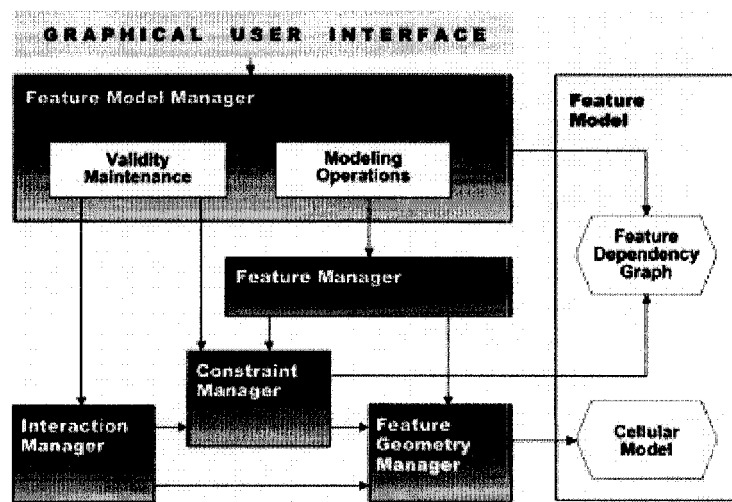


Figure 26 Architecture du modeleur de caractéristiques Spiff [35]

Lederman [48] a travaillé en collaboration avec « Airbus Germany », le « German Aerospace Center (DLR) » et le « Swiss Federal Institute of Technology » de Zurich sur un projet développant de nouvelles méthodes qui permettent d'améliorer la précision, l'efficacité et la flexibilité de la prédiction de données diverses utilisées dans la conception préliminaire des structures d'un avion.

Lederman utilise les UDFs et des scripts VB pour créer un « objet dynamique » qui modifie automatiquement le nombre de cadres et de lisses dans la maquette numérique de l'avion lorsque la longueur d'une section du fuselage est allongée ou raccourcie, la figure 27 ci-dessous présente la structure générale du modèle.

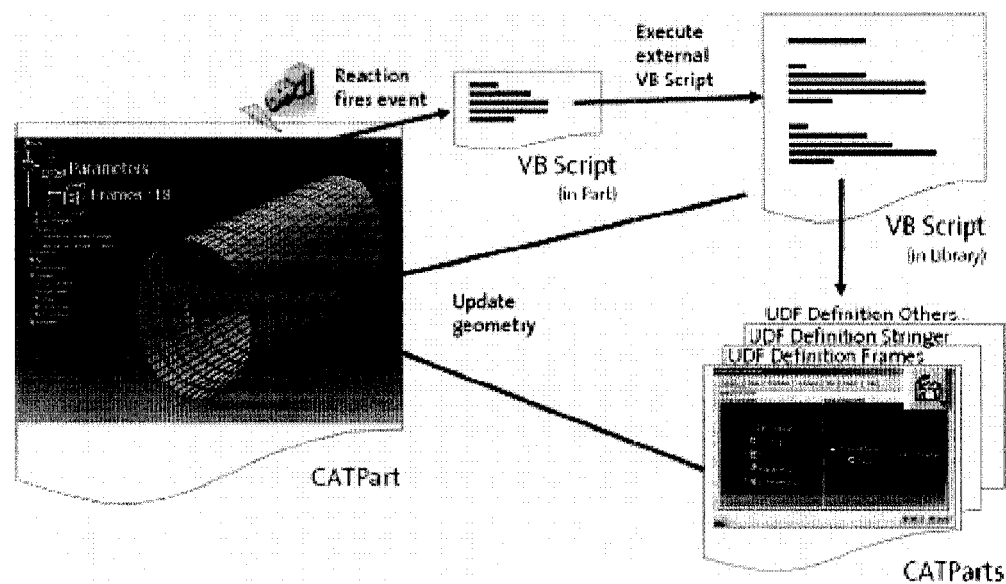


Figure 27 Structure d'un modèle dynamique faisant varier le nombre de cadres et de lisses [48]

Ce type de modèle est lié et réagit à une procédure d'analyse par éléments finis optimisant le nombre de lisses et de cadres dans une section du fuselage pour minimiser le poids total de l'avion, la figure 28 démontre la structure globale du mécanisme.

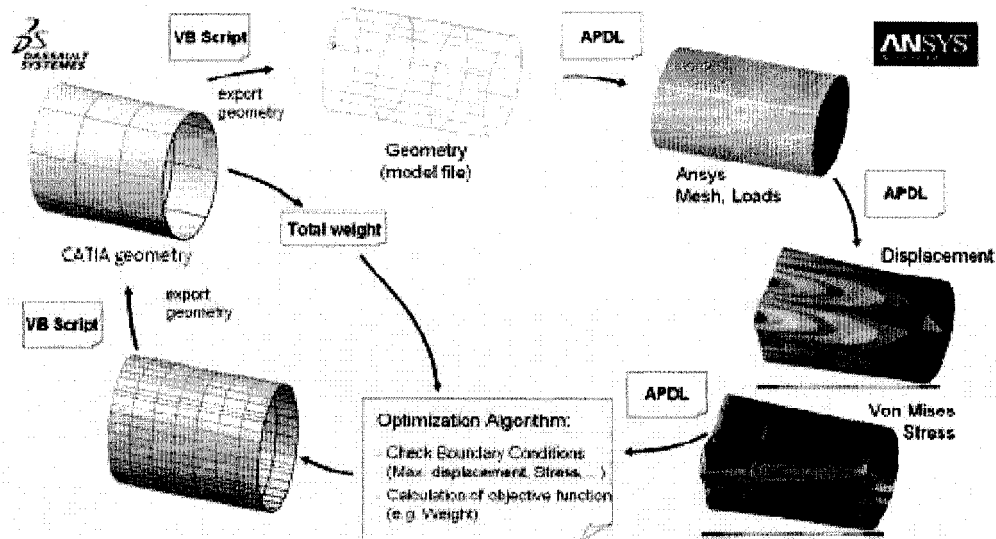


Figure 28 Représentation schématique de la procédure d'optimisation [48]

La géométrie est tout d'abord exportée de Catia V5 dans un fichier à l'aide d'un script VB. Par la suite, le modèle est importé dans ANSYS, à l'aide d'un script APDL, où les différentes forces sont appliquées et où une analyse par éléments finis est effectuée. On ajoute, déplace ou modifie les cadres et les lisses à l'aide d'un script APDL et on exporte la géométrie d'ANSYS qui est ensuite importée dans Catia V5 à l'aide d'un script VB.

Dans la même philosophie d'association, Subramani [49] propose un algorithme permettant de mettre à jour automatiquement les caractéristiques des modèles entre différents domaines, par exemple, la conception détaillée de la pièce, la planification de son processus de fabrication et l'assemblage.

Dans l'environnement d'édition de modèles « multi-way » de Subramani, la « vue » dans laquelle l'édition et les modifications ont lieu se nomme « edition-view », les caractéristiques dans cette vue qui sont créées, effacées ou modifiées se nomme « edit-feature » et finalement, la vue qui doit être mise à jour se nomme « target-view ». Par

exemple, la hauteur de la nervure 1 (Rib 1) de la figure 29 est modifiée par le concepteur.

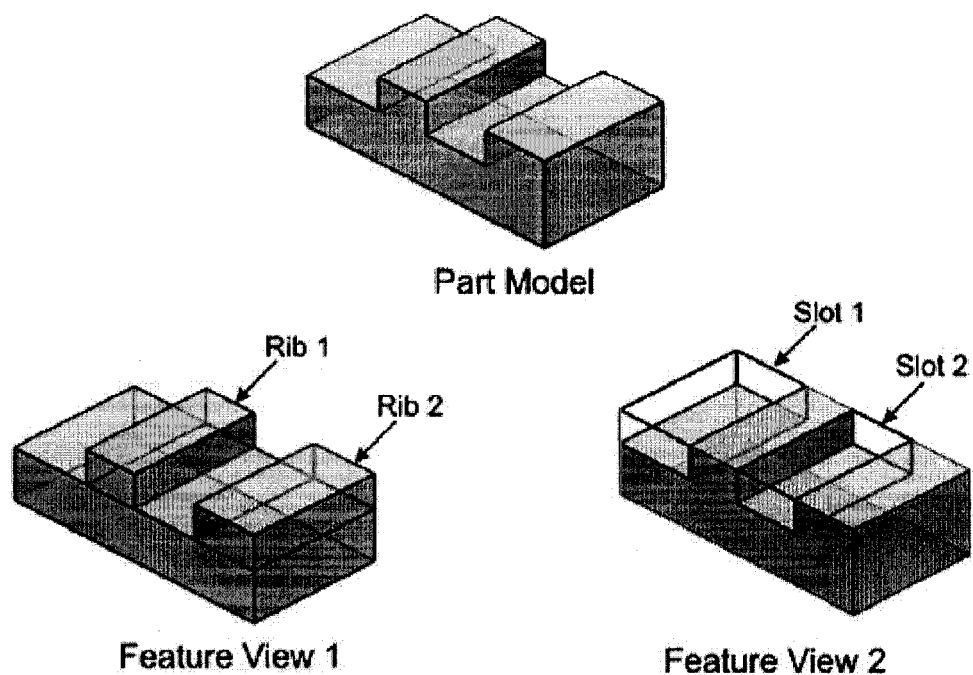


Figure 29 Exemple de vues multiples de caractéristiques [49]

La modification de la hauteur de la nervure dans la « vue » ou le domaine de conception doit se refléter dans celui de la fabrication en ajoutant un nouveau parcours d'usinage, voir la figure 30 ci-dessous.

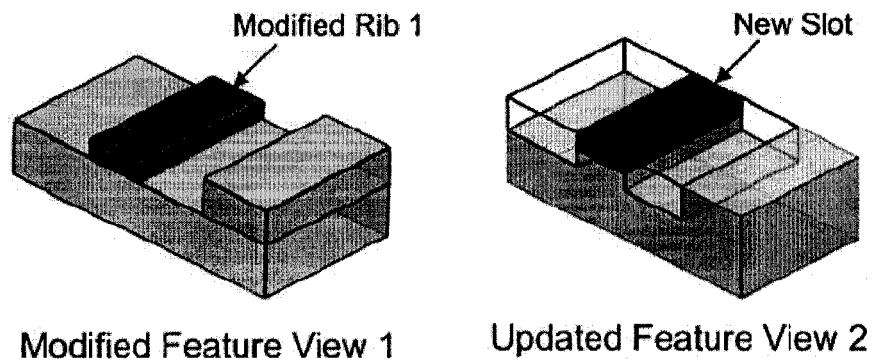


Figure 30 Associativité entre différentes « vues » d'une pièce [49]

Subramani identifie trois types de modifications possibles dans un environnement distribué basé sur les caractéristiques :

1. les changements au niveau des dimensions et des contraintes dans un modèle ;
2. l'ajout et la suppression de caractéristiques ;
3. les changements au niveau des relations entre les formes des éléments d'une vue dues aux changements survenus dans une autre vue.

La procédure de mise à jour qu'il a développée traite des deux premiers types mentionnés ci-haut.

Son algorithme de mise à jour utilise, entre autre, une table de classification des faces, « Face Classification table » (FCT), construite en fonction des interactions détectées entre les faces des différents volumes des caractéristiques soumis à l'entrée du processus. Cette table est sous la forme d'une matrice de grandeur $n_f \times n_f$ ou n_f est le nombre total de faces dans le groupe de caractéristiques soumis à l'entrée. Il existe neuf types d'arrangements spatiaux relatifs entre les faces, voir la figure 31 ci-dessous.

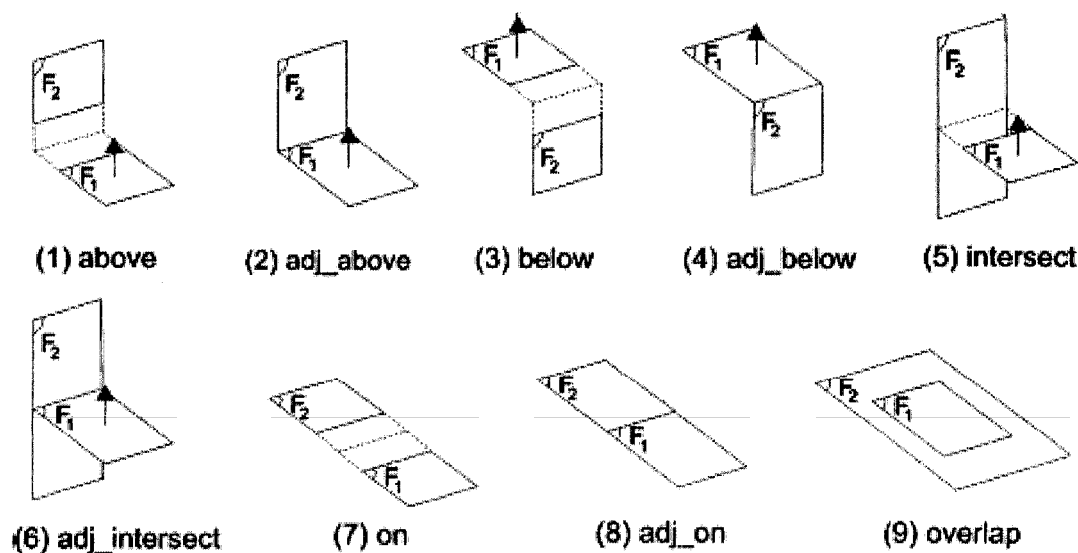


Figure 31 Types de classification des faces F_2 en fonction de F_1 [49]

Par la suite, un graphique des relations entre les caractéristiques, « Feature Relationship Graph » (FRG), dont les nœuds sont des caractéristiques et où une arrête entre les nœuds dénote une relation d'adjacence ou de dépendance entre deux nœuds, est construit. Ce graphique est emmagasiné sous la forme d'une matrice dont l'élément M_{ij} est soit 1 ou 0 si la caractéristique « i » est le père ou l'enfant de la caractéristique « j ». Une caractéristique est classifiée comme l'enfant d'une autre si l'une de ses faces a été répertoriée de type « overlap », voir la figure 31, dans le FCT.

2.5 Synthèse de la revue de littérature

Cette section est dédiée à la synthèse de la revue bibliographique permettant de situer nos travaux par rapport à l'existant.

Comme le démontre Ullman [13], Hyman [14] et Jenkinson [7] le processus de conception est au centre du succès des entreprises œuvrant dans le développement de produits. Il est donc pertinent de développer des outils permettant de raccourcir les cycles de conception, ce qui est l'un des objectifs de ce travail.

Hyman [14], Suh [15], Finger [16;17], Reymen [18;19] et Hale [23] proposent des modèles de processus de conception appliqués aux différents domaines de développement de produits. Il ne fait pas partie de ce travail de proposer un nouveau modèle, mais plutôt d'identifier celui qui est pertinent à nos travaux. Raymer [22] propose un modèle du processus de conception adopté généralement en aéronautique et montré à la figure 17, page 38. On distingue également sur cette figure que nos travaux s'adressent particulièrement aux phases conceptuelle, préliminaire et détaillée du processus de conception. Cependant, le typage des entités, que nous verrons dans le prochain chapitre, est effectué lors de la phase conceptuelle. La construction sommaire de l'assemblage des APRs comprenant entre autre le nombre d'éléments de structure, la forme de l'OML, l'espace requis dans la maquette numérique pour certaines pièces et systèmes sont définis lors de la phase de conception préliminaire. La définition des

poches d'allègement au sein des tôles de revêtement des APRs relève de la phase détaillée.

Kidd [26], Wright [28], Ullman [13] et Hyman [14] démontrent que les changements d'ingénierie, et surtout la façon dont ils sont propagés, sont des facteurs déterminant agissant sur les coûts de production d'un produit. Il a été établi que les coûts associés à une correction augmentaient d'un facteur 10 selon qu'elle survenait pendant la phase conceptuelle, préliminaire ou détaillé du processus de conception. Il a également été démontré que les changements étaient propagés plus rapidement à l'aide de systèmes informatiques plutôt qu'à la main. Toutefois, ces ouvrages ne proposent pas de méthodes précises pour la propagation des changements.

Giguère [3-5] a démontré qu'il était possible de propager automatiquement les changements tout en maintenant la cohérence des données au sein de la maquette en utilisant les outils de Catia V5 afin de détecter les changements lorsqu'ils surviennent. Ces travaux sont pertinents aux nôtres, car nous traitons des mêmes concepts que nous implantons sur la même plateforme, mais en utilisant de nouveaux outils disponibles sous Catia V5 R14.

Rappelons que durant l'année 2001, chez Bombardier Aéronautique, plus de 13000 changements d'ingénierie ont été demandés [2].

Cunnigham [32], Dixon [33], Bidarra [34;35], Deneux [36], Zimmermann [37;38] et Hummel et Brown [39] propose des définitions variées de ce qu'est une « caractéristique ». Pour ce travail, nous retenons celle de Deneux [36] définissant une caractéristique comme étant une unité d'information qui décrit une agrégation de propriétés d'un modèle produit pertinente à une vue spécifique du produit.

Les travaux de Shah & Mantyla [29] sont pertinents à ce travail, car les caractéristiques en sont à la base tant au niveau des concepts que des outils utilisés dans la plateforme de développement employée pour le prototype.

Deneux [40], dans son travail sur le « re-design », propose de représenter le savoir à l'aide d'un modèle composé de concepts et de relations entres-eux. Ces travaux sont pertinents et se rapprochent de nos concepts d'empreinte, de pseudo-empreinte et de liens technologiques capturant le savoir-faire. Toutefois, son approche est basée sur les fonctions établies au début de la conception et s'adresse aux conceptions subséquentes à un premier essai, le « re-design ». De plus, son modèle tient compte de l'imprécision de la connaissance, contrairement au notre dont le savoir-faire doit être précis et bien défini.

Fruchter [41] traite la connaissance au sein d'une entreprise de façon contextuelle et a développé des outils permettant de la capturer et de la stocker à l'intérieur d'une base de données accessible aux usagers. La connaissance est donc une « donnée » contenue dans un système autonome et distinct des outils CAO de l'entreprise. Nos travaux sont plutôt basés sur le développement d'outils CAO incorporant la connaissance et le savoir-faire de l'entreprise.

Les travaux de Zimmermann [37;38], portant sur les liens entre les différents « Engineering Object », se rapprochent des nôtres et des liens technologiques que nous utilisons pour capturer le savoir-faire d'ingénierie. Zimmermann met l'accent, toutefois, sur une structure et une taxonomie unique des liens assurant que toute les applications accédant les modèles obtiennent la même information. Ce projet met plutôt l'accent sur les associations entre caractéristiques et les gains de temps qu'ils permettent d'obtenir sur certaines tâches de modélisation.

Shah [43] et Hoffman [46] ont travaillé sur l'élaboration de langages et de mécanismes permettant de créer des caractéristiques spécifiques à l'utilisateur ou à un domaine en

particulier. Il ne fait pas partie des objectifs de ce travail de développer de tels outils, mais plutôt d'utiliser ces mécanismes présent dans la plateforme de développement que nous utilisons pour notre prototype d'application.

Lederman [48] propose une approche de conception qu'il nomme « parametric-associative design », approche de conception paramétrique et associative. Il utilise des UDFs et des scripts VB pour modifier automatiquement le nombre de lisses et de cadres dans une portion du fuselage d'un avion. Lederman se concentre sur l'arrimage des données circulant entre la plateforme CAO contenant la maquette numérique de l'avion et des calculs qui sont fait à l'aide d'un logiciel externe. Ces travaux sont pertinents aux nôtres en ce sens que nous utilisons également des UDFs et des programmes VB, seulement nous demeurons à l'intérieure de la plateforme CAO.

Tous ces auteurs s'intéressent à un aspect particulier, que se soit le processus de conception, la propagation des changements d'ingénierie, la capture et la gestion du savoir-faire, l'utilisation des UDFs pour créer des caractéristiques propres à l'utilisateur ou à un domaine en particulier, ainsi qu'à la création d'associations entre caractéristiques.

Toutefois, nous n'avons pas trouvé d'écrits traitant de l'ensemble de ces sujets dans un seul projet. C'est pourquoi nous pensons que le développement d'outils orientés métier, encapsulant le savoir-faire d'ingénierie et propageant automatiquement les changements (modifications et corrections) au sein d'une maquette numérique d'assemblage est pertinent et actuel, car ces outils permettent de diminuer le temps de modélisation, de réduire les cycles de développement et par conséquent permettent de diminuer les coûts de production. Ces outils permettent également d'améliorer la fiabilité du produit en rendant le savoir-faire accessible à un plus grand nombre d'utilisateur au sein d'une entreprise.

CHAPITRE 3

LES CONCEPTS FONDAMENTAUX À VALIDER

Ce chapitre décrit les concepts mis de l'avant pour atteindre les objectifs posés. Ces concepts seront validés par l'entremise du prototype d'application traitant de la conception des poches d'allègement contenues dans les APRs d'un avion. Nous débutons le chapitre en établissant la terminologie que nous utiliserons tout au long de ce travail. Par la suite, nous abordons la temporalité des associations, la cardinalité des associations, l'agrégation et la décomposition des associations ainsi que le concept de lien de dérivation. Finalement, nous traitons de concepts déjà mis de l'avant par quelques uns de nos collègues du LIPPS, tel que le typage des entités ainsi que les concepts d'empreinte et de pseudo-empreinte.

3.1 Terminologie

Cette section est dédiée à la description du sens que nous donnons à certains termes que nous utiliserons tout au long de ce travail.

3.1.1 Termes génériques de la CAO

Nous débutons par l'élaboration de termes génériques à la CAO de façon générale.

<i>Maquette numérique</i> ou « <i>Digital Mock-Up</i> »:	Représentation géométrique d'un produit, généralement en 3D, réalisée à l'aide d'outils CAO en vue de le visualiser, l'analyser, de le contrôler et d'en simuler certains comportements [50].
---	---

<i>Modèle Produit :</i>	Regroupement de fichiers comprenant les matériaux, les pièces, les composants, les sous-assemblages, les assemblages et de l'équipement. Le terme produit englobe également celui de famille de produits [51].
<i>Pièce ou composant :</i>	Correspond à un ou plusieurs items raccordés ensemble qui ne peuvent être démontés sans détériorer l'usage pour lequel la pièce a été conçue [51].
<i>Assemblage :</i>	Correspond à une combinaison de pièces qui, jointes ensemble, remplit une fonction spécifique et peut être désassemblée sans la dégradation d'aucune pièce (exemple : assemblage de ventilateur, amplificateur de fréquences audio, etc.) [51].
<i>Paramètre :</i>	Correspond à une donnée numérique utilisée dans la définition d'une géométrie. Ce peut être une distance, un angle, un nombre entier, etc.
<i>Entité ou élément géométrique :</i>	Correspond à un point, une courbe, une surface, un solide ou une combinaison de ceux-ci et sont définies selon les coordonnées tridimensionnelles de leur position [51].
<i>Entité topologique :</i>	Elles sont composées d'entités géométriques et/ou topologiques de complexité de bas niveau, tel que des arêtes, des faces ou des objets et exprime la connexion et l'ordre des relations [51].
<i>Opérations géométriques de bas niveau :</i>	Opérations effectuées sur des entités géométriques telles que la création d'un prisme à l'aide d'une esquisse (souvent nommé « extrusion »), l'union

	et/ou la soustraction de prismes, la délimitation d'un prisme à l'aide d'un plan ou d'une surface, etc.
<i>Opérations géométriques de haut niveau :</i>	Opérations effectuées sur une entité ou un ensemble d'entités géométriques à l'aide d'un autre ensemble d'entités (résultante d'opérations géométrique de bas niveau) ou à l'aide de caractéristiques. Ces opérations manipulent des caractéristiques ou des objets porteurs de savoir-faire d'ingénierie.
<i>Contexte de conception :</i>	Portion de la maquette numérique dans lequel le concepteur travaille et sur laquelle s'appuie la construction de nouvelles pièces [3].
<i>Caractéristique :</i>	Unité d'information qui décrit une agrégation de propriétés d'un modèle produit pertinente à une vue spécifique du produit [36].
<i>Encapsulation :</i>	En programmation orientée objet, cette opération consiste à regrouper, dans des entités distinctes (objets), les données et les procédures (méthodes) qui les manipulent [52].

3.1.2 Savoir-faire et associations

Un terme central à ce travail est celui du savoir-faire représenté par des associations entre objets. Nous réutiliserons des définitions proposées par Tremblay dans son travail sur la propagation des changements dans un contexte PLM [53-55].

<i>Savoir-faire :</i>	Le savoir-faire est un ensemble de connaissances en rapport avec une association qui permet d'agir
-----------------------	--

sur les objets associés. Un savoir-faire correspondant à une tâche spécialisée sera formalisé dans un lien associant deux objets techniques [53-55].

L'intégration du savoir-faire d'ingénierie spécifique à une tâche réalisée à l'aide de l'outil CAO permet d'homogénéiser les pièces, mais aussi de capturer l'intention de conception plutôt que le résultat de la conception. De plus, cela permet à une organisation de récupérer une partie des connaissances acquises par un concepteur durant ses années de service qui, autrement, seraient perdues lors de son départ.

Cette définition fait appel à deux autres termes que nous pensons également judicieux de préciser. Il s'agit d'*objet technique* et d'*association*. Nous reprenons également les définitions proposées par Tremblay [53-55].

Objet technique : Le terme objet technique englobe, de manière non-exhaustive, des éléments géométriques, des caractéristiques, des pièces, des assemblages, des documents, des articles, des fonctions et des tâches [53-55] (cette définition diffère de celle de Maurino [24]).

Association : Essentiellement, une association indique qu'il existe une dépendance entre deux objets techniques [53-55].

À leur tour, ces définitions font référence à d'autres termes que nous voulons spécifier afin d'être précis dans leur utilisation. Il s'agit des termes *relation*, *lien* et *contrainte*. Nous adaptons les définitions proposées par Tremblay [53-55].

Relation : Une relation établit une dépendance abstraite entre deux objets techniques. Une relation n'est pas suffisamment formalisée pour qu'un outil logiciel puisse l'utiliser pour propager un changement. Ainsi, l'interprétation et l'utilisation d'une relation nécessite une intervention humaine. Par exemple, une relation peut contenir la matrice de positionnement d'une pièce par rapport à une autre pièce. Si une pièce est modifiée, la relation n'est suffisante à elle seule pour propager automatiquement le changement et maintenir ainsi l'intention de conception et la cohérence du modèle.

Lien : Un lien définit une dépendance formelle entre différents objets techniques. Il représente un savoir-faire relatif à une tâche qui utilise un langage procédural ou une approche d'intelligence artificielle. Une tâche est définie comme étant une portion d'un travail à effectuer. Un lien peut être mis en application à l'aide d'outils logiciels et nécessiter occasionnellement l'intervention humaine. Le lien de dérivation décrit par Giguère [3-5], créant des soyages et une découpe à l'intersection d'un cadre et d'une lisse, est un exemple de lien spécifique capturant le savoir-faire basé sur une méthodologie propre au domaine aérospatial. Un lien peut être établi à l'intérieur d'un outil CAO (utilisant des modules de gestion de la connaissance ou d'UDFs comme dans Catia V5) ou à l'extérieur de l'outil en utilisant les APIs.

Contrainte : Une contrainte représente une dépendance formelle, spécifique et non décomposable entre différents objets techniques. Les contraintes sont gérées par l'application (dans notre cas un système CAO). L'application peut donner accès aux contraintes par le biais d'APIs, à l'aide de macros ou simplement par l'interface usager. Une contrainte de parallélisme établie entre deux droites constitue un bon exemple de ce type d'association. Conséquemment, les contraintes sont des dépendances indécomposables utilisées pour créer des liens.

Tremblay organise les termes *association*, *relation*, *lien* et *contrainte* en sous-ensembles. Ainsi, l'association contient les trois ensembles relation, lien et contrainte [53-55]. Il établit également un rapport entre le niveau d'abstraction et la formalisation du savoir-faire. Plus le savoir-faire est complexe, plus le niveau d'abstraction augmente. Le niveau d'abstraction est proportionnel à la complexité des objets associés. On peut prévoir que les objets associés par des relations sont des documents concernant par exemple une pièce, un outillage, etc. Les objets associés par des liens sont des caractéristiques, tels un soyage ou une découpe. Tandis que les objets associés par des contraintes sont de bas niveaux et portent un sens très précis, tels qu'une perpendicularité ou un parallélisme entre deux droites.

3.2 Temporalité des associations

Les associations ont également une temporalité qui peut être de deux types : persistante ou éphémère. Catia V4 fonctionne essentiellement à l'aide d'associations éphémères. Tandis que Catia V5 fonctionne à l'aide d'associations persistantes.

Une association est le terme général indiquant qu'il existe une dépendance entre deux objets techniques. La relation, le lien et la contrainte sont des cas particulier

d'association et le concept de temporalité s'applique également à eux. Nous reprenons les définitions d'association persistante et éphémère proposées par Tremblay [53-55].

3.2.1 Association persistante

Une association persistante a pour principale caractéristique d'être permanente. Seules la destruction ou la modification de l'association peuvent mettre un terme à la dépendance entre les différents objets techniques qu'elle associe. L'association persistante facilite la propagation d'un changement dans le cas d'une modification [53-55]. Par exemple, une contrainte de parallélisme établie de façon persistante entre deux droites maintient le parallélisme lorsqu'une des deux droites est modifiée.

Dans le travail de définition de soyages et de découpes effectué par Giguère [3-5], des liens technologiques persistant sont établis entre l'empreinte et la pseudo-empreinte. Lors d'un changement de position ou de géométrie de l'empreinte, les changements sont propagés à la pseudo-empreinte grâce à ces liens.

Dans le présent projet, la persistance des liens que nous établiront au sein du contexte d'assemblage jouera un rôle crucial. En effet, c'est grâce à cette persistance que pourra s'effectuer la propagation automatique des changements et que nous garantirons la cohérence des données entre les différents modèles de l'assemblage.

3.2.2 Association éphémère

Une association est dite éphémère lorsque la dépendance entre les différents objets techniques qu'elle associe existe seulement au moment de sa création. Seul le résultat de cette opération demeure et non le mécanisme ayant servi à sa création. Par exemple, une contrainte de parallélisme peut être établie de manière éphémère entre une droite existante et une droite en cours de création. En effet, la contrainte n'existe alors que pendant la création de la droite. Cette association ne protège donc pas la relation entre

les deux objets techniques après une modification puisque la contrainte n'existe plus [53-55].

Catia V4 est largement basé sur l'emploi de contraintes éphémères et, conséquemment, les changements doivent être propagés à la main entraînant des délais et des coûts lors de la modification de pièces.

3.3 Cardinalité des associations

Un autre concept mis de l'avant par Tremblay et que nous réutilisons dans ce travail traite de la cardinalité des associations.

La cardinalité d'une association se rapporte au nombre d'objets techniques qu'elle associe. Ainsi, une association peut compter m éléments de référence et n éléments cibles. Tremblay dénombre quatre cas spécifiques : 1×1 (cas 1), $m \times 1$ (cas 2), $1 \times n$ (cas 3) et $m \times n$ (cas 4) et les objets techniques constituant les nœuds d'une association 1×1 forment une paire [53].

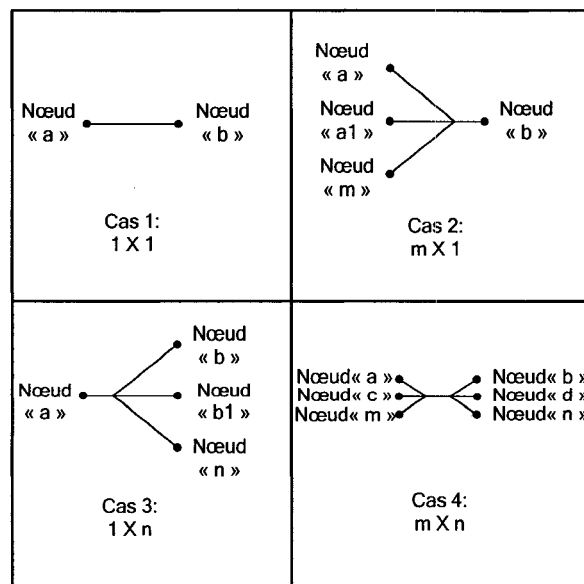


Figure 32 Quatre cas de cardinalité [53]

La cardinalité d'une association est relative au niveau de granularité considéré. En effet, selon le cas 4 de la figure 32, ce type de cardinalité peut représenter l'association de plusieurs paires d'objets techniques [53].

3.4 Agrégation et décomposition des associations

Le dernier concept développé par Tremblay [53] que nous retenons est celui d'agrégation et de décomposition des associations.

Une agrégation est le regroupement d'un ensemble d'associations ou d'objets techniques en une seule entité afin d'en faciliter la manipulation à un niveau d'abstraction plus élevé. Cette notion s'applique tant aux nœuds qu'aux associations. Tremblay utilise l'exemple de l'intention de conception d'un lien rainure-clavette pour illustrer ce concept, voir la figure 33 ci-dessous.

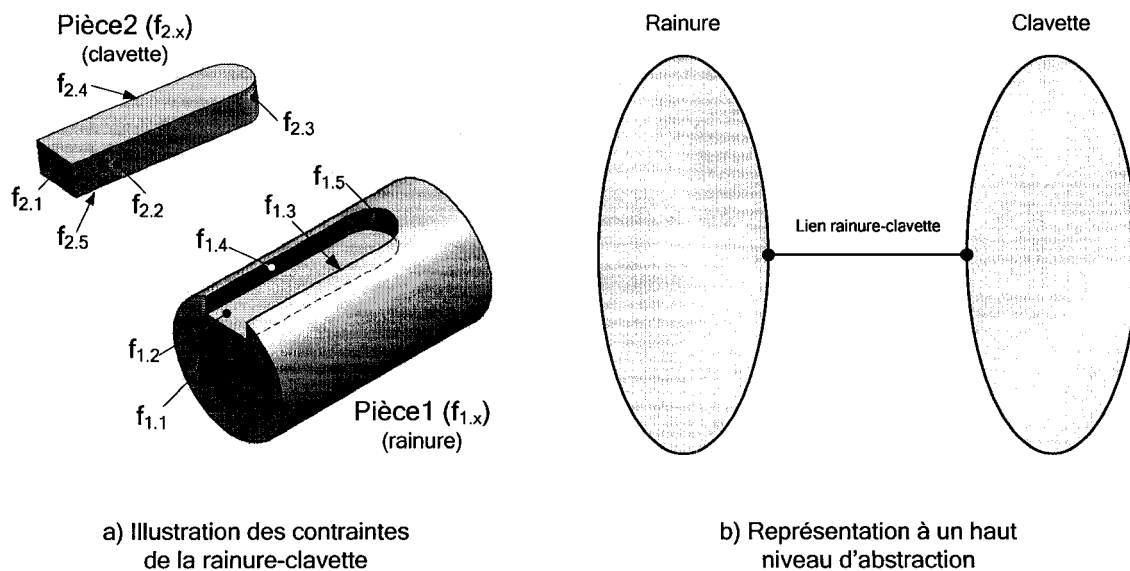


Figure 33 Exemple d'agrégation du lien rainure-clavette [53]

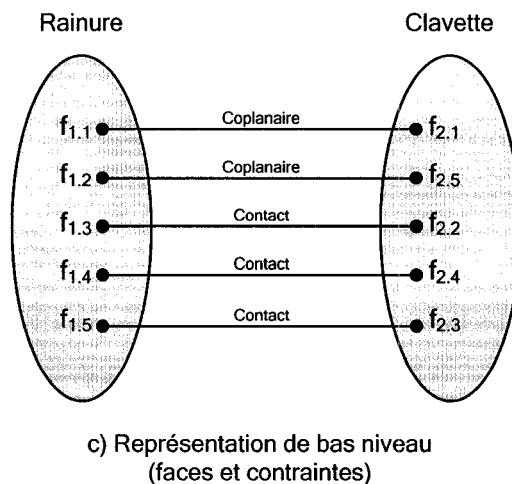


Figure 34 Représentation bas niveau du lien rainure-clavette [53]

Dans un outil CAO idéal, basé sur la conception par caractéristiques, un concepteur n'aurait qu'à définir le lien rainure-clavette à un haut niveau d'abstraction et l'application se chargerait d'établir les contraintes de contacts entre les différentes faces, voir figure 34.

Le concept de décomposition des associations correspond à effectuer le chemin inverse de l'agrégation, voir la figure 35.

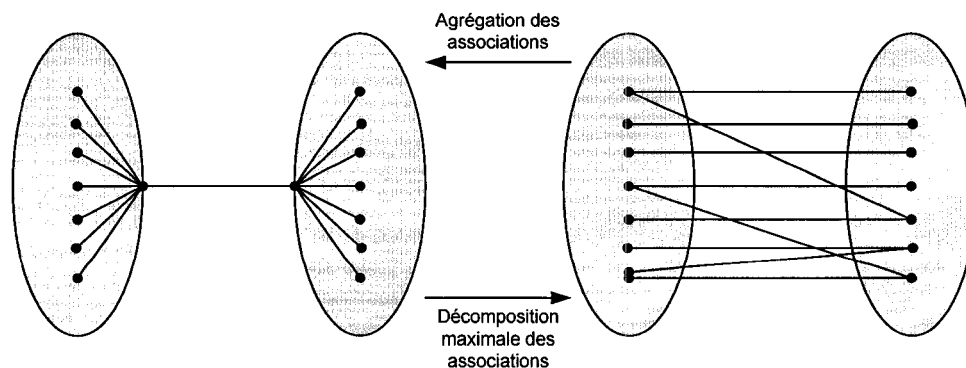


Figure 35 Exemple d'agrégation et de décomposition maximale au niveau des associations [53]

Tremblay [53] distingue deux types d'agrégations d'associations : l'agrégation d'associations parallèles, regroupant des associations parallèles en une seule unité, et l'agrégation d'associations séquentielles, regroupant une succession d'agrégation en une seule unité. Finalement, une agrégation peut être de type mixte.

3.5 Lien de dérivation

Giguère [3-5] a défini, dans son travail sur l'automatisation des soyages et des découpes à l'intersection d'un cadre et d'une lisse, le *lien de dérivation*, *l'empreinte*, *la pseudo-empreinte* et *la caractéristique contextuelle*. Nous reprenons l'essence de ses définitions pour les mettre en phase avec notre travail ainsi qu'avec celui de Tremblay [53-55].

<i>Lien de dérivation :</i>	Lien spécialisé qui exprime une dépendance unidirectionnelle d'une caractéristique cible envers une caractéristique de référence. Il joue un rôle essentiel dans la propagation des changements.
<i>Empreinte :</i>	Information propre à une caractéristique de référence déterminée par extraction à partir d'un groupe de pièces de référence [3-5].
<i>Pseudo-empreinte :</i>	Information propre à une caractéristique cible. Elle est déterminée automatiquement par l'application d'un savoir-faire propre à une tâche de conception à l'empreinte extraite du contexte de conception.
<i>Caractéristique contextuelle :</i>	Méta-caractéristique se composant d'une ou plusieurs caractéristiques de référence dont on extrait l'empreinte, d'un lien de dérivation capturant le savoir-faire propre à une tâche de conception, ainsi que d'une caractéristique cible dont on définit la pseudo-empreinte.

3.6 Synthèse sur les associations et les liens

Nous venons de définir différents aspects du concept d'association, dont le lien constitue une classe spécifique fondamentale à ce travail. L'organisation du concept implique la définition de plusieurs termes dont le sens usuel est similaire, mais auxquels nous attribuons, dans le contexte de ce travail, des nuances très importantes. Nous les avons regroupés dans un tableau pour mieux démontrer leurs différences.

Tableau IV

Synthèse du concept d'association (adapté de Tremblay [53-55])

Abstraction des associations	Formalisation du savoir-faire	Association	Objets Techniques impliqués	Savoir-Faire	Niveau d'abstraction	Outils
+	-	Relation	Documents, assemblages, pièces, etc.	Non-formalisé	Agrégation de liens	Applications automatiques
↑	↓	Lien	Caractéristiques	Formalisé, (tâches spécifiques)	Agrégation de contraintes	UDFs
-	+	Contrainte	Entités géométriques	Non-décomposable	-	Opération géométrique de bas niveau

Comme le démontre le tableau IV, la relation, le lien et la contrainte sont différentes classes d'associations. Également, comme l'indique la colonne «Abstraction des associations», une relation est plus abstraite qu'un lien et un lien est plus abstrait qu'une contrainte. Le niveau d'abstraction augmente donc lorsque l'on passe d'une contrainte à une relation.

Le terme « abstraction » est relatif au contenu « porté » par l'association et à son sens de façon générale. Par exemple, une contrainte de perpendicularité entre deux « entités géométriques » de type « lignes » est très concrète en ce qui concerne la signification de l'association, car elle « contraint » deux lignes à être perpendiculaires. De plus, comme l'indique la colonne «Savoir-faire», la connaissance portée par cette association ne peut

être décomposée à un plus bas niveau. Il est à noter que le niveau de formalisation du savoir-faire est à l'inverse de l'abstraction des associations. En effet, plus une association est concrète, plus le savoir-faire spécifique à la tâche est formalisé et précis.

Finalement, pour établir une association de type « contrainte », l'outil CAO doit pouvoir effectuer des opérations géométriques de bas niveau, ce que l'ensemble des plateformes CAO font actuellement.

Pour une association de type « lien », les objets techniques liés sont des caractéristiques et le savoir-faire nécessaire pour l'établir est formalisé. Par exemple, le lien de dérivation utilisé par Giguère [3-5], dans le cas des soyages et des découpes, nécessite la connaissance des règles utilisées pour accomplir cette tâche de modélisation spécifique. Le niveau d'abstraction du lien est moindre que celui de la relation et lorsque l'utilisateur manipule un lien, il manipule en fait une agrégation de contraintes. Comme nous le verrons un peu plus loin dans ce travail, l'un des outils CAO permettant la manipulation de liens est l'UDF (User-Defined Features).

Dans le cas d'une association de type « relation », les objets techniques reliés sont des documents, des assemblages, des pièces, etc. Ce type d'association est non-formalisé et bien qu'il nous informe sur l'existence d'une association entre deux pièces ou deux assemblages, par exemple, la relation en elle-même ne contient aucune connaissance formalisée permettant d'agir sur un des objets reliés en fonction de l'état de l'autre objet. Au niveau de l'abstraction de la relation, lorsque l'utilisateur manipule une relation, ce qu'il manipule pourrait éventuellement correspondre à une agrégation de liens. Finalement, les outils CAO capables de manipuler ce type d'association sont les applications automatiques développées à l'aide d'un langage procédural tel que les macros sous Catia V5 ainsi que les APIs VB ou C++ disponibles sous les différentes plateformes CAO.

3.7 Le typage des entités

Un des concepts fondamentaux au centre de ce travail est le *typage des entités*. Le typage des entités permet de leur associer des méthodes permettant à leur tour d'attribuer un comportement spécifique à une entité ou à un groupe d'entités géométriques dans un contexte donné. Ainsi, dans un certain contexte, un trou peut être pratiqué dans un solide pour définir l'alésage associé à un arbre, tandis que dans un autre contexte il est défini avec un filetage pour accueillir un boulon.

Dans notre cas d'application, les pièces de structure, telles que les lisses et les cadres, seront typées pour leur donner un sens dans le contexte et pour pouvoir porter des actions. Un cadre, par exemple, donne la forme au fuselage de l'avion et dans le contexte des poches d'allègement, définit la limite droite ou gauche d'une poche. De même, la lisse permet de rigidifier la structure de l'aéronef et dans notre cas définit la limite supérieure ou inférieure d'une poche. De plus, le typage des pièces de structure et de la caractéristique poche nous permettra de les trier et de les repérer rapidement dans les modèles.

Ce concept nous mène directement à un autre qui lui est très proche, celui d'approche de conception par caractéristiques.

3.8 Concept d'empreinte et pseudo-empreinte

Le concept central à ce travail est celui d'empreinte / pseudo-empreinte. Il nous supportera dans le processus de création d'une caractéristique de type «poche» dans le cadre du développement de notre prototype d'application. François Giguère a utilisé ce concept pour la première fois en 2001 lors du développement d'une application d'insertion et de mise à jour automatique de soyages et de découpes dans un cadre à l'intersection d'une lisse.

Globalement, l’empreinte et la pseudo-empreinte sont reliées par l’entremise d’un lien spécialisé, le lien de dérivation. La figure 36 représente, au niveau d’abstraction du lien, l’ensemble «Empreinte-Lien-Pseudo-empreinte» formant une «caractéristique contextuelle».

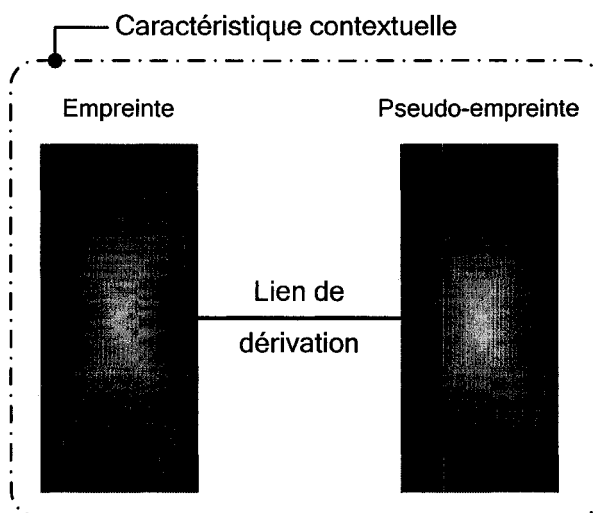


Figure 36 Agrégation du lien de dérivation de l'empreinte et de la pseudo-empreinte

L’empreinte est une «information» propre à une caractéristique de référence. Cette information peut être de différents types et est généralement obtenue par extraction à partir du contexte de conception. Il peut s’agir d’objets techniques, tels que des éléments géométriques, des pièces, des assemblages, des caractéristiques ou encore de nombres désignant un angle, une distance ou un numéro de pièce. Le type d’information constituant l’empreinte est large de telle sorte que ce concept peut s’appliquer à une multitude de cas de conception et de modélisation.

Dans le cas de l’application de Giguère, l’objectif visé est d’automatiser la définition des soyages et des découpes effectués à l’intersection cadre-lisse. Le point de départ de cette opération consiste à identifier la lisse, représentant la pièce de référence dans le processus, à l’endroit où la découpe dans le cadre doit être faite. C’est à partir de la lisse que l’empreinte est ensuite extraite. Dans ce cas-ci, elle se compose des lignes formant

le profil de la lisse à l'intersection du cadre et Giguère utilise un plan pour l'obtenir. Une fois extraite, l'information constituant l'empreinte appartient à une caractéristique de référence et un lien est établie entre la lisse et la caractéristique de référence, voir la figure 37.

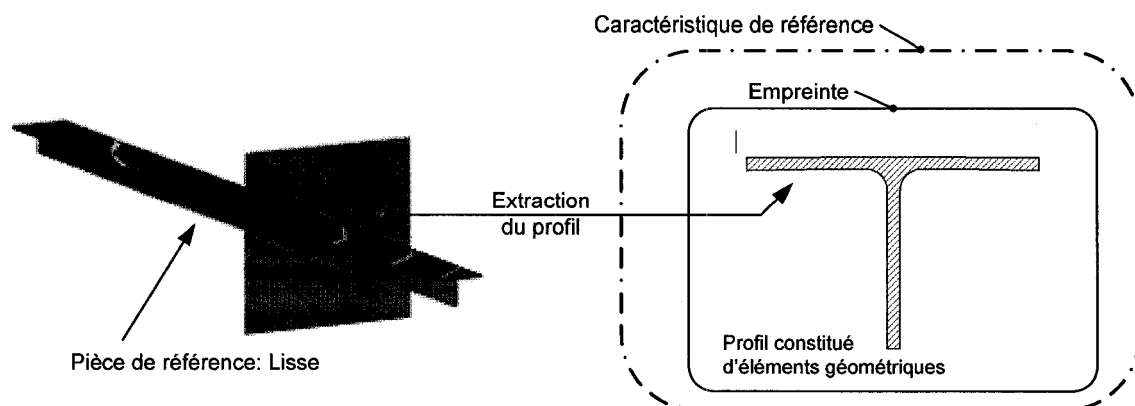


Figure 37 Extraction de l'empreinte dans le cas de la découpe et du soyage (adapté de Giguère [3])

La pseudo-empreinte est une «information» propre à la caractéristique cible. Elle est obtenue par l'application de savoir-faire d'ingénierie propre à une tâche, dans ce cas-ci, la modélisation de la découpe et des soyages dans le cadre. En appliquant les règles de conception formelles à l'empreinte, un nouveau profil est dessiné dans lequel est contenu la découpe et les soyages. Tout comme l'empreinte, la pseudo-empreinte peut être constituée de différents types d'informations, dans ce cas-ci des éléments géométriques, et appartient à une caractéristique cible, voir la figure 38.

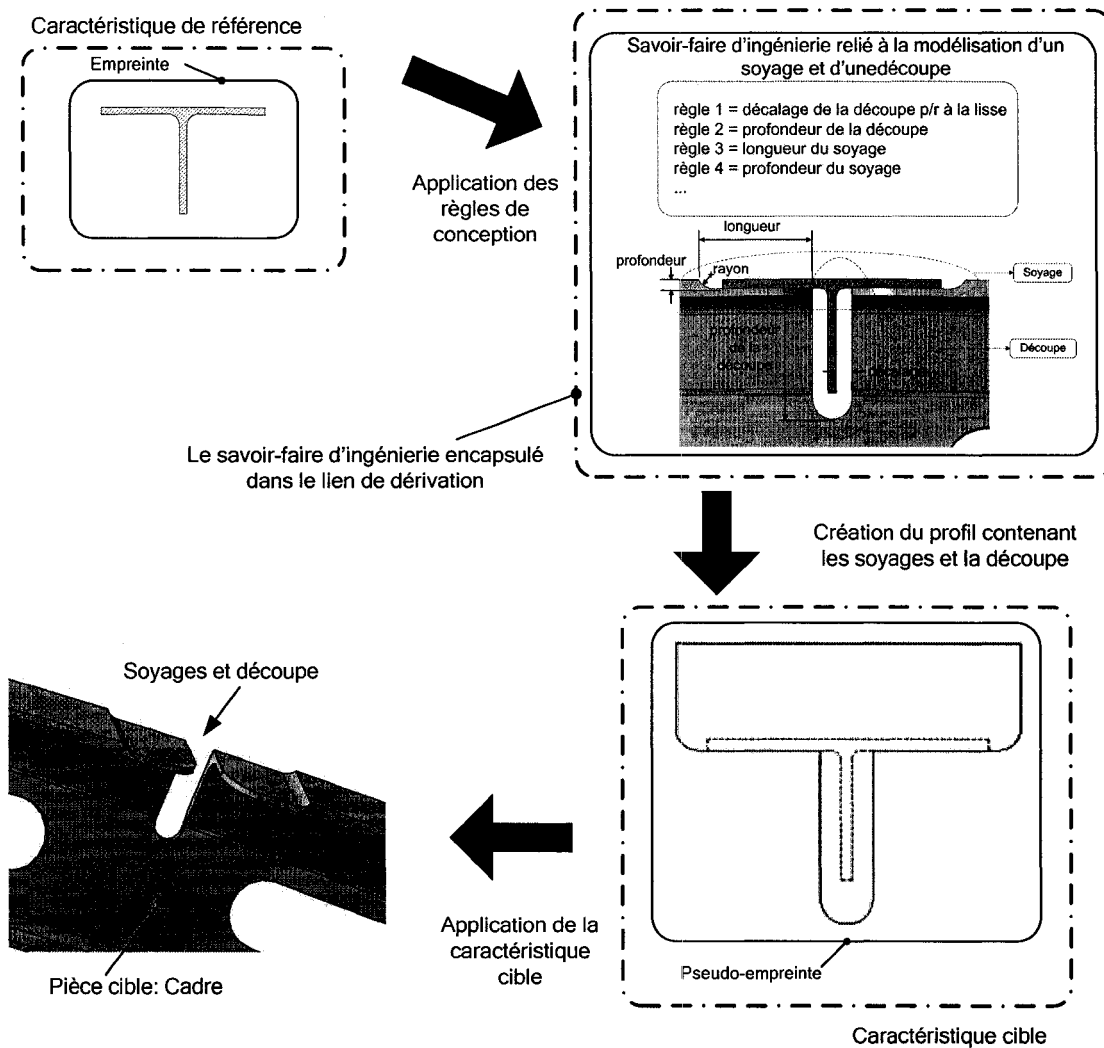


Figure 38 Création de la pseudo-empreinte et application de la caractéristique cible (adapté de Giguère [3])

Finalement, la caractéristique cible est appliquée à la pièce cible, dans ce cas-ci, le cadre nécessitant une découpe à l'intersection de la lisse. Le résultat est une adaptation du cadre afin que la lisse puisse s'y assembler correctement, soit les soyages et la découpe, voir la figure 38.

Le lien spécialisé de dérivation associe l'empreinte et la pseudo-empreinte à l'aide des règles de conception qu'il intègre. L'information transigeant par ce lien est

unidirectionnelle et lorsqu'il y aura des corrections et/ou des modifications au sein de l'assemblage ou des pièces le constituant, l'extraction, l'application des règles de conception et l'application de la caractéristique cible à la pièce cible seront «rejoués» afin de propager les changements aux pièces impactées.

Le processus d'application du concept d'empreinte/pseudo-empreinte peut donc se résumer à quatre étapes distinctes :

- Extraction de l'empreinte à partir «d'objets» de références contenues dans le contexte de conception et formation de la caractéristique de référence,
- Application à l'empreinte de savoir-faire propre à une tâche de modélisation,
- Formation de la pseudo-empreinte et de la caractéristique cible,
- Application de la caractéristique cible à des «objets» contenue dans le contexte de conception.

Dans le cas de Giguère, l'empreinte contenue dans la caractéristique de référence utilisait une seule pièce, une lisse. La caractéristique cible, contenant la pseudo-empreinte, s'appliquait également à une seule pièce, le cadre. Au niveau de la relation entre la pièce lisse et la pièce cadre, la cardinalité correspond au cas 1 décrit précédemment à la section 3.3, i.e. 1 X 1 (lisse X cadre), voir la section «a) Relation lisse-cadre» de la figure 39. Cette relation se décompose en deux liens, «SectionLisse-Soyages» et «SectionLisse-Découpe», qui associent la caractéristique de référence à la caractéristique cible. La portion «SectionLisse» des liens correspond au profil de la lisse, extrait à l'intersection du cadre, qui est contenue dans l'empreinte et forme la caractéristique de référence. Les portions «Soyages» et «Découpe» correspondent à certaines parties formant le profil de la pseudo-empreinte constituant la caractéristique

cible, voir la section «b) Décomposition de la relation lisse-cadre en liens» de la figure 39. La cardinalité de chacun de ces liens est également de 1 X 1.

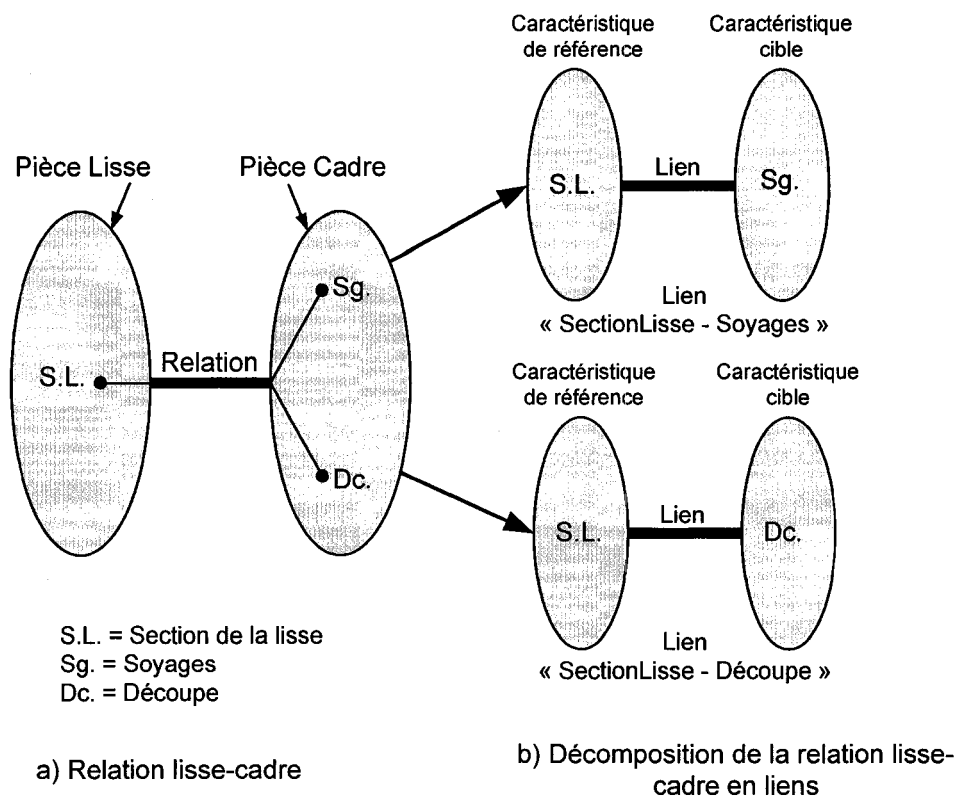
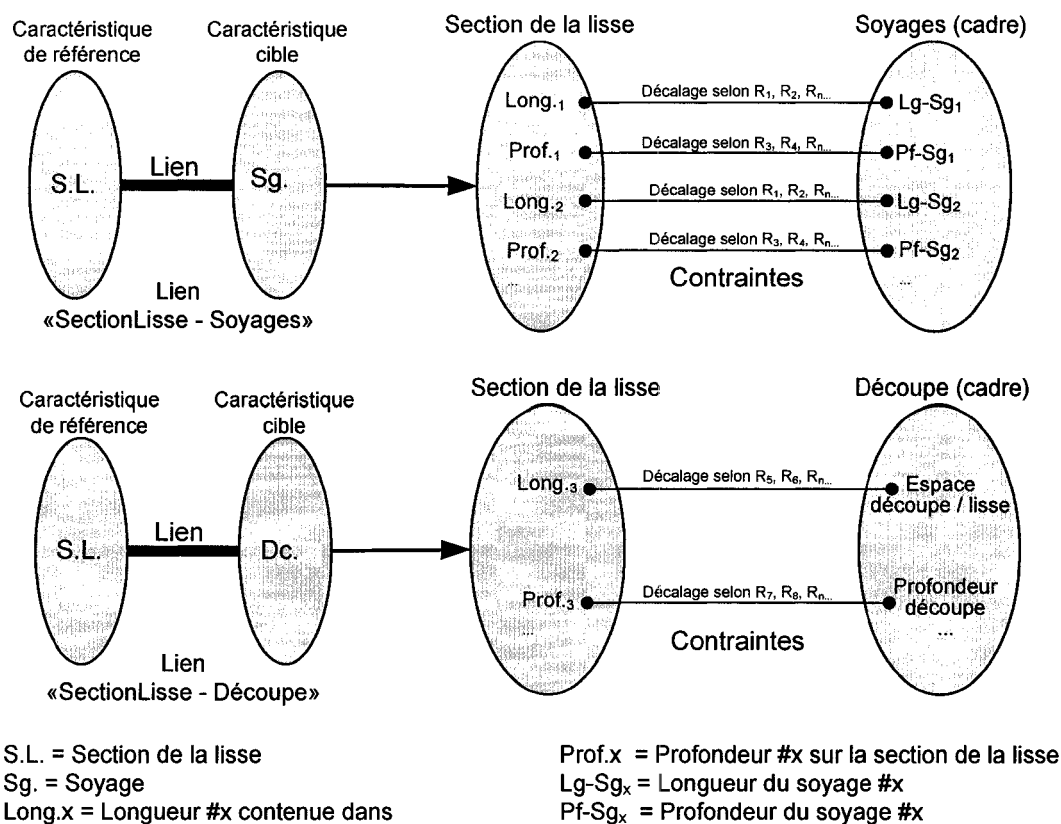


Figure 39 Relation et liens associant la lisse et le cadre

Finalement, ces deux liens se décomposent à leur tour en contraintes associant les différentes entités géométriques contenues dans la section de la lisse et celles contenues dans le profil de la pseudo-empreinte. Ce dernier est utilisé afin de modéliser les soyages et la découpe dans le cadre, voir les sections a) et b) de la figure 40.



a) Liens «SectionLisse – Soyages»
 et «SectionLisse - Découpe»

b) Décomposition des liens en contraintes

Figure 40 Liens et contraintes associant la lisse et le cadre

3.8.1 Concept d’empreinte et pseudo-empreinte dans le cas des poches

Comme mentionné précédemment, nous voulons faire un pas de plus pour démontrer le caractère générique de ce concept en l’appliquant à un cas de complexité plus élevé. Dans notre contexte de poches d’allègement, l’empreinte est déterminée conjointement par quatre pièces de référence : les deux cadres et les deux lisses entourant la poche à créer.

Le lien technologique de dérivation est formalisé par les règles de conception des poches (elles sont détaillées à la section 4.4) que l’on applique à l’empreinte pour obtenir une

pseudo-empreinte. À son tour, la pseudo-empreinte est appliquée à une seule caractéristique cible, soit la poche à créer dans la tôle de revêtement. La figure 41, schématise les quatre étapes d'application du concept dans le contexte de création d'une poche d'allègement.

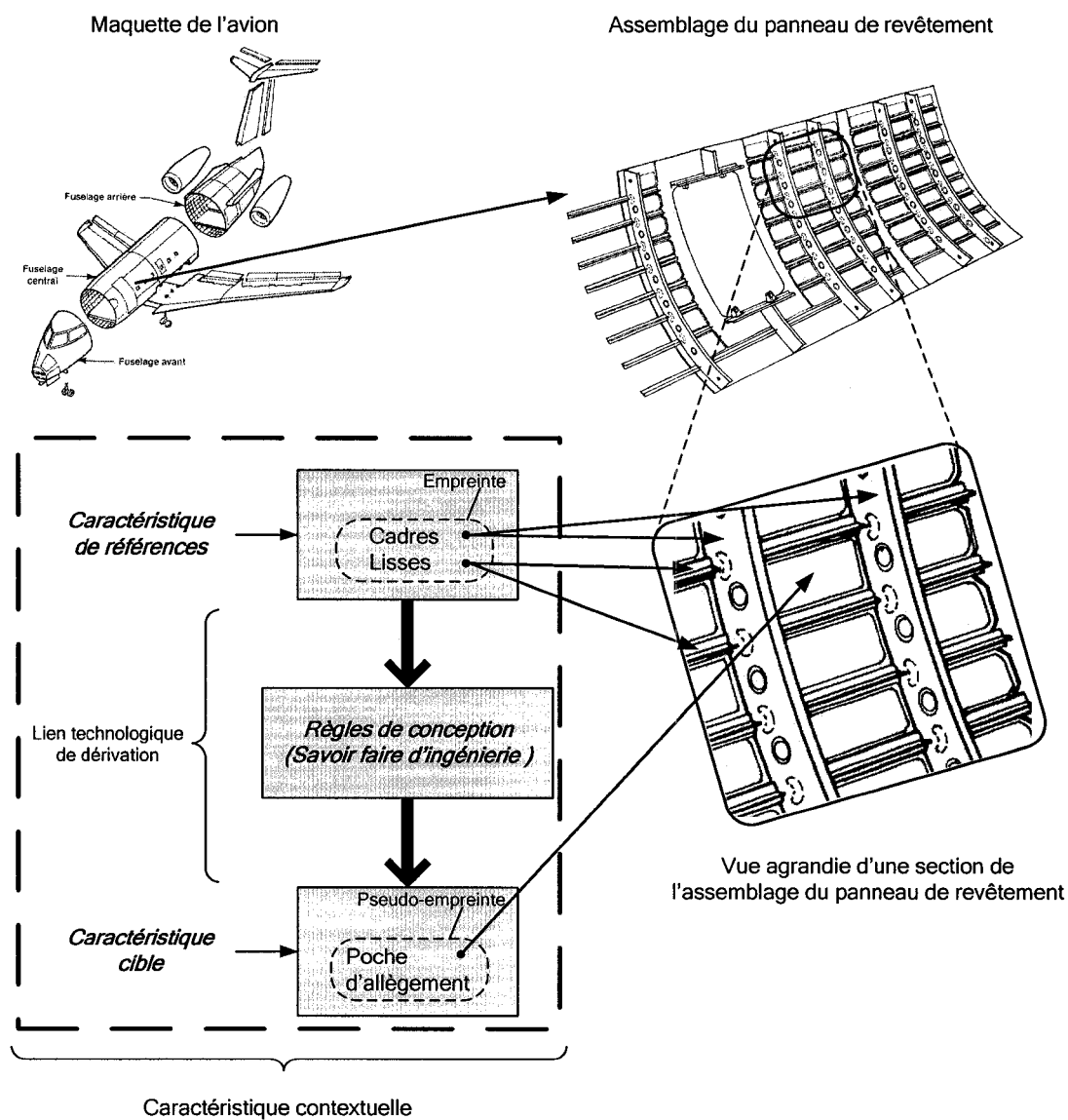


Figure 41 Exemple de caractéristique contextuelle menant à la définition d'une poche d'allègement

Le niveau de cardinalité de cette relation, impliquant cinq pièces (la tôle, deux lisses et deux cadres), correspond au cas 2 de la figure 32, $m \times 1$ où $m > 1$. Le lien technologique de dérivation que nous utilisons est également unidirectionnel, mais cette fois ci, de six caractéristiques appartenant à cinq pièces de référence (la tôle, deux lisses et deux cadres) vers une caractéristique cible appartenant à la pièce cible ; la cardinalité du lien est donc de 6×1 au niveau de l'abstraction des caractéristiques.

Pour une poche donnée, le lien de dérivation est une agrégation de contraintes entre des objets techniques de bas niveaux appartenant aux caractéristiques pertinentes des deux cadres, deux lisses, l'épaisseur nominale de la tôle et l'OML (informations tirées du contexte d'assemblage), d'une part, et les objets techniques formant le profil et le fond de la poche d'allègement, d'autre part, (figure 42).

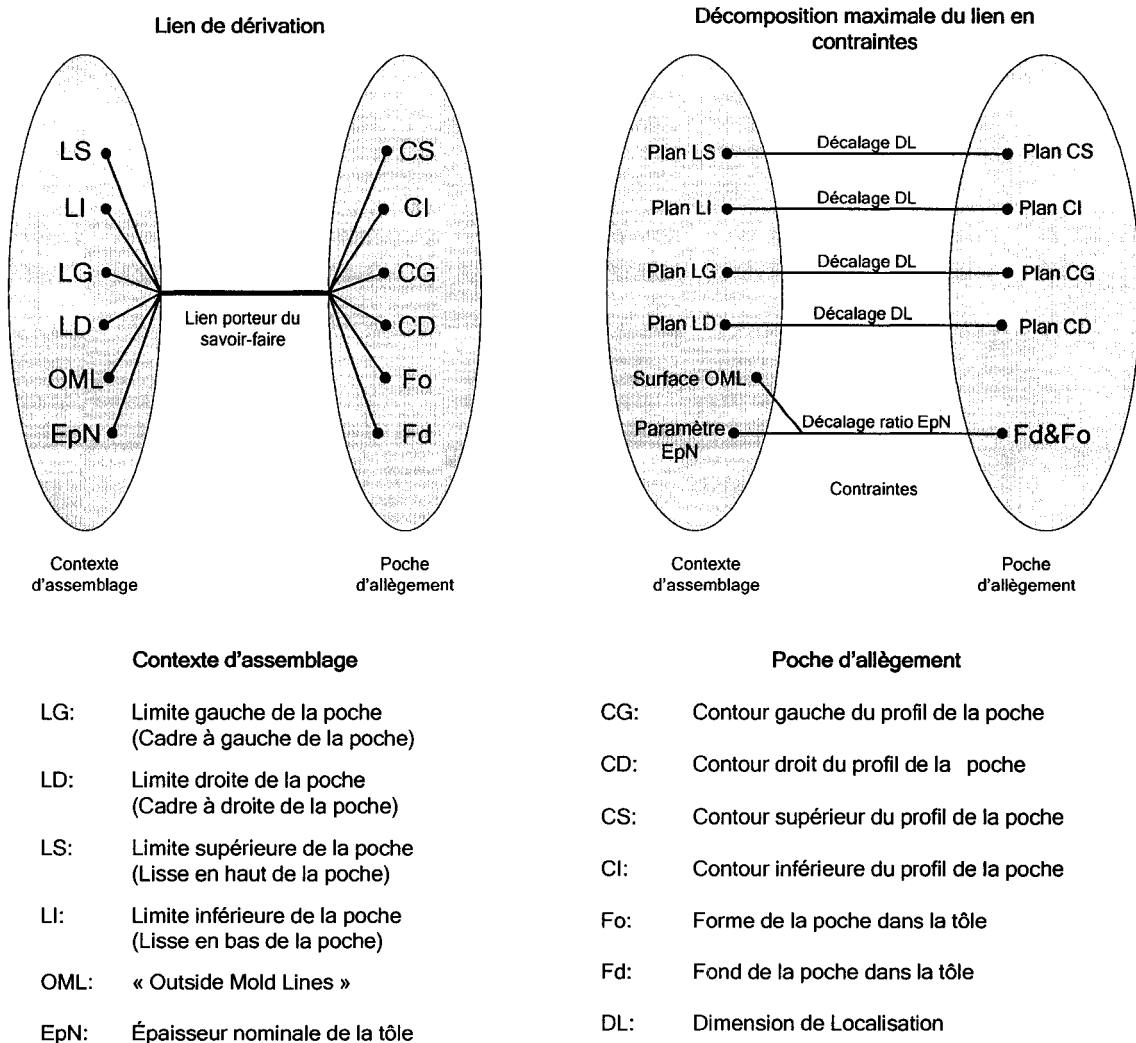


Figure 42 Lien de dérivation de la poche et sa décomposition maximale en contraintes

3.8.2 Différents contextes d'application du concept d'empreinte et pseudo empreinte

Dans le cas de l'application de soyages et de découpes de Giguère ainsi que dans celui de notre prototype d'application de définition des poches, les informations tirées du contexte de conception proviennent de pièces et les objets techniques constituant l'empreinte sont des éléments géométriques. Pareillement, l'application de la pseudo-empreinte est faite dans les deux cas à des pièces comprises dans un assemblage.

L'application du concept d'empreinte / pseudo-empreinte ne se limite pas qu'à ce cas de figure.

Prenons le cas d'application de Lederman [48] qui utilise des UDFs et des scripts VB pour travailler sur le poids de l'avion. Le point de départ est un certain état de la maquette numérique d'une partie du fuselage de l'avion comprenant un nombre de lisses et de cadres. L'empreinte se formerait de ce nombre et possiblement inclurait l'assemblage. L'application du savoir-faire serait intégrée dans les règles régissant l'analyse par éléments finis faite sur l'assemblage afin d'optimiser son poids. La pseudo-empreinte se formerait de l'assemblage contenant le nouveau nombre de lisses et de cadres automatiquement ajoutés ou enlevés à l'aide des scripts.

Dans le cas d'application de Michaud [6] qui concerne la conception d'un gabarit de découpe utilisé dans le processus de fabrication des poches d'allègement, le concept d'empreinte / pseudo-empreinte s'applique également. L'objectif est d'obtenir un modèle numérique tridimensionnel du gabarit dont les ouvertures correspondent aux poches d'allègement devant être usinées dans la tôle de revêtement des APRs. Dans ce processus, l'empreinte est formée de la surface IML de l'APR comprenant les ouvertures des poches. L'application du savoir-faire d'ingénierie propre à la modélisation de gabarits de découpe est faite sur l'empreinte afin de créer la pseudo-empreinte, correspondant, dans ce cas-ci, à une nouvelle surface. Finalement, l'application de la pseudo-empreinte correspond à épaissir la surface obtenue afin de créer le modèle du gabarit de découpe. Lors de corrections et/ou de modifications apportées à l'APR, les changements sont automatiquement propagés au gabarit de découpe des poches.

Finalement, le concept d'empreinte / pseudo-empreinte peut s'appliquer dans le processus de gestion des modifications. En effet, lorsqu'une demande de changement est faite (ECR), plusieurs documents doivent être mis à jour. Certaines informations, comme par exemple les numéros de version des pièces impliquées dans le changement,

doivent être modifiés. Dans ce cas-ci, l’empreinte constituerait le numéro de la pièce. L’application du savoir-faire sur l’empreinte, constitué de différentes règles à suivre lorsqu’une pièce évolue, donnerait naissance à un nouveau numéro qui formerait la pseudo-empreinte. Celle-ci s’appliquerait à différentes cibles, soit les documents contenant le numéro de la pièce devant être mis à jour. La cardinalité de la relation entre le modèle de la pièce avant la modification et les différents documents devant être mis à jour est de $1 \times n$.

3.9 Synthèse générale

Comme nous l’avons mentionné précédemment, une caractéristique est définie comme étant une unité d’information qui décrit une agrégation de propriétés d’un modèle produit pertinente à l’une de ses vues spécifiques. Les gammistes, par exemple, ne voient pas une pièce de la même façon qu’un concepteur ou qu’une personne pratiquant des analyses par éléments finis. Dans ce travail, nous adoptons le point de vue de la conception.

La conception par caractéristiques permet de définir des pièces et des assemblages mécaniques au moyen d’objets liés à l’intention de conception. Par exemple, lorsqu’un concepteur utilise la fonction trou et qu’il lui définit une profondeur, un diamètre, un type de lamage et un filetage, on comprend que l’intention du concepteur est de créer un trou dans le but d’accueillir un boulon correspondant au trou. L’interface de l’outil CAO avec laquelle le concepteur interagit est spécifique à l’intention de définir un trou et lui permet d’ajuster ces différents paramètres. Une fois que les informations nécessaires sont saisies dans l’outil CAO, ce dernier se charge de créer la géométrie et d’effectuer les opérations nécessaires à la création de la caractéristique « trou ». Cette approche est globalement plus rapide, enrichit le modèle sur l’intention de conception et accélère le traitement des changements, car il suffit de modifier les paramètres de la caractéristique au lieu d’exécuter des opérations géométriques de bas niveau.

Dans notre cas d'application impliquant des poches d'allègement, nous commençons par personnaliser l'outil CAO en définissant une caractéristique de type « *poche* ». Celle-ci contiendra toute la géométrie nécessaire à sa définition ainsi que le tissu d'associations permettant de la lier au contexte d'assemblage, i.e. au quadrilatère formé des pièces de structure qui l'entoure et à la tôle de revêtement. Les associations établies au sein de la caractéristique seront porteuses du savoir-faire d'ingénierie relatif à la conception des poches sous forme de paramètres agissant sur les différentes contraintes établies à l'intérieur même de la caractéristique et qui pourront être modifiés ultérieurement afin que l'ont puisse contrôler la forme et le profil de la poche.

Lors de l'instanciation d'une caractéristique poche, le concepteur n'aura qu'à spécifier les pièces de structure reliées à la poche et à évaluer les paramètres requis et l'application se chargera d'insérer la nouvelle poche à l'endroit et selon les critères spécifiés par le concepteur. De plus, il sera toujours possible au concepteur de modifier ultérieurement les paramètres régissant la poche.

L'utilisation de cette approche permet de libérer le concepteur des manipulations et des opérations géométriques ennuyantes, répétitives et dépourvues de créativité actuellement nécessaires à la création de chaque poche.

Par la suite, nous avons développé une application orientée métier automatisant complètement l'insertion et la mise à jour automatique des poches d'allègement dans les APRs du fuselage d'un avion. Cette application permet de valider et de faire un pas de plus vers la généralisation des concepts en instanciant de façon automatique une série de liens au sein de l'APR, là où au départ seules des relations sont présentes, tel que décrit au chapitre 4.

de typage, d'empreinte et de pseudo-empreinte. De plus, ce travail applique certaines définitions proposées par Tremblay [53-55] et permettent de les valider.

CHAPITRE 4

PROTOTYPE DE VALIDATION

Le chapitre 1, mise en contexte du cas d'application, nous a permis de bien comprendre le contexte de conception des poches d'allègement et de formaliser la démarche de leur création actuelle en quatorze étapes distinctes sous Catia V4. Le chapitre 2 nous a permis de situer l'ensemble de ce travail par rapport à l'existant. Le chapitre 3 détaille les concepts que nous voulons valider à l'aide du cas d'application. Le présent chapitre est consacré à la description du processus de personnalisation de l'outil CAO ainsi qu'au développement du prototype d'application automatisant la tâche de modélisation des poches. Il est à noter que l'ordre dans lequel les sections se succèdent suit le plus fidèlement possible les étapes que nous avons franchies pour réaliser le prototype.

4.1 Choix du contexte et objectifs spécifiques du prototype

La définition (terme que nous utiliserons pour désigner la création, l'insertion et la mise à jour) des poches d'allègement forme un cas d'application intéressant à solutionner pour plusieurs raisons.

Tout d'abord, ce cas d'application correspond à une réalité industrielle. Deuxièmement, il constitue un bon médium pour tester les concepts présentés au chapitre 3. En troisième lieu, le gain de temps promis par la personnalisation de l'outil CAO ainsi qu'à l'automatisation de ce processus manuel est considérable. Il est fonction du nombre total de poches à insérer et dans un avion de transport régional, tel que le CRJ-700 (un appareil à 70 places de Bombardier) par exemple, ce nombre est estimé à 2 500 [8]. Selon l'outil CAO et la méthodologie de conception vues au chapitre 1, cela fait un total de 35 000 étapes que les concepteurs doivent effectuer pour créer l'ensemble des poches et ce chiffre ne tient pas compte des changements d'ingénierie.

Nous avons donc développé une application spécialisée, orientée métier, qui automatise la définition des poches d'allègement. Les sept objectifs spécifiques suivant sont visés :

1. L'application doit pouvoir créer des poches d'allègement pour les APRs à simple et double courbures ;
2. L'application doit pouvoir créer une poche d'allègement en mode pas à pas afin que le concepteur puis insérer une seule poche en indiquant seulement les éléments de références de la poche à insérer ;
3. L'application doit pouvoir créer automatiquement toutes les poches contenues dans un APR sans l'intervention de l'utilisateur ;
4. L'application doit pouvoir propager automatiquement toute correction et/ou modification apportés à l'assemblage et ayant un impact sur les poches ;
5. L'application doit intégrer le savoir-faire d'ingénierie lié à la conception des poches d'allègement dans les APRs et en donner l'accès au concepteur ;
6. L'application doit utiliser au maximum les fonctions standards de l'outil CAO utilisé comme plateforme de développement ;
7. L'application doit utiliser le minimum de portions de codes (Visual Basic ou C++).

Ces objectifs permettent de circonscrire le domaine d'opération de notre prototype d'application. C'est à l'aide de ces lignes directrices et de cet état d'esprit que le développement de ce dernier fut abordé et réalisé.

4.2 Terminologie liée au prototype

Cette section définit les différents termes et définitions s'appliquant au contexte de conception des poches d'allègement dans les APRs que nous utiliserons tout au long de ce chapitre.

<i>Contour d'une poche :</i>	Correspond au périmètre du prisme utilisé pour creuser la tôle et créer une poche.
<i>Forme de la poche :</i>	Correspond à la face du prisme de la poche conditionnée par l'OML.
<i>Position de la poche :</i>	Correspond à l'emplacement de la poche dans la tôle. Elle est directement rattachée aux pièces de références.
<i>Limite supérieure de la poche:</i>	Correspond à la face plane de la lisse définissant la limite supérieure à laquelle s'applique la dimension de localisation du contour de la poche.
<i>Limite inférieure de la poche :</i>	Correspond à la face plane de la lisse définissant la limite inférieure à laquelle s'applique la dimension de localisation du contour de la poche.
<i>Limite gauche de la poche :</i>	Correspond à la face du cadre définissant la limite gauche à laquelle s'applique la dimension de localisation du contour d'une poche.
<i>Limite droite de la poche :</i>	Correspond à la face du cadre définissant la limite droite à laquelle s'applique la dimension de localisation du contour d'une poche.

Lors de notre séjour chez Bombardier [8], nous avons établi trois actions spécifiques liées à la définition des poches d'allègement : *la création ou l'insertion d'une poche, la destruction d'une poche et la mise à jour d'une poche*. Nous les avons formalisées selon les définitions suivantes :

<i>Création ou insertion d'une poche :</i>	Processus consistant à créer une poche d'allègement dans la tôle de revêtement et dont la position est fonction des pièces de référence.
<i>Destruction d'une poche :</i>	Opération qui consiste à effacer une poche de la tôle de revêtement.
<i>Mise à jour d'une poche :</i>	Processus adaptant la poche au contexte d'assemblage suite à la modification d'un(des) lien(s) ou paramètres de la poche.

4.3 Granularité des objets considérés par l'application

L'APR est représenté à l'aide d'une maquette numérique d'un assemblage composé des modèles contenant les lisses, les cadres et la tôle de revêtement. L'assemblage possède une structure hiérarchique qui peut être schématisée afin de situer le niveau d'opération auquel s'adresse notre application. Dans le même ordre d'idée, nous situons également l'application de François Giguère [3-5], consistant à automatiser la création et la mise à jour des soyages et des découpes d'un cadre, qui est similaire à la nôtre.

Tel que le démontre la figure 43 ci-dessous, l'application que nous avons développée agit à l'intérieur même du modèle numérique de la tôle de revêtement. En comparaison, celle de Giguère [3-5] opère à l'intérieur du modèle contenant le cadre. Ces deux applications opèrent au niveau des caractéristiques.

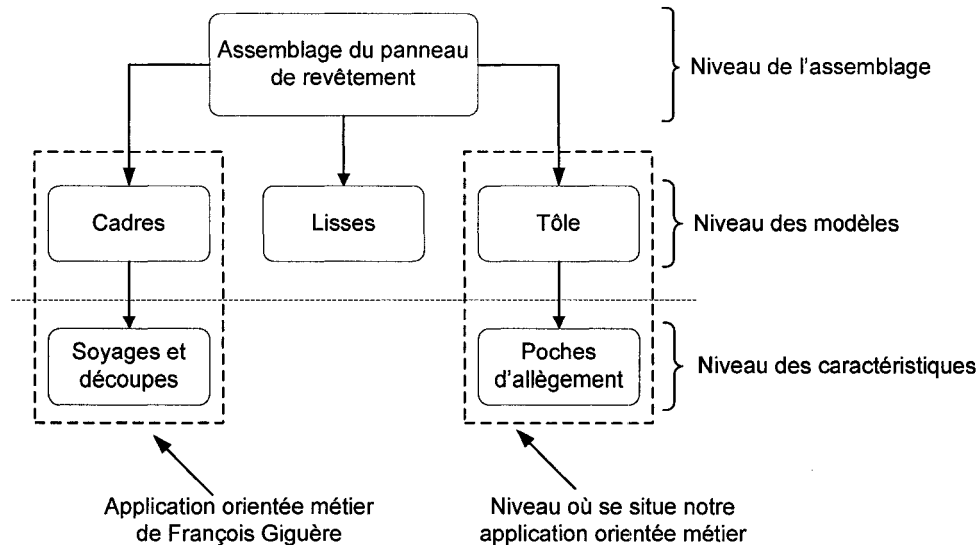


Figure 43 Granularité considérée par l'application orientée métier des poches d'allègement

4.4 Règles de conception de la caractéristique « poche » du prototype

Après avoir formalisée la démarche de création des poches sous Catia V4 au chapitre 1, nous dégagons cinq règles nécessaires à la définition d'une poche (se référer à la figure 15 de la section 1.6):

1. La dimension de localisation situe le contour de la poche entre les pièces de référence qui l'entoure. Valeur par défaut : 0.100" ;
2. Le rayon typique des congés de raccordement du contour est de 0.500" ;
3. La surface maîtresse OML conditionne la forme de la poche ;
4. L'épaisseur nominale de la tôle de revêtement se situe entre 0.060" et 0.120". Valeur par défaut : 0.100" ;
5. La profondeur de la poche dans la tôle correspond à une fraction de son épaisseur nominale, typiquement 1/3.

Nous voulons créer une caractéristique contextuelle de type « poche » et, pour ce faire, les règles établies précédemment sont encapsulées sous forme de propriétés dans la caractéristique.

Ces propriétés correspondent au savoir-faire d'ingénierie associé aux poches d'allègements qui permettent d'obtenir la pseudo-empreinte à partir de l'empreinte. Les propriétés se divisent en deux catégories suivant l'approche de Shah et Mäntylä [29]; il y a les propriétés intrinsèques et extrinsèques de la caractéristique « poche ».

La dimension de localisation, le rayon des congés de raccordement du contour et la profondeur de la poche dans la tôle sont des propriétés intrinsèques à la caractéristique poche. Elles sont accessibles à l'utilisateur pour qu'il puisse contrôler son comportement et indépendantes des autres caractéristiques du modèle.

Les deux lisses et les deux cadres entourant la poche, l'OML ainsi que l'épaisseur nominale de la tôle sont des éléments contextuels qui alimentent des propriétés extrinsèques de la caractéristique poche, i.e. la limite supérieure, inférieure, droite et gauche de la poche, sa forme et le fond de la poche. Ces éléments doivent être explicitement pointés lorsqu'une nouvelle caractéristique poche est instanciée.

La figure 44 à la page suivante, démontre les paramètres et les contextuels de la caractéristique poche. Il est à noter que la figure 44 est établie selon un observateur regardant vers l'APR à partir de l'intérieur du fuselage. De plus, *les arrondies de fond de poche* ne sont pas modélisés dans notre application car ils sont laissés à la discrétion du concepteur, d'une part, et que cela facilitait le développement de notre application, d'autre part.

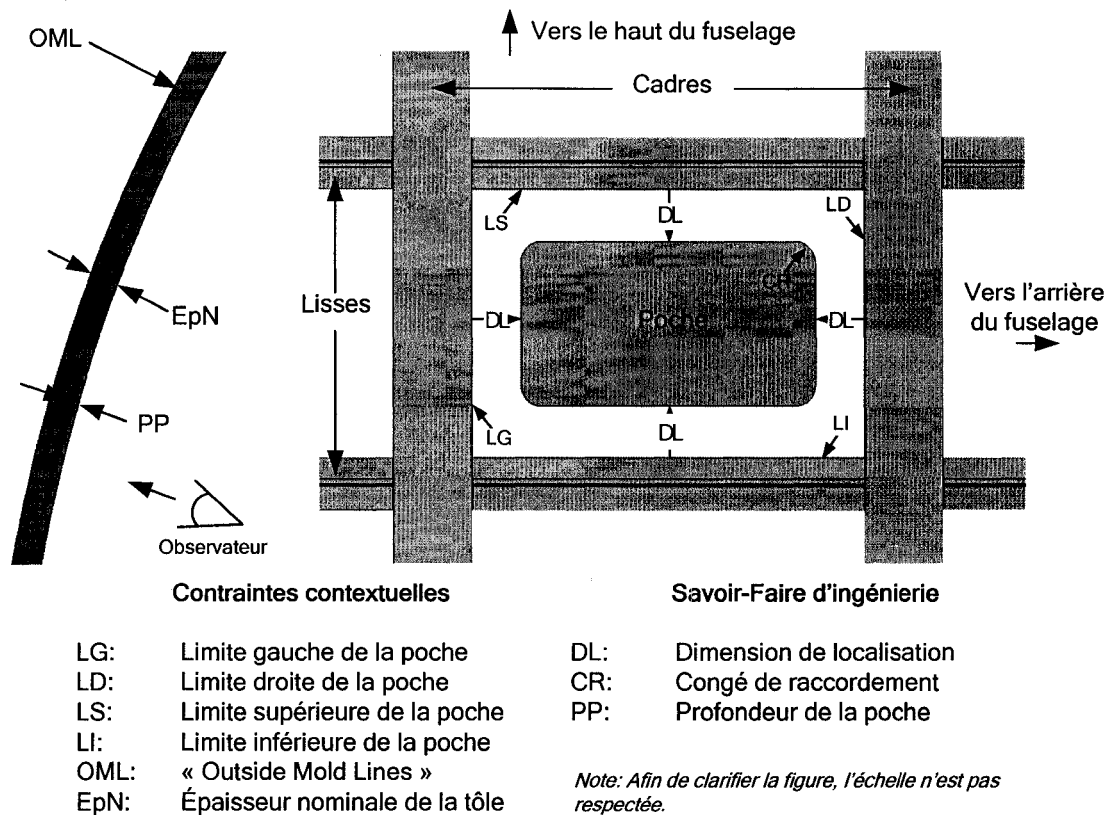


Figure 44 Paramètres et liens contextuels de la caractéristique poche

Nous avons défini, jusqu'à maintenant, le contexte de notre cas d'application ainsi que les concepts que nous utiliserons dans notre application. Voyons maintenant comment nous y arriverons.

4.5 Le choix d'une plateforme CAO

Nous voulons intégrer les concepts vus au chapitre précédent à un prototype d'application. Pour ce faire, nous devons choisir une plateforme CAO permettant d'automatiser la création des poches d'allègement.

La plateforme ciblée est l'outil CAO de nouvelle génération, Catia V5, de Dassault Systèmes. Ce choix est tout d'abord guidé par le fait que Catia est déjà largement

utilisée dans le domaine aéronautique et notre partenaire industriel, Bombardier Aéronautique, opère actuellement une migration vers cette version [8].

En deuxième lieu, nous voulons chiffrer les coûts et le potentiel de gain de l'automatisation de cette tâche sous une plateforme pertinente à notre partenaire industriel.

Troisièmement, Catia V5 intègre déjà, dans sa version de base, plusieurs outils facilitant l'intégration des concepts que nous voulons valider et par le fait même facilite la réalisation de notre projet d'automatisation.

Finalement, il est possible d'employer plusieurs types et niveaux de programmation sous Catia V5 et l'ÉTS dispose des licences nécessaires à leur utilisation. En effet, les *API (Application programming interface)* des langages Visual Basic et C++ sont accessibles afin d'interroger et de manipuler les modèles CAO de Catia V5 à l'aide de lignes de code.

4.6 Quelques différences entre Catia V5 et Catia V4

Catia V5 est un outil de nouvelle génération dont la philosophie d'utilisation diffère beaucoup de celle de V4. Il ne fait pas partie des objectifs de ce travail de dresser une liste exhaustive de ces différences, mais nous prenons quelques lignes pour souligner certaines améliorations technologiques de V5 par rapport à V4.

Catia V5 est un outil CAO très puissant, mais aussi très couteux et comme les entreprises n'ont pas tous les mêmes besoins de modélisation et les mêmes capacités financières, Dassault Systèmes offre son outil CAO sous forme de « solutions » classées en différents « Domaines Catia », chacun d'eux étant subdivisé en ateliers. Par exemple, le domaine « Mechanical Design » regroupe différents ateliers nécessaires à la modélisation mécanique tel que « Part Design », « Assembly Design », « Composites

Design », « Weld Desing », « Sheetmetal Design », etc, ainsi que des ateliers de conception d'outillages et de production de dessins 2D [56]. Chacun des ateliers est à son tour composé d'outils permettant d'effectuer des tâches de modélisation de bas et de haut niveau.

La première différence, qui est fondamentale entre Catia V5 et V4, est la philosophie de conception à la base de l'utilisation de V5 qui repose principalement sur la création d'esquisses permettant, entre autre, de créer les primitives. Cet outil permet de paramétrer et de contraindre le contour à l'intérieur et à l'extérieur de l'esquisse, i.e. d'en lier des éléments à des plans, des courbes, des points, des arêtes et des faces de prismes, etc. Il en résulte, entre autre, la possibilité de définir des contours très complexes, contenant toutes sortes de liens tout en demeurant modifiable ultérieurement. De plus, un outil de maintenance de la validité du contour permet à l'utilisateur de résoudre rapidement les « incohérences » lorsqu'elles surviennent et bloquent la conception.

Deuxièmement, Catia V5 permet d'établir des liens de toutes sortes et contrairement à V4, ils sont persistants, i.e. ils demeurent « actifs » dans le modèle et c'est le moteur d'inférence de V5 qui les gère. Cette capacité de V5 jouera un rôle crucial dans le développement de notre application.

Troisièmement, Catia V5 dispose d'une solution de gestion de la connaissance, « Knowledgeware », très évoluée. Les différents ateliers de ce domaine Catia permettent, entre autre, de réutiliser les connaissances liées au savoir-faire d'ingénierie d'une organisation en créant des relations entre des paramètres et des éléments géométriques, en incorporant des règles de vérification au sein des modèles et en définissant différentes configurations possibles d'un même produit à l'aide de tables de conception.

Quatrièmement, Catia V5 possède une fonction de « *publication* » d'objets à l'intérieur des ateliers de conception de pièces et d'assemblage permettant de gérer les accès externes aux éléments des modèles. Comme nous le verrons plus loin, l'un des choix méthodologiques que nous avons fait est de limiter le logiciel à créer des liens uniquement avec les objets publiés. Cela peut sembler rigide, mais cette façon de faire contribue à structurer l'assemblage et à maintenir la cohérence entre les modèles. De plus, dans notre approche, le respect de ce choix méthodologique est crucial afin de maintenir la chaîne d'associations entre les modèles.

Finalement, un autre outil puissant et flexible de Catia V5, et sur lequel repose ce travail, est celui qui permet de créer des caractéristiques définies par l'utilisateur, les UDFs (User-Defined Features). En effet, il permet de créer, de typer et de gérer des caractéristiques et même des pièces spécifiques à un usager ou à un domaine (nous approfondirons plus en détail l'utilisation de ce module dans une section subséquente).

4.7 Stratégie d'exploitation de Catia V5 pour le développement de l'application

Nous allons maintenant détailler de quelles façons nous utilisons la plateforme CAO Catia V5 et les modules métiers présentés dans la section précédente afin de développer notre application.

Dans un premier temps, nous nous sommes familiarisés avec Catia V5 en mode « *interactif* », i.e. comme le font actuellement les concepteurs. Lors de cette étape nous avons construit une maquette numérique représentant un assemblage de panneau de revêtement, que nous désignerons maintenant par « assemblage type ». Cette maquette, représentative de celles que l'on retrouve chez notre partenaire industriel, est utilisée tout au long du développement de notre application.

Durant cette étape, nous avons exploré les différentes fonctions et outils du logiciel afin de mettre au point des méthodologies de conception pour les pièces de l'assemblage (les lisses, les cadres et la tôle), ainsi que pour la caractéristique poche.

Dans un deuxième temps, nous avons utilisé l'API de Catia V5 pour développer un programme Visual Basic permettant d'instancier automatiquement les poches contenues dans un APR en limitant les interventions de l'utilisateur.

4.8 Éléments définissant le cadre, la lisse et la tôle de revêtement

Ce travail repose en bonne partie sur l'utilisation des caractéristiques. Pour ce faire, nous avons recouru à l'outil UDF de Catia V5 pour créer une caractéristique de type poche. Nous avons également utilisé des UDFs pour encapsuler les pièces composant l'assemblage, i.e. le cadre, la lisse et la tôle. Cette approche permet, entre autre, de typer les pièces et facilite la programmation de l'application.

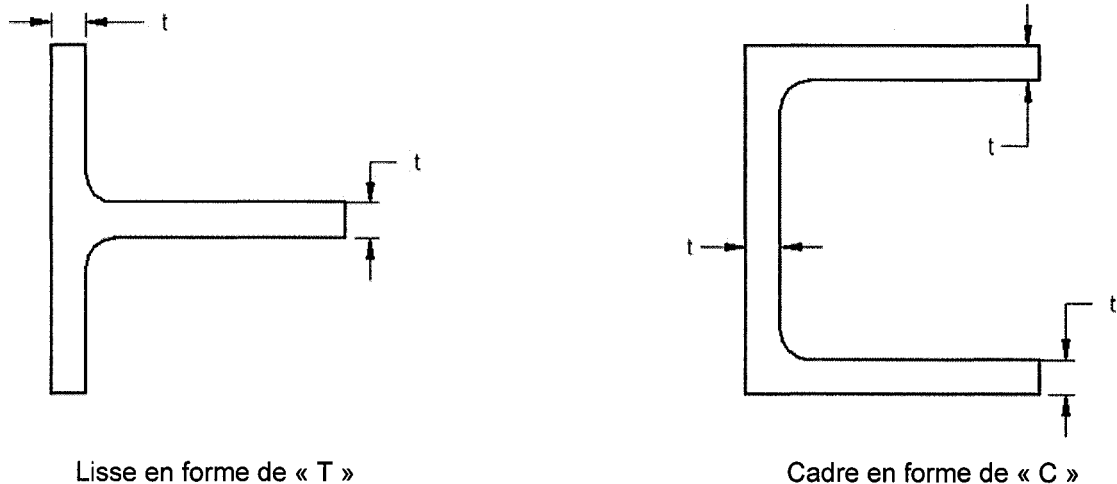
Avant d'aller plus loin dans la description de la démarche de création des pièces, nous devons établir les éléments nécessaires à la définition d'un cadre, d'une lisse et d'une tôle de revêtement.

Un cadre est suffisamment défini pour les besoins de l'application lorsque l'on connaît :

- Son type (dans notre cas le cadre est profilé en forme de C) ;
- La surface maîtresse OML régissant la forme de la face en contact avec la tôle ;
- Le « Fuselage Station » qui le positionne longitudinalement dans la maquette de l'avion ;
- L'épaisseur du métal utilisé pour fabriquer le cadre, voir la figure 45. Cette propriété est ajoutée pour le développement d'un UDF de soyages et de découpe, voir section 5.3.2.1.

Une lisse est suffisamment définie pour les besoins de l'application lorsque l'on connaît :

- Son type (dans notre cas elle est profilée en forme de T) ;
- La surface maîtresse OML régissant la forme de la face en contact avec la tôle ;
- Son plan de positionnement radial dans la maquette de l'avion ;
- L'épaisseur du métal utilisée pour fabriquer la lisse, voir la figure 45. Cette propriété est ajoutée pour le développement d'un UDF de soyages et de découpe, voir section 5.3.2.1.



Lisse en forme de « T »

Cadre en forme de « C »

Figure 45 Épaisseur du métal utilisé pour modéliser les lisses et les cadres

Une tôle de revêtement est suffisamment définie pour les besoins de l'application lorsque l'on connaît :

- La surface maîtresse OML lui donnant sa forme ;
- L'épaisseur de la tôle.

4.9 Structure et associations d'éléments de l'APR sous Catia V5

Nous avons défini les paramètres nécessaires à la définition des pièces de l'APR. Il faut maintenant comprendre comment les pièces, les caractéristiques, les paramètres et les éléments géométriques peuvent interagir dans l'atelier d'assemblage de Catia V5.

L'un des problèmes rencontrés lors du développement du prototype est causé par la présence de références circulaires attribuables à une mauvaise hiérarchie des pièces dans l'assemblage. Pour éviter ce type de blocage, une analyse minutieuse des associations est primordiale.

Dans notre cas, cette analyse se traduit par un graphique d'interdépendance des pièces, des caractéristiques, des paramètres et des différents éléments géométriques impliqués dans l'assemblage. Ce graphique indique les interactions entre ces objets ainsi que le parcours des associations au sein de la structure de l'assemblage, voir la figure 46 ci-dessous.

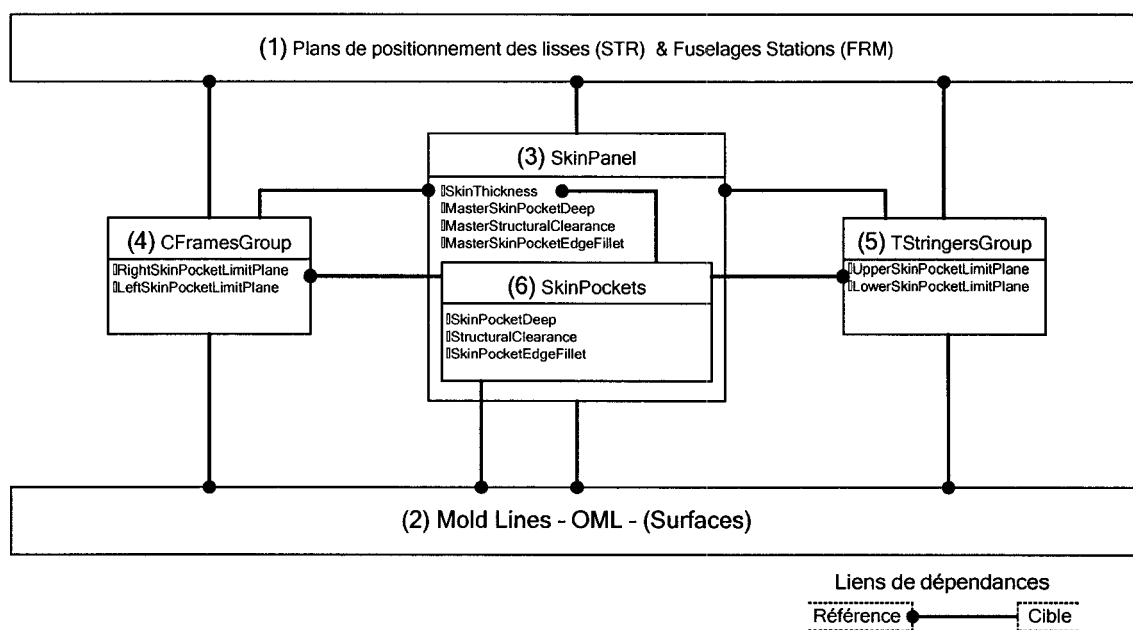


Figure 46 Graphique d'interdépendance des pièces, des caractéristiques, des paramètres et des éléments géométriques impliqués dans l'assemblage

Cet exercice permet de dresser « le plan de construction » de l'assemblage. Il définit l'ordre dans lequel les pièces doivent être introduites dans l'assemblage, le nombre d'associations, de paramètres ainsi que les accès nécessaires aux informations externes pour chaque pièce et paramètres.

Les rectangles nommés «Plans de positionnement des lisses (STR) & Fuselages Stations (FRM)» et «Mold lines – OML – (Surfaces)» représentent les modèles devant apparaître à la tête de la hiérarchie de l'assemblage, car leur définition ne repose sur aucune pièce en amont. Par la suite, le modèle contenant la tôle, «SkinPanel» peut être instancié. Il n'a besoin que des modèles déjà présents dans l'assemblage pour être défini. De même, les pièces de structure, telles que les lisses et les cadres, sont instanciés et associées aux autres modèles dont ils ont besoin pour être définis. Finalement, les poches, «SkinPockets» dans le graphique, sont instanciées dans le modèle de la tôle en se rattachant à quatre des cinq modèles déjà présents dans l'assemblage. Il est à noter que l'APR sans les poches d'allègement est définie durant la phase de conception préliminaire, tandis que les poches sont introduites lors de la phase de conception détaillée.

Les associations entre les caractéristiques, les pièces et les paramètres sont représentées à l'aide d'un trait noir et se font d'une cible vers une référence. Le point noir à l'une des extrémités représente le côté référence de l'association. C'est une sorte de «grappin» à une caractéristique, une pièce, un paramètre ou un élément géométrique. L'autre extrémité du trait représente le côté cible de l'association.

Cet exercice peut sembler anodin, mais l'établissement d'une hiérarchie adéquate fait épargner un temps considérable lié au blocage et au déverminage de l'assemblage.

Le processus de développement de notre application est en soi un processus de conception. Nous devons mener plusieurs tâches simultanément et de façon itérative, si bien qu'au fur et à mesure que nous découvrons les fonctions de Catia V5, nous adaptons et modifions nos méthodologies de conception et parfois même la structure de l'assemblage. Les paramètres inscrits dans le graphique ci-haut sont issus des différents tests méthodologiques auxquelles s'est ajoutée l'expérience acquise au cours du processus de développement. Toutefois, nous retenons que le graphique

d'interdépendance doit être établi, ne serait-ce que sous forme d'ébauche, avant de débiter les activités de définition géométrique.

Les méthodologies de conception du cadre, de la lisse, de la tôle et de caractéristique poche ont été mises au point de façon à ce que ces objets soient instanciés sous forme d'UDFs dans l'assemblage. Avant d'aller plus loin, il est primordial de décrire et de comprendre le processus de création et de manipulation des UDFs sous Catia V5.

4.10 Utilisation des UDFs (User-Defined Features)

Comme nous l'avons mentionné précédemment, les UDFs permettent de créer des caractéristiques, mais on peut également les utiliser pour créer des pièces. Lors du processus d'apprentissage de Catia V5 et de la fonction UDF, nous avons noté plusieurs avantages associés à leur utilisation ce qui justifie l'investissement de temps nécessaire à le maîtriser (nous le montrerons plus loin à l'aide de données chiffrées). Notons entre autre que :

1. Les instances d'UDFs permettent de diminuer le nombre d'éléments géométriques encombrant l'arbre des spécifications;
2. La création d'un prototype d'UDF permet d'encapsuler et de réutiliser le savoir-faire d'ingénierie sous formes de paramètres, de règles, d'associations, etc.;
3. Les prototypes d'UDFs peuvent être gérés et classés dans un catalogue;
4. L'instanciation des UDFs peut se faire à partir d'un catalogue, d'un modèle ou directement en le sélectionnant dans le modèle ayant servi à le définir ;
5. La géométrie accessible dans le prototype et l'instance des UDFs est contrôlable grâce à la fonction de publication;
6. Il est possible de typer le prototype d'UDF et d'effectuer des opérations de recherches sur ses instances tout comme sur une entité « *standard* » du système;

7. Les liens externes permettant de récupérer de l'information du contexte (comme entrées de l'UDF) ainsi que ceux permettant au contexte d'exploiter les sorties d'une instance d'UDF sont maintenus à jour dynamiquement ;
8. Les instances d'UDFs peuvent être manipulées par les commandes de copier / coller standards à toutes les applications Windows ;
9. Dans notre cas, l'utilisation d'instances d'UDFs ne nécessite aucune contrainte d'assemblage.

Toutefois, durant le développement du prototype d'application, nous avons constaté que pour créer correctement un UDF, on ne peut recourir à n'importe laquelle des fonctions de Catia V5 et selon n'importe quel ordre sans qu'il n'y ait des difficultés lors de l'instanciation. De plus, les messages d'erreurs générés par le système sont parfois ambigus dans l'interface de création de V5 et ils ne sont guère plus clairs lors de la création d'UDFs.

C'est pourquoi, avant de débiter à travailler avec les UDFs, il est crucial de bien comprendre la structure et le fonctionnement des UDFs de Catia V5, exposés ci-dessous à la figure 47.

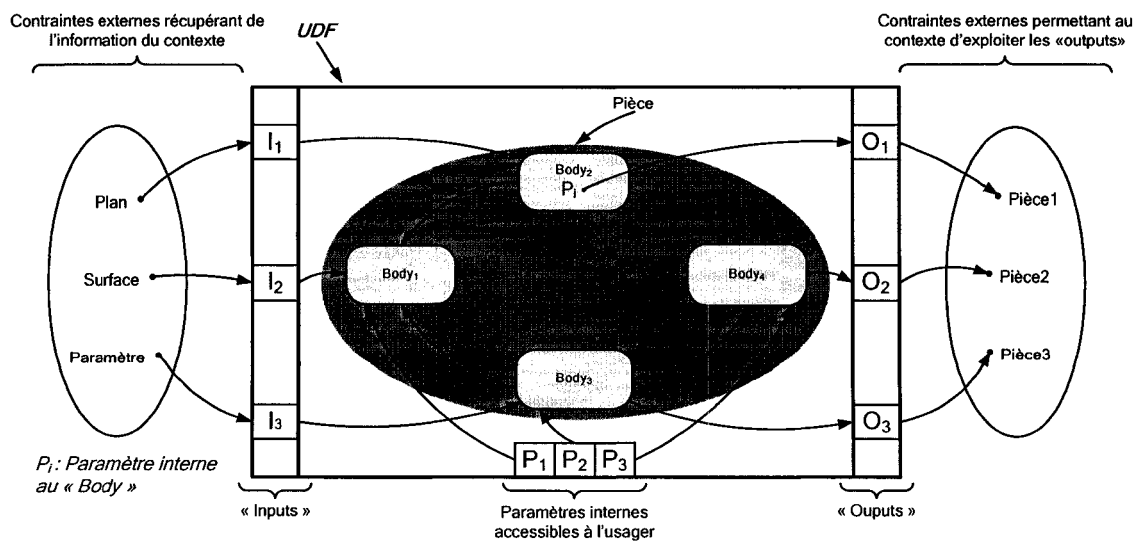


Figure 47 Structure d'un prototype d'UDF (Users-Defined Features) dans Catia V5

L'UDF de Catia V5 est un peu comme une boîte noire dans laquelle on introduit des entrées, « inputs », via des contraintes externes à l'UDF récupérant de l'information du contexte. La source de ceux-ci peut se présenter comme étant la valeur d'un paramètre, tel qu'un nombre désignant une distance, un angle, etc, ou un élément géométrique tel qu'une ligne, une face, un point, etc., et peut se trouver à l'intérieur même du modèle ou dans un autre modèle.

Les entrées fournies doivent correspondre explicitement à ce que l'UDF requiert pour être instancié. Par exemple, si l'UDF a besoin d'une surface comme entrée et que l'utilisateur pointe la face plane d'une extrusion, un message d'erreur sera affiché.

Ces entrées sont ensuite traitées par l'UDF par un processus qui consiste à « rejouer » les étapes de modélisation qui le constituent. Lorsque les entrées sont fournies, une instance de l'UDF est créée dans le modèle l'accueillant et devient une entité géométrique pouvant être traitée par les outils de modélisation de Catia V5.

Par défaut, une seule sortie, « output », est définie lors de la création d'un prototype d'UDF et elle correspond au résultat géométrique de celui-ci, comme par exemple un prisme, une découpe, etc. Il est toutefois possible d'ajouter des sorties à un UDF permettant au contexte de les exploiter en créant d'autres contraintes externes vers eux.

Finalement, pour que l'utilisateur puisse contrôler le comportement de l'instance d'UDF, des paramètres peuvent être publiés par le prototype. Ceux-ci agissent sur les contraintes internes à l'UDF et sont indépendants du contexte (comme c'est le cas de P_i à la figure 47). Un UDF, certaines de ses sorties ou de ses paramètres publiés, sont autant d'objets pouvant devenir des entrées pour un autre UDF et créer une chaîne d'UDFs. L'une des stratégies de ce travail est d'ailleurs basée sur ce principe et nous y reviendrons un peu plus loin.

4.11 Représentation d'une instance d'UDF dans l'arbre de spécifications

Lorsqu'un UDF est instancié dans un modèle, il est représenté par une icône, qui peut être personnalisée, dans l'arbre des spécifications. Un petit (+) apparaîtra à côté de l'instance si des éléments géométriques ou des paramètres internes sont accessibles. Dans l'exemple de la figure 48 ci-dessous, on montre une instance d'UDF d'un cadre en forme de «C» dans l'arbre de spécifications. Quatre paramètres sont publiés et sont représentés par une icône en forme de cube. Un élément géométrique, de type plan, est également accessible et il est représenté par une autre icône. Finalement, dans l'exemple de la figure 48, un autre élément, intitulé « *Activity=true* », apparaît sous la liste des paramètres publiés. Il nous informe de l'état de l'instance dans le modèle où il est inséré. Naturellement le terme « *true* » nous indique que l'instance est active dans le modèle, tandis que le terme « *false* » nous indique que l'instance est désactivée. Cet exemple montre l'aspect général que prend une instance d'UDF dans l'arbre. Nous verrons plus en détails la signification et la raison des paramètres présents dans l'instance d'UDF du cadre.

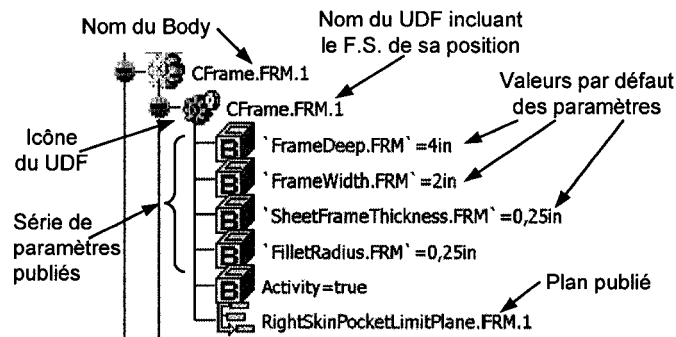


Figure 48 Représentation générale d'une instance d'UDF dans l'arbre des spécifications

Jusqu'à maintenant, nous avons établi d'un côté les paramètres nécessaires à la définition de chaque pièce constituant notre assemblage type, les liens contextuels interagissant entre eux et la hiérarchie de l'assemblage. De l'autre côté, nous avons décrit la structure de l'outil UDF de Catia V5. Voyons maintenant comment unir tous ces éléments afin de les intégrer dans un assemblage dynamique, paramétré et associatif.

4.12 Méthodologie de conception de l'assemblage type sous Catia V5

Nous discutons du cas le plus simple, celui d'un APR du fuselage central à simple courbure, dont les poches d'allègement sont de profondeur uniforme. Les éléments de structure se composent de cadres en forme de « C » et de lisses profilées en forme de « T ». La figure 49 présente un assemblage de ce type.

Les flèches permettent de positionner l'APR par rapport à l'ensemble du fuselage et serviront à déterminer une nomenclature pour les pièces de référence nécessaires à la définition des poches. En se plaçant comme observateur à l'intérieur du fuselage et en regardant l'APR de face, on peut constater que le dos de chaque cadre servira de limite gauche à une poche tandis que le ventre, soit le coté ouvert du cadre, servira de limite droite. De même, la face du dessus du profil « T » de la lisse servira de limite pour le dessous de la poche et la face du dessous servira de limite pour le dessus de la poche.

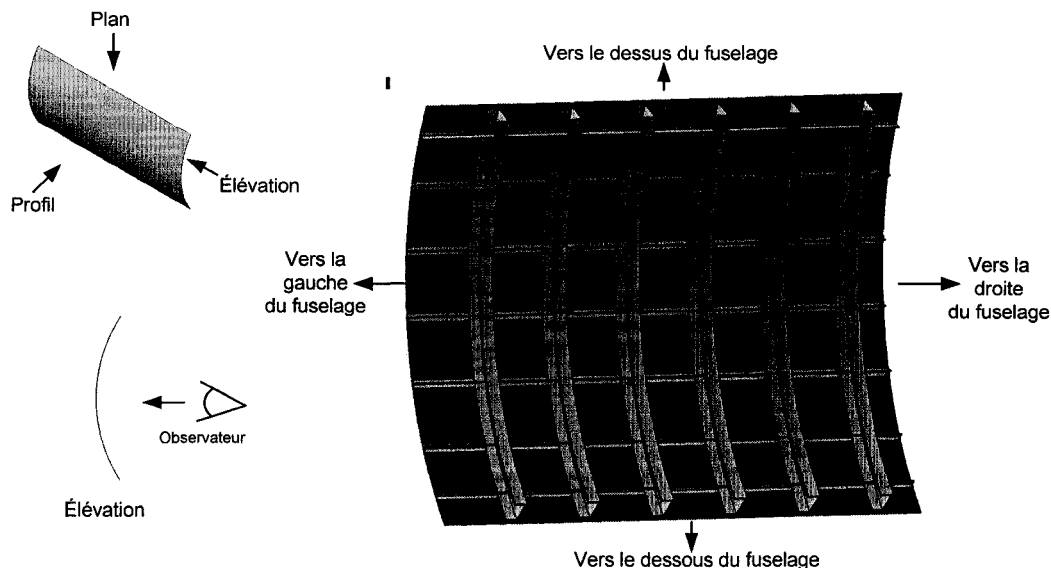


Figure 49 Exemple d'APR du fuselage central

Comme déjà mentionné, les pièces constituant l'ARP (les cadres, les lisses et la tôle) sont encapsulées dans des UDFs. Nous avons publié les paramètres et les éléments

géométriques nécessaires afin de créer des associations entre les différentes pièces de l'assemblage.

La figure 50 montre la structure de l'assemblage type utilisé. Cette hiérarchie respecte le graphique d'interdépendance des pièces, des caractéristiques et des éléments géométriques vu précédemment. Ainsi, les modèles nommés « PositionPlanes », contenant les F.S. et les plans de positionnement pour les lisses, et « MoldLines », contenant l'OML, sont placés en tête de l'assemblage. Les modèles suivants sont positionnés en fonction de leurs liens contextuels. De cette façon, nous évitons le problème des relations circulaires, et peu importe le niveau hiérarchique d'une modification, toutes les pièces en aval seront modifiées.

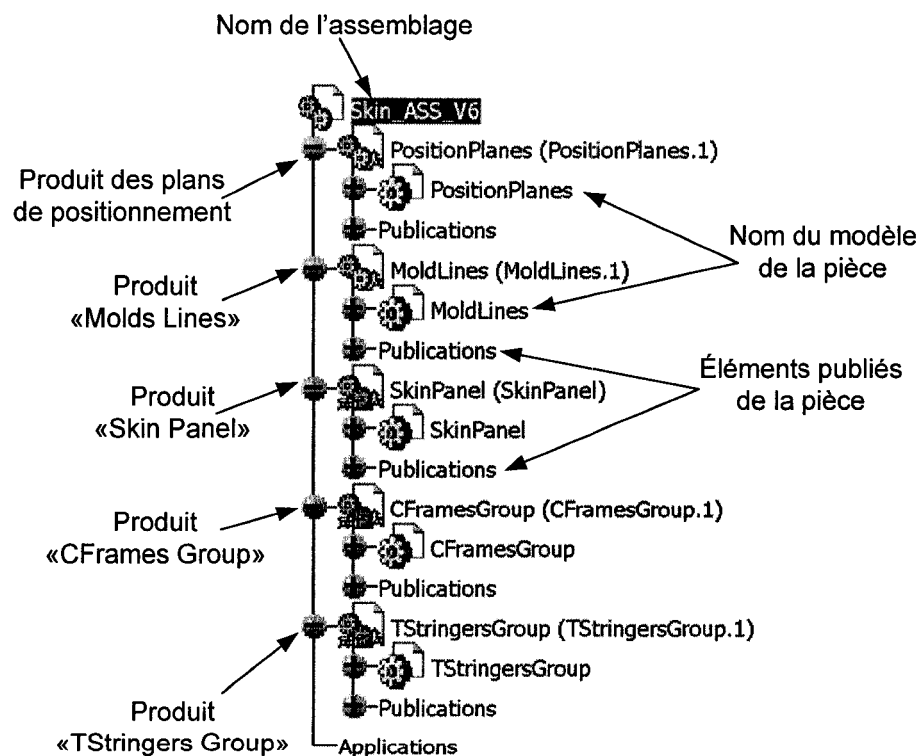


Figure 50 Arborescence de l'assemblage

Nous regroupons tous les modèles des pièces de même type ensemble. Les modèles contenant les cadres en forme de «C», par exemple, sont regroupés dans un même

produit que nous nommons «CFramesGroup». De même, tous les modèles contenant les lisses en forme de «T» sont regroupés dans un produit nommé «TStringersGroup». Chaque instance de cadre et de lisse fait partie d'un «body» correspondant à son plan de positionnement dans l'avion. Par exemple, l'UDF du cadre associé au «Fuselage Station » 1 est placé dans un « body » nommé «CFrame.FRM1» et le nom de l'UDF est le même, soit «CFrame.FRM.1», voir la figure 51. Cette structure facilite l'approche par programmation présentée plus loin pour automatiser la définition des poches en limitant la recherche des lisses et des cadres utilisés pour les limites des poches.

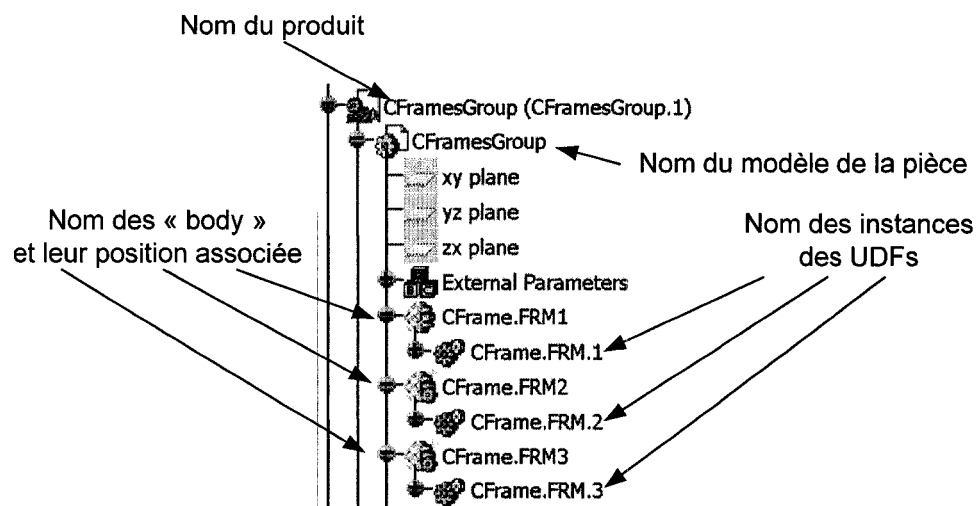


Figure 51 Arborescence et nomenclature du produit « Cadre en C » et ses UDFs

Avant d'aller plus loin et de décrire la structure des UDFs que nous avons créés pour l'assemblage type, il faut d'abord discuter de la méthodologie de modélisation de la caractéristique « poche ».

4.13 Différents essais de méthodologies de modélisation de poches

Cette section décrit les différentes approches que nous avons explorées pour la modélisation des poches d'allègement sous Catia V5. Notre objectif consistait à définir une méthode nous permettant d'encapsuler la poche dans un UDF et de créer ainsi une caractéristique poche. La séquence et le choix des étapes de modélisation, les types

d'entités géométriques et des paramètres utilisés dans le modèle servant de prototype au UDF de la poche ne sont pas triviaux. Il nous a fallu explorer plusieurs combinaisons afin de mettre au point une méthode compatible avec la fonction UDF. Rappelons également que cette méthode devait nous permettre de répondre aux sept exigences prescrites à la section 4.1.

Les premières méthodologies que nous avons explorées suivaient la même technique que celle employée sous Catia V4. Toutefois, sous Catia V5, cette méthode nous occasionnait plusieurs problèmes.

Plusieurs approches de modélisation de la poche s'offraient à nous. La première que nous avons explorée, et la plus « naturelle » considérant la philosophie de conception basée sur l'utilisation d'esquisses construites sur des plans de Catia V5, consistait à obtenir le contour de la poche sous forme d'une esquisse contrainte et rattachée au contexte de conception. Par la suite, nous modélisions la poche en réutilisant l'esquisse et la fonction « CutOut » pour créer une découpe dans la tôle. L'avantage principal à utiliser cette méthode est le regroupement des contraintes et des liens entre l'esquisse et les éléments de structure de l'APR à un seul endroit. Malheureusement, cette approche nous causait plusieurs problèmes.

Le premier problème est lié à l'utilisation du plan servant de support pour la construction de l'esquisse. En effet, ce dernier doit être tangent à la courbure de la surface IML et les courbes extraites des cadres et des lisses doivent être projetées sur ce plan. Le décalage des limites de la poche suivent donc un parcours euclidien plutôt que selon la courbure de l'IML. En procédant de la sorte, nous introduisons une erreur s'accroissant à mesure que le rayon de courbure de la surface OML diminue. De plus, avec la version R14 de Catia V5, il s'est avéré impossible de réutiliser dans un UDF une esquisse composée d'éléments géométriques issus de projections. Finalement, dans le cas d'une surface à double courbure, il fallait multiplier les opérations afin d'obtenir un plan tangent à la surface et ces opérations ne pouvaient être rejouées par l'UDF.

Comme nous voulions établir une méthode universelle pour les APRs à simple et à double courbure, l'approche du plan pour définir l'esquisse de la poche s'est avérée inadéquate. Nous devons donc trouver une méthode qui ne l'utilisait pas.

La seconde approche de modélisation que nous avons explorée consistait à définir le contour de la poche directement sur la surface IML en utilisant les éléments filaires obtenus par extraction des courbes limites des pièces de référence et d'inclure les congés de raccordement avant d'effectuer l'extrusion. Cette méthode multipliait considérablement les opérations et menait également à l'instabilité dans l'UDF.

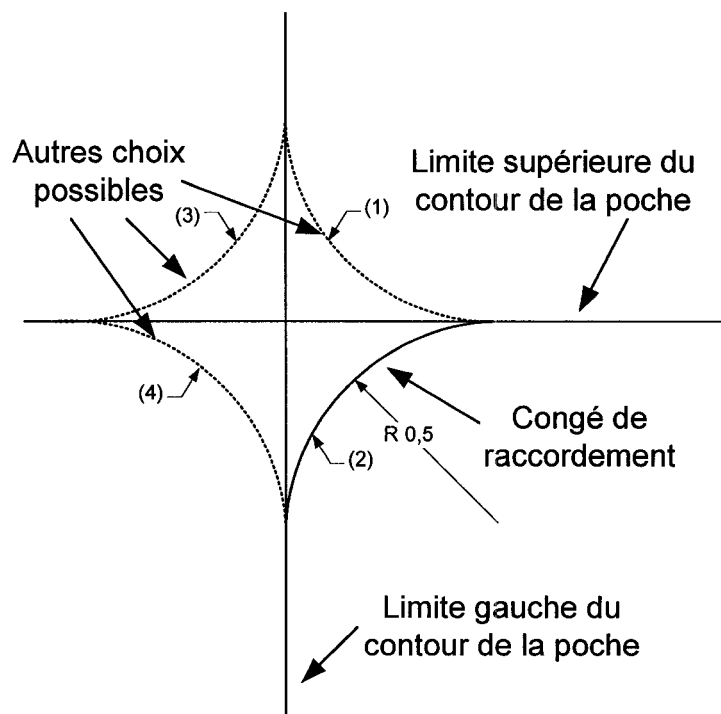


Figure 52 Choix de quatre congés possibles sous Catia V5

Par exemple, lorsqu'un congé de raccordement est effectué sur deux lignes, Catia V5 détecte quatre possibilités et demande l'intervention de l'utilisateur pour compléter la commande, voir la figure 52, ce qui devient problématique pour un processus que l'on veut systématique et sans intervention de l'utilisateur.

Il nous est apparu très clair qu'il était impossible de travailler directement avec les courbes limites extraites des lisses et des cadres, notamment à cause des multiples choix devant être effectués par l'utilisateur lors de « relimitations ». Toutefois, comme ces courbes sont extraites à l'aide d'arrêtes appartenant à des faces planes, il était très facile de définir des plans à partir des faces et de les rendre accessibles à l'utilisateur via les publications, voir la figure 53 et la figure 54 ci-dessous.

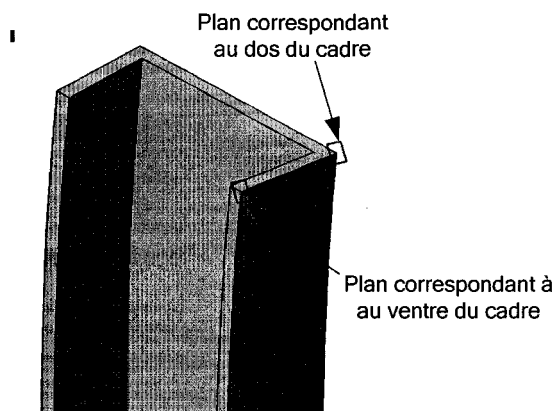


Figure 53 Plans correspondant au dos et au ventre du cadre

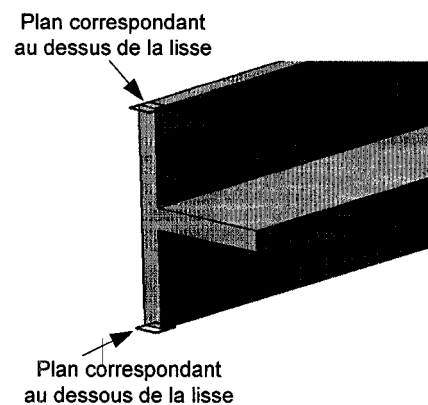


Figure 54 Plans correspondant aux faces supérieure et inférieure de la lisse

Finalement, nous avons tenté d'utiliser des courbes créées à l'aide d'intersections et des plans mentionnés dans le paragraphe précédent, mais il nous fallait créer une surface puis l'épaissir pour enfin en soustraire l'extrusion du modèle de la tôle. L'approche consistant à utiliser des courbes s'est avérée incompatible avec les UDFs et c'est pourquoi nous l'avons écartée.

Les connaissances dégagées de ces différents tests nous ont menés à l'élaboration de lignes directrices que nous récapitulons à la section suivante.

4.14 Résumé des conditions préalables et choix méthodologiques retenus

L'outil CAO de nouvelle génération Catia V5, que nous avons utilisé pour le développement de notre prototype, repose sur une philosophie de modélisation différente de celle de Catia V4 et, conséquemment, nécessite une approche différente.

Nous résumons certaines conditions préalables dans le but d'établir un point de départ au développement d'une méthodologie de modélisation des poches en plus de nous aider à structurer notre travail à l'aide des outils de Catia V5.

1. L'utilisation de la fonction UDF afin de créer une caractéristique de type poche ;
2. Les liens contextuels entre les modèles de l'assemblage sont établis uniquement à l'aide des objets publiés ;
3. La maquette numérique de l'assemblage incorpore deux « produits » à la tête de sa hiérarchie, les « PositionPlanes » et les « Mold Lines ». Les plans et l'OML y sont publiés pour permettre les accès externes aux modèles ;
4. Les éléments de structure sont regroupés selon leur type respectif sous un même « produit » et dans des « bodys » différents ;
5. Le concepteur de la pièce tôle intègre un paramètre désignant l'épaisseur du métal et le publie. Il place également les paramètres « directeurs » de la forme et du contour des poches dans un groupe nommé « SkinPocketParams » et la nomenclature suivante est adoptée :
 - i. Dimension de localisation : « MasterStructuralClearance »
 - ii. Congé de raccordement: « MasterSkinPocketEdgeFillet »
 - iii. Profondeur de la poche : « MasterSkinPocketDepth »

Note : chacune des poches instanciées dans le modèle sera liée à ces paramètres. Ceci permettra de contrôler l'ensemble des paramètres à un

seul endroit, ce qui économisera du temps lors d'un changement de profondeur des poches par exemple.

6. Le concepteur de la pièce cadre définit et publie des plans correspondant à ses limites, soit le dos et le ventre du cadre profilé en forme de « C », en adoptant la nomenclature suivante :

- i. Dos du cadre : « LeftSkinPocketLimitPlane »
- ii. Ventre du cadre : « RightSkinPocketLimitPlane »

7. Le concepteur de la caractéristique lisse définit et publie des plans correspondant à ses limites, soit les faces supérieure et inférieure, pour une lisse profilée en forme de « T », en adoptant la nomenclature suivante :

- i. Dessus de la lisse : « LowerSkinPocketLimitPlane »
- ii. Dessous de la lisse : « UpperSkinPocketLimitPlane »

Ces conditions préalables nous permettent d'établir le point de départ de notre contexte de conception.

4.15 Formalisation de la méthodologie de conception des poches d'allègement sous Catia V5

La section précédente nous a permis d'établir les conditions préalables au développement d'une méthodologie de conception des poches d'allègement. L'un des paradigmes à la base de ce travail est l'approche de conception par caractéristiques. Cette section est donc consacrée à la formalisation de la méthodologie de conception d'une caractéristique de type « poche » à l'aide de l'outil d'encapsulation UDF de Catia V5. Le tableau V, ci-dessous, résume les étapes à suivre.

Tableau V

Méthodologie de modélisation des poches d'allègement en mode interactif sous Catia V5

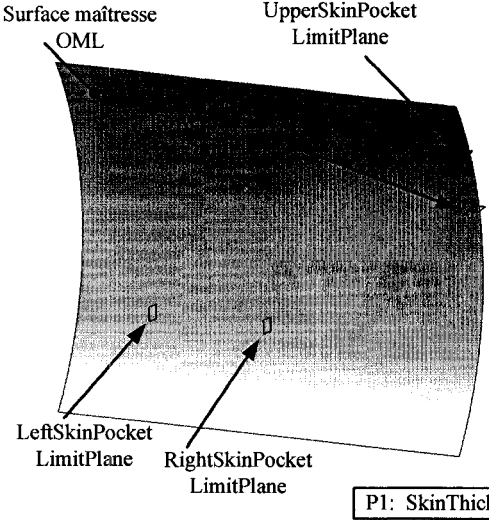
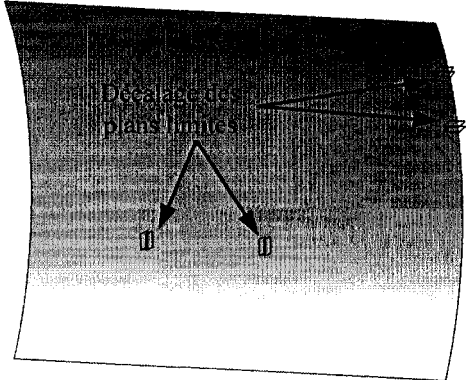
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #1 (préliminaire):</p> <p>Ouverture d'un nouveau modèle et construction des éléments nécessaires à l'élaboration du prototype de la poche, soit : une surface OML, quatre plans représentant les limites des éléments de structure et un paramètre désignant l'épaisseur de la tôle.</p> <p><i>Note : Cette étape permet de simuler les « inputs » nécessaire au UDF.</i></p>	 <p>Surface maîtresse OML</p> <p>UpperSkinPocket LimitPlane</p> <p>LeftSkinPocket LimitPlane</p> <p>RightSkinPocket LimitPlane</p> <p>P1: SkinThickness</p>
<p>Étape #2 :</p> <p>Décalage des plans limites servant à définir le contour de la poche d'allègement en utilisant le paramètre standard de 0.100".</p>	 <p>Décalage des plans limites</p>

Tableau V (suite)

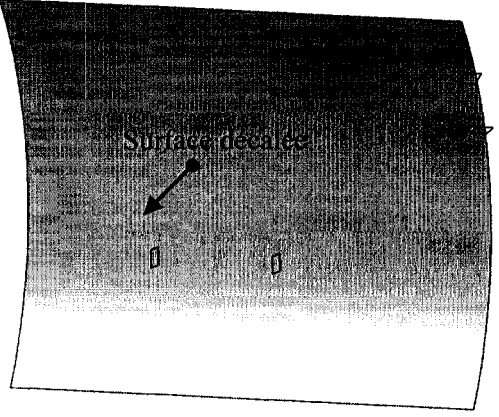
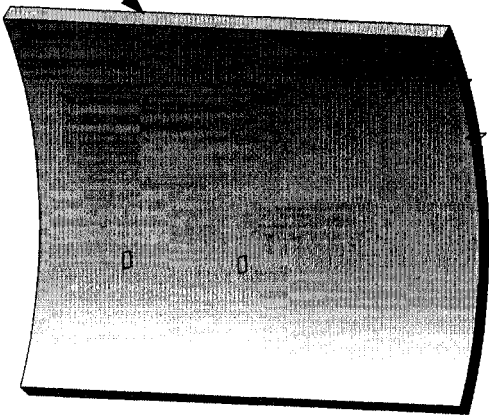
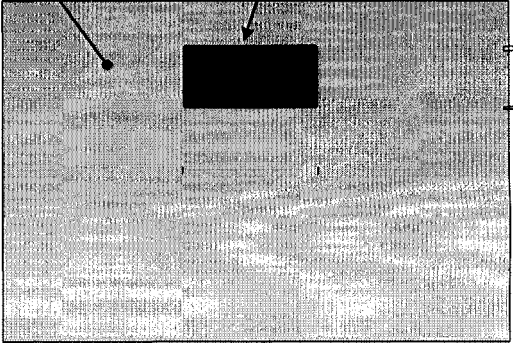
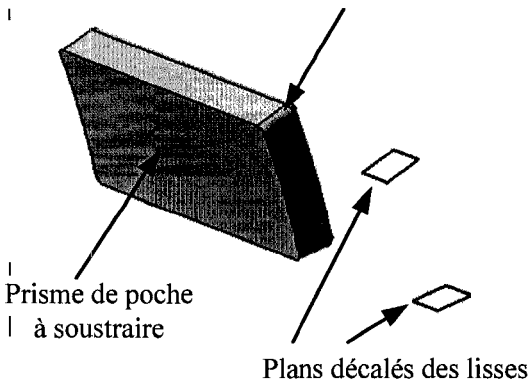
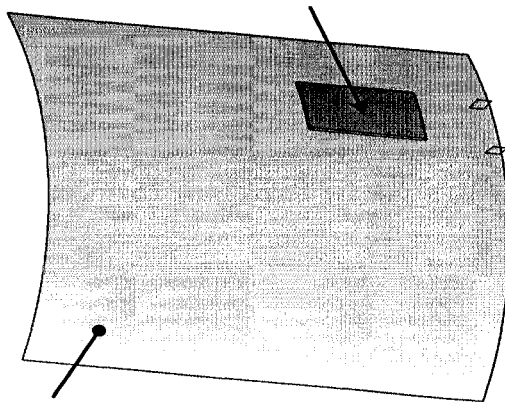
Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #3 :</p> <p>Création de la surface délimitant le fond de la poche à créer en utilisant le paramètre d'épaisseur de la tôle « SkinThickness » et le décalage de la surface délimitant le fond de la poche.</p>	
<p>Étape #4 :</p> <p>Création d'un solide en épaississant la surface créée précédemment et qui délimite le fond de la poche. L'épaisseur ici n'a pas beaucoup d'importance en autant qu'elle excède celle de la tôle. Nous avons utilisé 2".</p>	<p>Solide représentant la tôle de revêtement</p> 
<p>Étape #5 :</p> <p>Découpe du solide à l'aide des quatre plans décalés de la dimension de localisation standard de 0.10". Nous obtenons un solide définissant le contour de la poche à soustraire.</p>	<p>Surface délimitant le fond de la poche</p> <p>Solide définissant la poche à soustraire</p> 

Tableau V (suite)

Étapes de la méthodologie	Illustration à l'aide d'un APR
<p>Étape #6 :</p> <p>On effectue maintenant les congés de raccordement aux quatre coins du solide en utilisant le paramètre standard de 0.500".</p>	<p style="text-align: center;">Congés de raccordement</p>  <p style="text-align: center;">Plans décalés des lisses</p>
<p>Étape #7 :</p> <p>On peut maintenant soustraire le solide précédent de celui représentant la tôle de revêtement pour créer ainsi la cavité correspondant à la poche d'allègement.</p>	<p style="text-align: center;">Poche d'allègement</p>  <p style="text-align: center;">Tôle de revêtement</p>

Comme nous l'avons expliqué au chapitre 1, la méthodologie observée en entreprise pour la création des poches d'allègement sous Catia V4 compte 14 étapes distinctes et demande la manipulation d'au moins quatre types d'éléments géométriques différents (des lignes, des plans, des surfaces et des solides). Celle que nous avons développée sous Catia V5 compte six étapes, n'utilise que trois types d'entités (des surfaces, des plans et des solides) en plus d'être paramétrée et associée aux pièces de référence grâce à des liens persistants, ce qui en fait un modèle totalement réactif au contexte.

4.16 Publications des éléments des UDFs cadre, lisse et tôle

Nous avons également utilisé l'outil UDF pour encapsuler les pièces structurales constituant l'APR (les lisses et les cadres), ainsi que la tôle de revêtement de l'APR. Les éléments géométriques et les paramètres publiés dans ces UDFs vont permettre de créer les liens contextuels nécessaires à la définition de la poche puis éventuellement à celui du soyage et de la découpe du cadre.

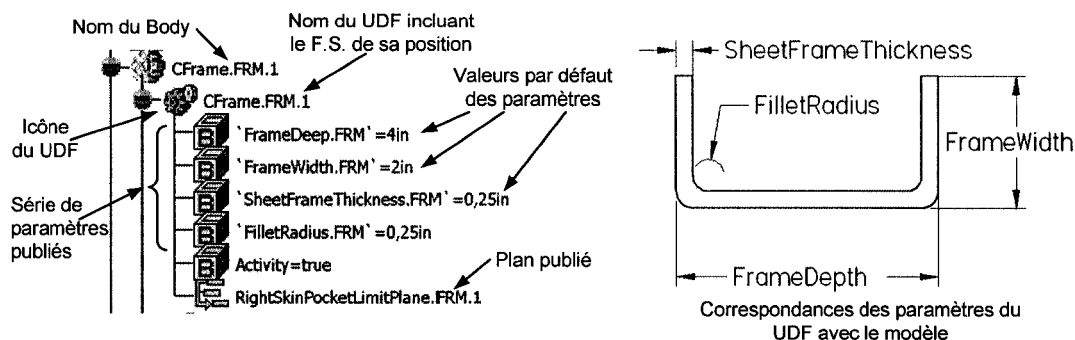


Figure 55 Paramètres et éléments géométriques publiés dans l'UDF cadre

La figure 55 montre les publications de l'UDF cadre et la correspondance avec le modèle géométrique de la pièce. Les paramètres «*FrameDepth.FRM*», «*FrameWith.FRM*», «*SheetFrameThickness.FRM*» et «*FilletRadius.FRM*» régissent la forme du cadre, permettant à l'utilisateur de la modifier au besoin, et serviront de contraintes contextuels d'entrées pour l'UDF de soyaages et découpes. Le plan «*RightSkinPocketLimitPlane*» correspond au plan du côté du ventre du cadre et servira de limite droite à la poche lors de son instanciation. On remarquera que pour le cadre, la limite gauche de la poche correspond à la face du dos du cadre et que ce dernier est positionné à l'aide du plan du F.S. correspondant à sa position dans le fuselage. Nous avons évité la duplication d'éléments géométriques en associant directement la limite gauche de la poche au F.S.

La figure 56 montre les publications de l'UDF de la lisse et la correspondance avec le modèle géométrique de la pièce. Les paramètres «*StringerHead.STR*», «*StringerBase.STR*» et «*StringerThickness*» régissent la forme de la lisse, permettant à l'utilisateur de la modifier au besoin, et serviront également de liens contextuels d'entrées pour l'UDF de soyages et découpes. Les deux plans publiés, «*BottomSkinPocketLimitePlane*» et «*UpperSkinPocketLimitePlane*» correspondent aux faces de la tête de la lisse et seront utilisés via les contraintes contextuelles d'entrées de la poche afin de définir les limites supérieure et inférieure de celle-ci.

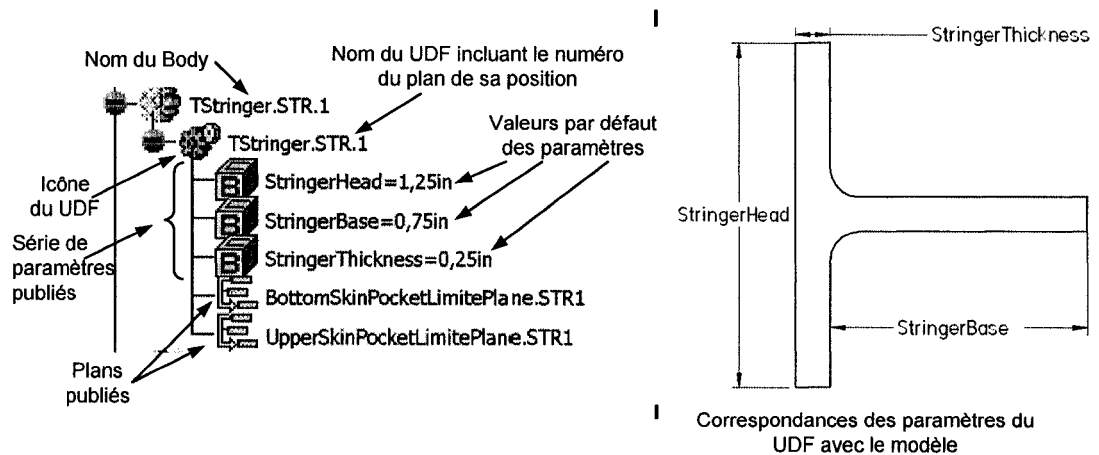


Figure 56 Paramètres et éléments géométriques publiés dans l'UDF lisse

Finalement, la figure 57 montre le paramètre «*SkinThickness*», correspondant à l'épaisseur de la tôle de revêtement. Ce dernier sera nécessaire pour l'instanciation et le positionnement à l'intérieur du fuselage des cadres, des lisses et de la poche.



UDF de la tôle

Figure 57 Paramètre publié dans l'UDF de la tôle

4.17 Assemblage type et fonctionnement des associations persistantes

Maintenant que nous avons défini chacun des modèles constituant notre assemblage type, soit les «Fuselage Stations», les «Mold Lines», la tôle de revêtement, le groupe de cadres et celui des lisses, voyons comment est construit l'assemblage type et comment il réagit lorsque nous effectuons des modifications et changeons la configuration de l'APR (simple courbure à double courbure).

Premièrement, la création de l'assemblage type à l'aide d'UDFs élimine complètement le recours aux contraintes d'assemblage entre les pièces qui sont normalement nécessaires au maintien de leur position. Cela allège considérablement l'assemblage et réduit les calculs de résolution de contraintes. En effet, la méthodologie de conception des pièces encapsulées dans les UDFs, fait en sorte que la pièce est correctement positionnée dès son instanciation. Cela est possible grâce à la série d'entrées fournies à l'UDF. Par exemple, lorsqu'un cadre est ajouté à l'assemblage, l'OML, le « Fuselage Station » et l'épaisseur de la tôle de revêtement sont nécessaires à son instanciation. L'UDF du cadre est conçu pour adapter la forme du cadre à l'IML tandis que la position longitudinale est donnée à l'aide du « Fuselage Station » et la position transversale est contrôlée en partie par l'épaisseur de la tôle.

Cette capacité qu'a notre assemblage de contrôler la réaction de ses différentes pièces et caractéristiques est un des aspects très intéressants pour la propagation des corrections et modifications subséquentes à sa définition initiale. En effet, si l'on devait modifier la surface OML, par exemple, on s'attendrait à devoir redéfinir et repositionner les cadres et les lisses de l'assemblage. Ce serait le cas sous Catia V4 et plusieurs heures seraient nécessaires pour propager cette modification. Or, il n'en est rien sous Catia V5, qui recalcule automatiquement les nouvelles formes et les nouvelles positions de la tôle, des lisses et des cadres engendrées par le changement. Ceci est possible grâce aux liens contextuels persistants établis entre l'OML et les éléments de structure, voir la figure 58

ci-dessous. Plus intéressant encore, le temps nécessaire au calcul de la mise à jour est remarquable, il faut dix secondes tout au plus.

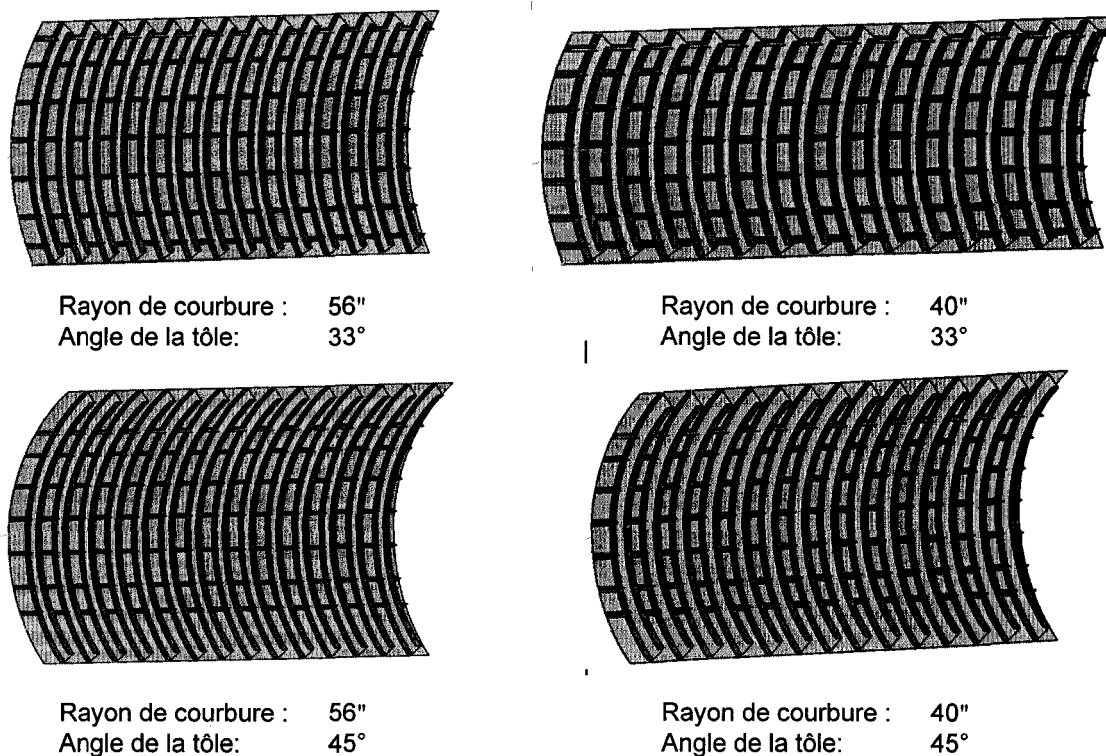


Figure 58 Différents rayons de courbure et angles de la tôle

De plus, cette mise à jour est complètement gérée par Catia V5. Lorsque Giguère a développé son prototype de soyages et de découpes en 2001, il a dû concevoir et développer lui-même un processus permettant la propagation des changements à cause de l'état d'avancement de Catia V5 à cette époque, ce qui lui a demandé plusieurs semaines de développement et, malheureusement, son application fonctionne uniquement sous Catia V5 R6.

Ainsi, les corrections et modifications, de forme ou de position, sont propagés automatiquement au sein de notre assemblage type par l'outil. De plus, il est possible de modifier les options de Catia V5 afin de contrôler les mises à jour de façon manuelle.

Dans tous les cas, dès que Catia V5 détecte un changement, toutes les pièces impactées sont affichées en rouge, indiquant à l'utilisateur qu'il doit lancer une mise à jour.

En deuxième lieu, la fonction de publication de Catia V5 permet de définir l'équivalent d'un pointeur en programmation. Les éléments géométriques en aval sont donc « attachés » à la publication et non directement aux éléments en amont. Ceci empêche un enchevêtrement de liens et par ailleurs en changeant le pointeur d'élément, le mécanisme de mise à jour propage automatiquement le changement au sein de l'assemblage. Par exemple, pour passer d'une configuration d'APR à simple courbure vers une configuration à double courbure, il suffit de changer la surface pointée par la publication et Catia V5 recalcule les nouvelles formes des lisses, des cadres et de la tôle correspondants.

La figure 59 illustre les différences entre les deux types de surfaces et, comme on peut le constater, la forme d'une lisse et d'un cadre pour la surface à simple courbure est la même peu importe sa position dans l'APR. Il en est tout autrement dans le cas d'une surface à double courbure.

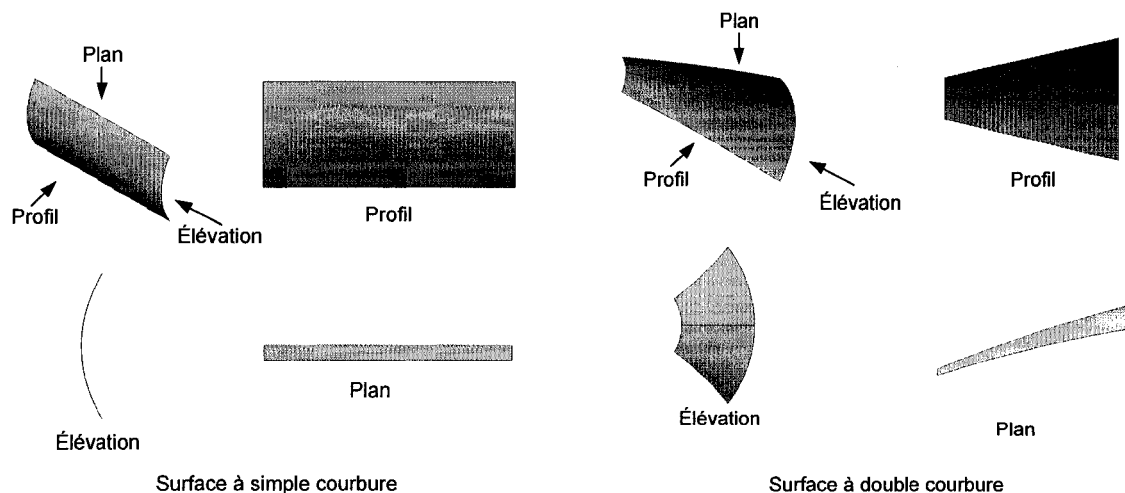


Figure 59 Exemple de surfaces à simple et double courbure

Catia V5 et la méthode d'encapsulation du cadre dans l'UDF prennent en charge cette adaptation, qui nécessite seulement 30 secondes de calculs pour l'ensemble de l'APR, et libère le concepteur de cette tâche.

La figure 60 montre les éléments de l'interface impliqués dans le changement de la surface OML simple courbure vers une surface à double courbure. Il suffit de modifier l'objet relié à la publication de la surface, nommé «OML», pour que Catia V5 recalculé les formes des pièces. La figure 61 montre le résultat final sur l'APR.

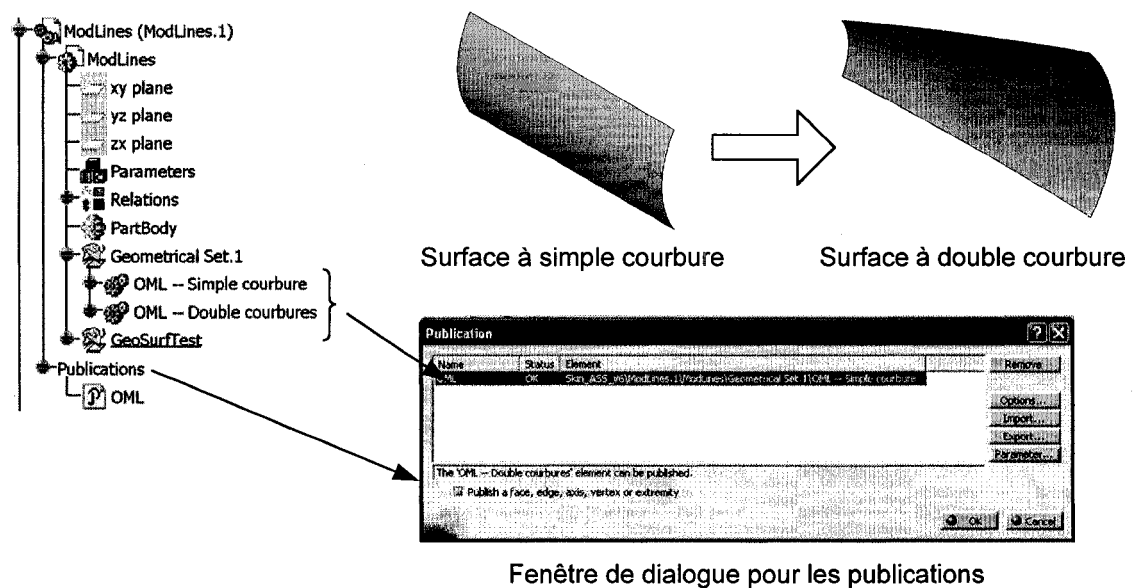


Figure 60 Processus de changement de la surface liée à la publication « OML »

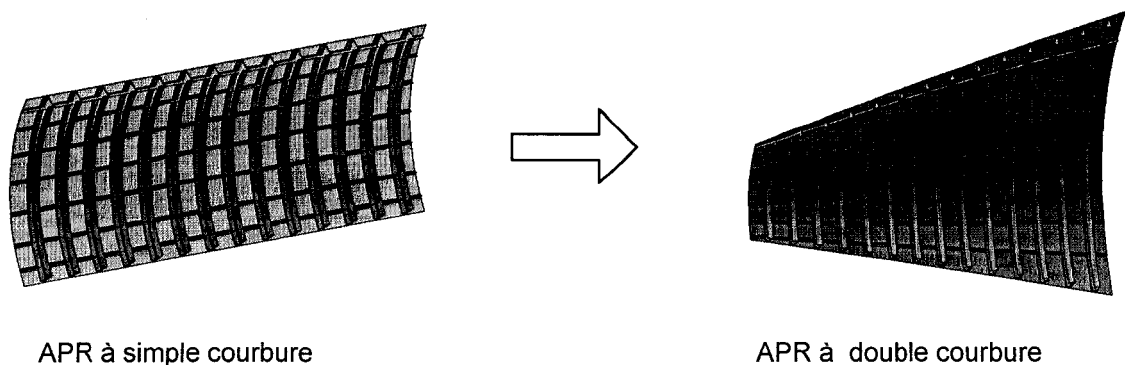


Figure 61 APRs basés sur des surfaces à simple et double courbure

Nous avons donc construit l'assemblage type sur lequel nous allons développer notre application d'insertion et de modification automatique de poches d'allègement.

4.18 Création de la caractéristique poche à l'aide d'un UDF

Cette section est dédiée au processus de création d'un UDF qui se divise essentiellement en deux parties soit : la définition de l'UDF et l'instanciation de l'UDF.

4.18.1 Définition de l'UDF poche

Maintenant que nous avons développé la méthode de création d'une poche d'allègement en mode interactif, il faut créer la caractéristique poche. Pour ce faire, nous utilisons l'outil UDF de Catia V5. Le tableau VI décrit les étapes nécessaires au processus de création de l'UDF.

Tableau VI

Étapes de création de la caractéristique poche à l'aide d'un UDF

Étape #1 :

À l'aide de l'interface de création des UDFs « *Userfeature Definition* », on insère tous les éléments géométriques constituant la caractéristique et nous définissons les éléments géométriques d'entrées nécessaires à son instantiation dans le contexte.

Éléments géométriques et paramètres internes au UDF

Entrées nécessaires à l'instanciation du UDF

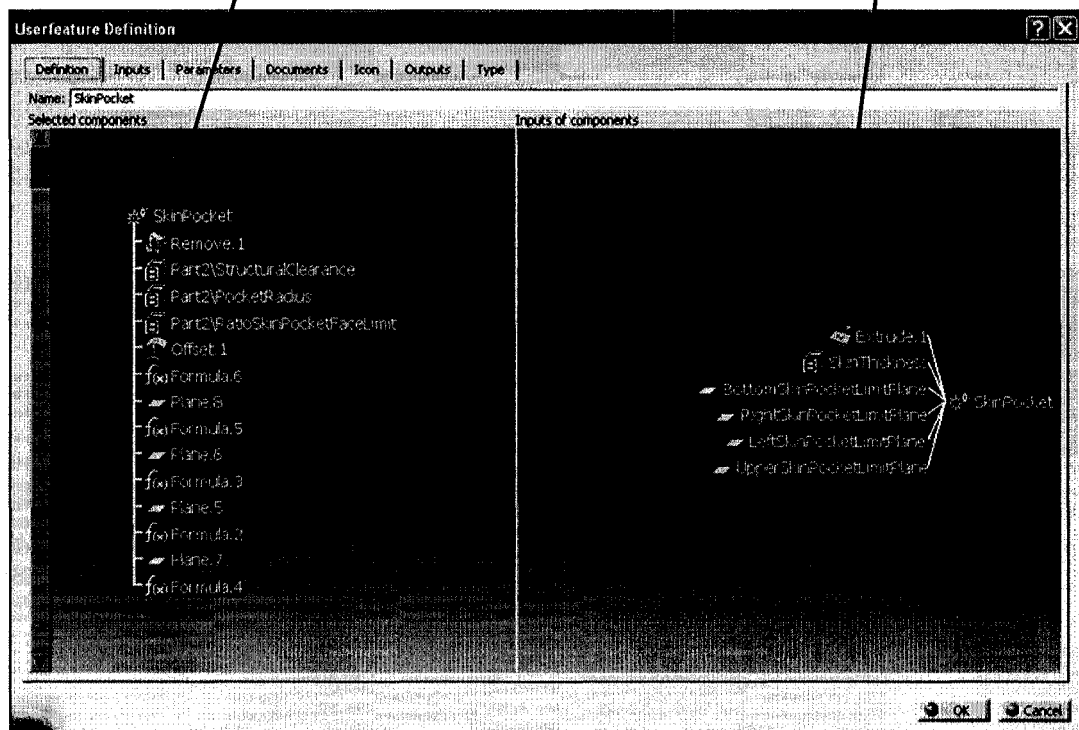


Tableau VI (suite)

Étape #2 :

Dans le deuxième onglet « *Inputs* » de l'interface, nous pouvons changer les noms désignant les entrées de l'UDF. Dans notre exemple, nous avons modifié le nom de la surface d'entrée « *Extrude.1* » pour « *OML* », qui est plus significatif pour l'utilisateur lors de l'instanciation de l'UDF.

Changement des noms des entrées

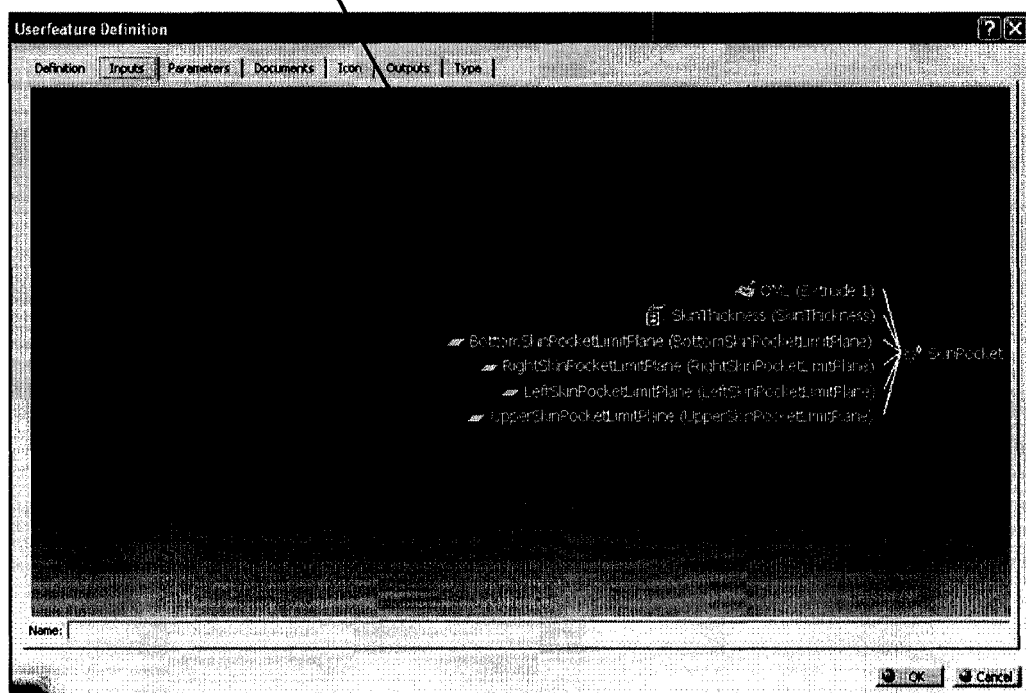


Tableau VI (suite)

Étape #3 :

Le troisième onglet de l'interface « *Parameters* » nous permet de publier certains paramètres afin de les rendre accessibles à l'utilisateur après l'instanciation. Dans notre exemple, il s'agit de la dimension de localisation « *StructuralClearance* », le rayon standard pour les congés de raccordement que nous nommons « *PocketRadius* » ainsi que le ratio permettant de définir le fond de la poche « *RatioSkinPocketFaceLimit* ».

Publication des paramètres de la poche

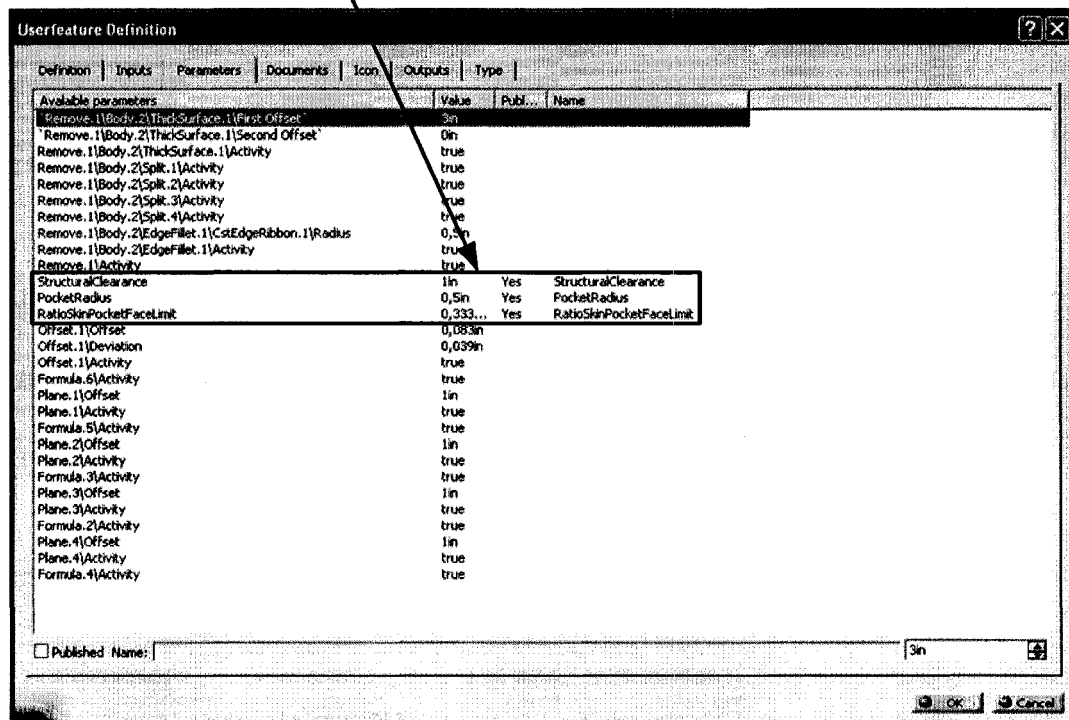
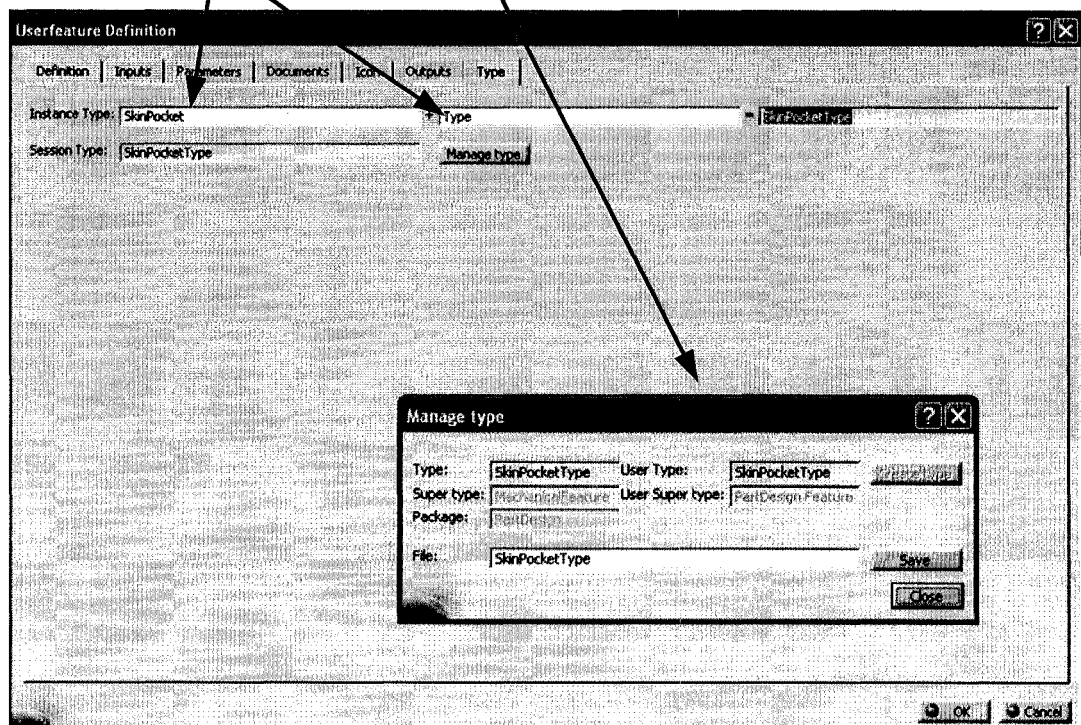


Tableau VI (suite)

Étape #4 :

Cette dernière étape, réalisée sous l'onglet « *Type* » de l'interface, consiste à typer la caractéristique pour l'intégrer dans la librairie standard d'entités du système, ce qui nous permettra d'effectuer des recherches en pointant le type « *SkinPocketType* ».

Définition et sauvegarde du type de la caractéristique

4.18.2 Test d'instanciation de l'UDF poche

Maintenant que l'UDF poche est créé dans un modèle simulant le contexte, il est temps de tester son instanciation.

Dans un premier temps, nous effectuons un test au sein du modèle qui a servi à sa définition, car tous les éléments géométriques nécessaires à l'instanciation y sont déjà présents. L'interface d'instanciation de l'UDF, « *Insert Object* », est démarrée en sélectionnant « *SkinPocket object* » puis « *Instanciation* » du menu contextuel obtenu en cliquant avec le bouton droit de la souris sur le nom de l'UDF apparaissant dans l'arbre des spécifications. Par la suite, pour chacune des entrées de l'UDF, il faut explicitement indiquer à quel élément géométrique du contexte elle correspond. Le modèle employé lors de la création de l'UDF est montré dans la partie inférieure de l'interface d'instanciation, aidant l'utilisateur à faire la correspondance, voir la figure 62 pour les détails.

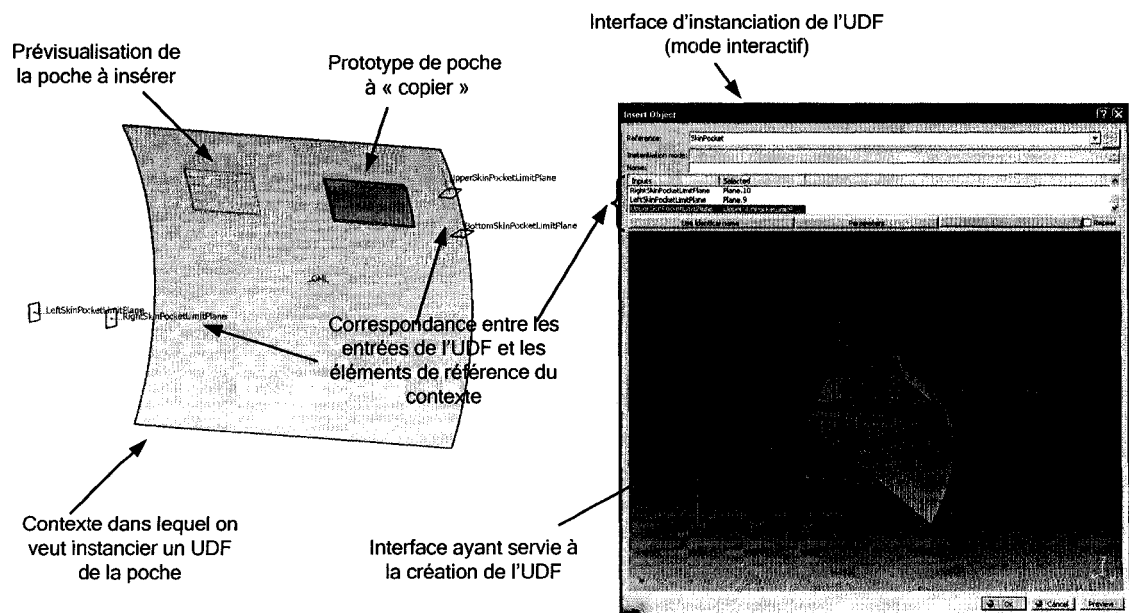


Figure 62 Test d'instanciation de l'UDF à l'intérieur de son modèle de création

Le but de cette manipulation est de s'assurer que le logiciel arrive à connecter les entrées de la caractéristique poche avec les éléments géométriques du contexte et que le comportement post-instanciation correspond à ce que nous nous attendons. Si ce test échoue, il est évidemment inutile d'aller plus loin et il faudra déverminer l'UDF.

Il est à noter que nous avons répété ces étapes à maintes reprises durant le développement de l'application afin de trouver la combinaison d'éléments géométriques à utiliser et l'ordre des opérations.

Une fois ce test réussi, il faut, dans un deuxième temps, tester l'instanciation dans le contexte de l'APR type que nous avons construit. Ce test est décisif, car il permet de terminer la validation de l'UDF dans un contexte où les éléments géométriques sont différents de ceux utilisés dans le modèle ayant servi à sa définition.

Il faut donc activer le modèle de la tôle dans l'assemblage et définir l'objet de travail, « *Define In Work Object* », comme étant le « body » de la tôle de revêtement. L'interface d'instanciation est la même qu'à la figure 62, excepté qu'elle est démarrée à l'aide de la fonction « *Instantiate From Document* » du menu « *Insert* » de Catia V5. Une fois la correspondance des entrées établie, nous obtenons une poche d'allègement dans la tôle au quadrilatère formé des lisses et des cadres indiqués. De plus, un nouvel objet, portant le nom de l'UDF et indexé du nombre d'instance, soit « SkinPocket.1 » dans notre cas, est ajouté dans l'arbre de spécifications, voir la figure 63 ci-dessous.

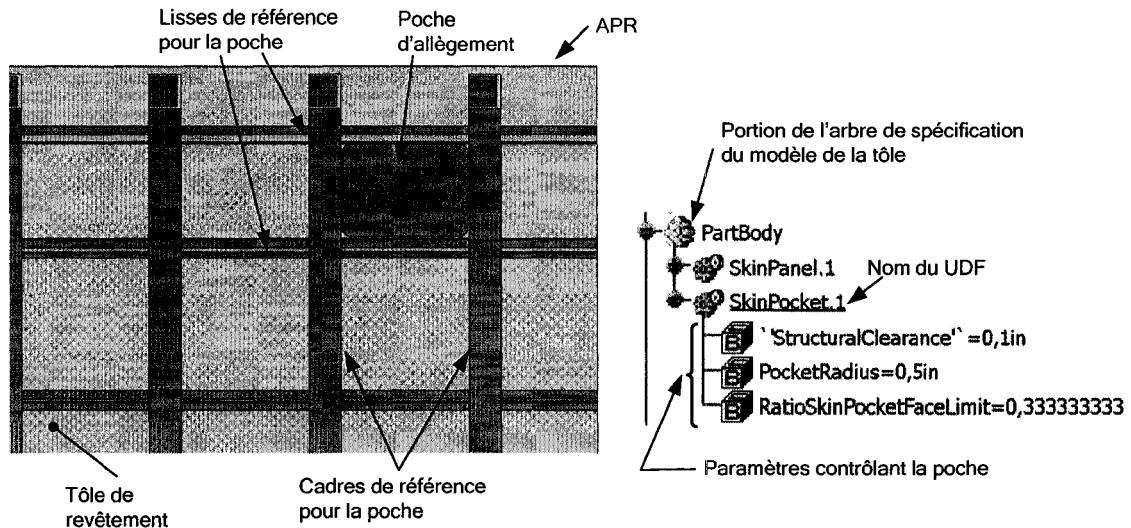


Figure 63 Instanciation de la poche dans l'APR type (surface simple courbure)

L'instanciation de la poche dans notre assemblage type a très bien réussi. On peut voir dans la portion de l'arbre des spécifications, à la droite de la figure 63, le nom de l'UDF et, en dessous, les paramètres permettant de modifier la géométrie et le comportement de la poche. Le paramètre « *StructuralClearance* » correspond à la dimension de localisation du contour de la poche et il est actuellement le même pour les lisses et les cadres. Il serait toutefois possible de publier séparément deux paramètres en modifiant l'UDF de la poche. Le paramètre « *PocketRadius* » correspond au rayon utilisé pour les congés de raccordement. Le paramètre « *RatioSkinPocketFaceLimit* » contrôle la profondeur de la poche à l'aide d'un ratio de 33% de l'épaisseur de la tôle. Ces trois paramètres sont, par défaut, indépendant les uns des autres afin de maximiser la souplesse de l'outil. Il est cependant possible d'établir un lien avec un paramètre « maître » à la tête du modèle de la tôle qui contrôlerait l'ensemble des « *StructuralClearance* », par exemple. C'est d'ailleurs la stratégie adoptée pour la programmation de notre application d'insertion automatique des poches, abordés à la section suivante.

Parmi les objectifs spécifiques visés par l'application, mentionnés à la section 4.1, l'application de définition automatique des poches d'allègement doit fonctionner pour des APRs basés sur des surfaces à simple et double courbure.

Grâce aux tests effectués précédemment, nous savons que la méthode fonctionne bien pour un APR basé sur une surface à simple courbure. Voyons maintenant si notre caractéristique poche, sous forme d'UDF, fonctionne pour un APR basé sur une surface à double courbure. Pour ce faire, nous changeons simplement la surface pointée par la publication de l'OML dans le modèle « Mold Lines » et Catia V5 propage le changement automatiquement.

Nous obtenons donc un APR basé sur une surface à double courbure. Par la suite, nous ajoutons deux autres instances de poches dans l'APR, une au centre et une autre à l'extrémité de l'APR où le rayon de courbure est le plus petit, voir la figure 64 pour le résultat. Le passage d'une configuration d'APR à simple courbure vers une configuration d'APR à double courbure est très simple et requiert peu de temps de calcul même avec la présence de poches.

Les vues agrandies des trois instances de poches de la figure 64 nous permettent de constater que le contour des poches suit parfaitement le parcours des lisses tout en respectant la dimension de localisation.

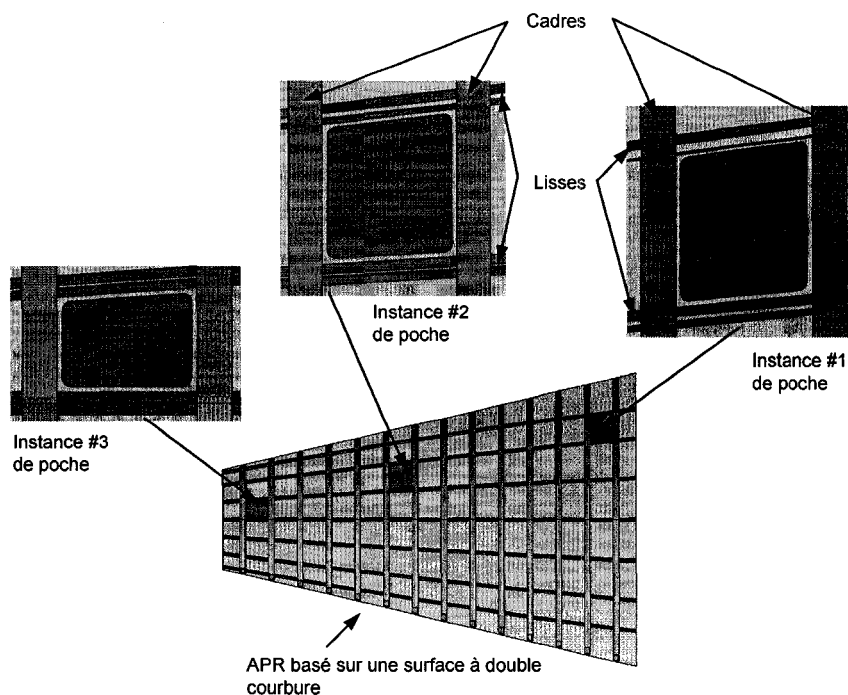


Figure 64 Instanciation des poches dans l'APR type (surface double courbure)

La structure de notre assemblage, basée sur l'emploi structuré d'associations persistantes, permet à l'utilisateur de se concentrer sur son travail de conception et de consacrer beaucoup moins de temps en modélisation géométrique, ce qui est un avantage indéniable.

Nous avons jusqu'à maintenant établie la marche à suivre pour créer une poche d'allègement à l'aide d'une méthodologie de conception concise en 6 étapes sous Catia V5. Nous avons également créé un UDF encapsulant cette nouvelle caractéristique. Finalement, nous avons testé son instanciation dans notre assemblage type à l'aide de l'interface appropriée, le tout en mode interactif.

4.19 Programmation de l'insertion automatique des poches dans l'APR

Nous avons déjà gagné beaucoup de temps en créant une caractéristique poche à l'aide d'un UDF encapsulant la géométrie ainsi que le tissu de contraintes entre les éléments

géométriques. Cette caractéristique peut être instanciée dans les APR à simple et double courbure en indiquant les entrées pertinentes du contexte plutôt qu'en recréant une poche à chaque quadrilatère formé par les lisses et les cadres. Toutefois, l'instanciation manuelle de chaque UDF poche nécessite le branchement de six entrées. Pour l'APR type contenant 78 poches, cela fait un total de 468 contraintes à établir explicitement. Il est possible d'obtenir un gain de temps supplémentaire en automatisant le processus d'instanciation des UDFs poche dans l'APR.

Cette section est dédiée à la description du programme d'instanciation automatique des poches que nous avons développé. L'utilisateur n'a alors qu'à pointer les pièces de référence avec lesquelles il faut établir des liens.

En deuxième lieu, le programme permet l'insertion automatique de toutes les poches requises dans l'APR sans l'intervention de l'utilisateur. Ces deux tâches seront programmées à l'aide d'un langage permettant la manipulation des éléments géométriques.

Sous Catia V5, il est possible d'utiliser plusieurs plateformes de programmation. La *RADE*, « *Rapid Application DEveloppement* », tout d'abord, utilisant le langage C++, est l'outil de développement le plus puissant, le plus coûteux, mais surtout le plus compliqué. À cause de l'investissement de temps nécessaire pour le maîtriser, cette plateforme constitue notre dernier choix. En deuxième lieu, il est possible d'utiliser le langage script de Catia V5, mais ce dernier est de trop bas niveau pour nos besoins.

Nous avons donc opté pour le langage Visual Basic afin de développer notre application. Ce choix est guidé par la simplicité du langage et les possibilités qu'il nous offre en fonction du peu de temps requis pour s'appropriier le langage et obtenir des résultats. La figure 65 expose l'organigramme général de fonctionnement de notre application.

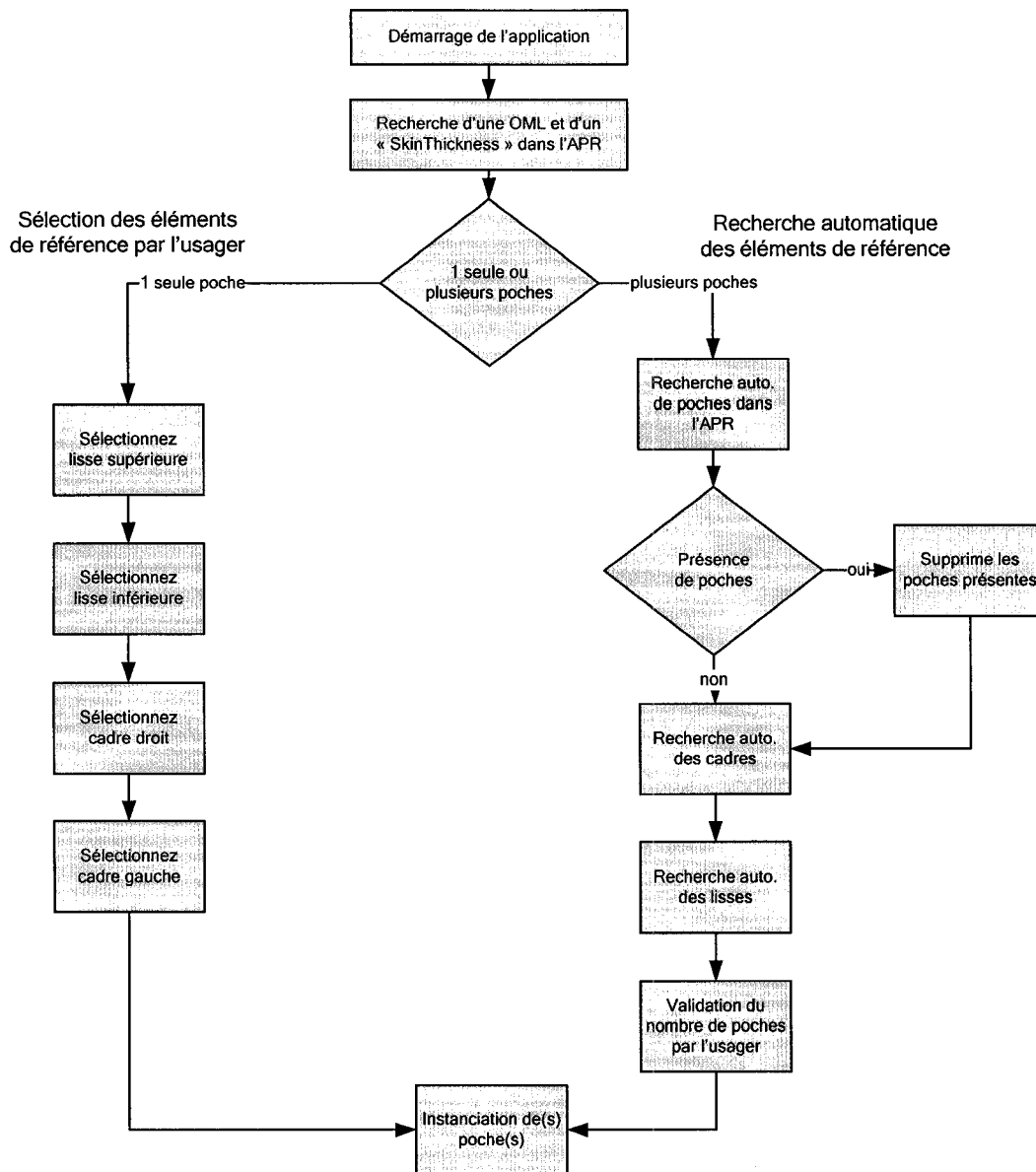


Figure 65 Organigramme général de fonctionnement de l'application

Tout d'abord, l'application vérifie la liste des paramètres publiés dans chaque produit pour trouver une OML et un paramètre définissant l'épaisseur de la tôle, « SkinThickness ». S'il y avait absence de ces deux objets, l'application en informerait l'utilisateur et se terminerait. L'application est conçue pour fonctionner avec une seule OML et un seul paramètre d'épaisseur de tôle.

Ensuite, l'application demande à l'utilisateur s'il veut créer une seule poche ou toutes les poches de l'APR. Dans le cas d'une seule poche, l'utilisateur est guidé pas à pas afin de lui permettre d'indiquer les pièces de références adjacentes à la poche qu'il veut créer. Pour chacune des pièces nécessaires, une boîte de dialogue apparaît pour l'informer de l'objet qu'il doit indiquer. De plus, une fois qu'il a cliqué sur le bouton « OK », un message apparaît, en bas à gauche de l'interface de Catia V5, indiquant ce que le système attend de lui, voir la figure 66 ci-dessous.

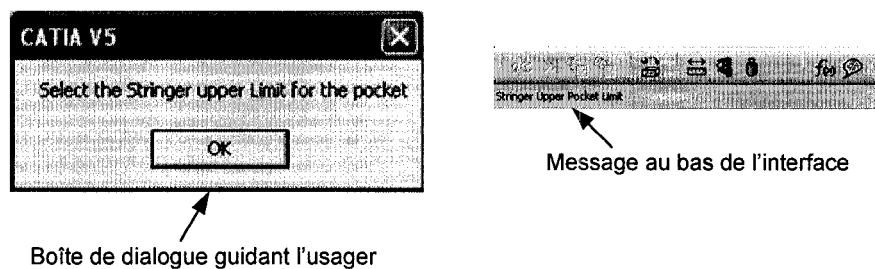


Figure 66 Boîte de dialogue et messages guidant l'utilisateur lors de l'exécution de l'application

Finalement, lorsque l'application demande à l'utilisateur d'indiquer un cadre, par exemple, elle n'accepte que ce type d'entrée et lorsque l'utilisateur déplace son curseur sur les éléments de l'assemblage, seuls les cadres seront mis en surbrillance. Cela est possible grâce aux différents types que nous avons attribués aux UDFs des pièces.

Une fois toutes les entrées fournies, l'application procède à l'insertion de la poche dans le quadrilatère indiqué. À l'instanciation, l'application relie automatiquement les entrées nécessaires à aux éléments géométriques indiqués par l'utilisateur. Ce processus est complètement transparent à l'utilisateur.

La deuxième option présentée à l'utilisateur est la possibilité d'insérer toutes les poches d'allègement pouvant être contenues dans l'APR. Lorsque l'utilisateur choisit cette option, l'application commence par vérifier la présence de poches dans l'APR à l'aide d'une fonction de recherche disponible sous Catia V5 et accessible via les APIs.

La figure 67 montre l'interface, en mode interactif, afin de permettre au lecteur de comprendre le processus. Nous indiquons l'atelier dans lequel se trouve le type d'objet que l'on recherche ainsi que son nom. Dans l'exemple de la figure 67, le système a trouvé trois occurrences de poches, les affiche en surbrillance dans l'arbre des spécifications et nous retourne leur chemin dans l'arbre.

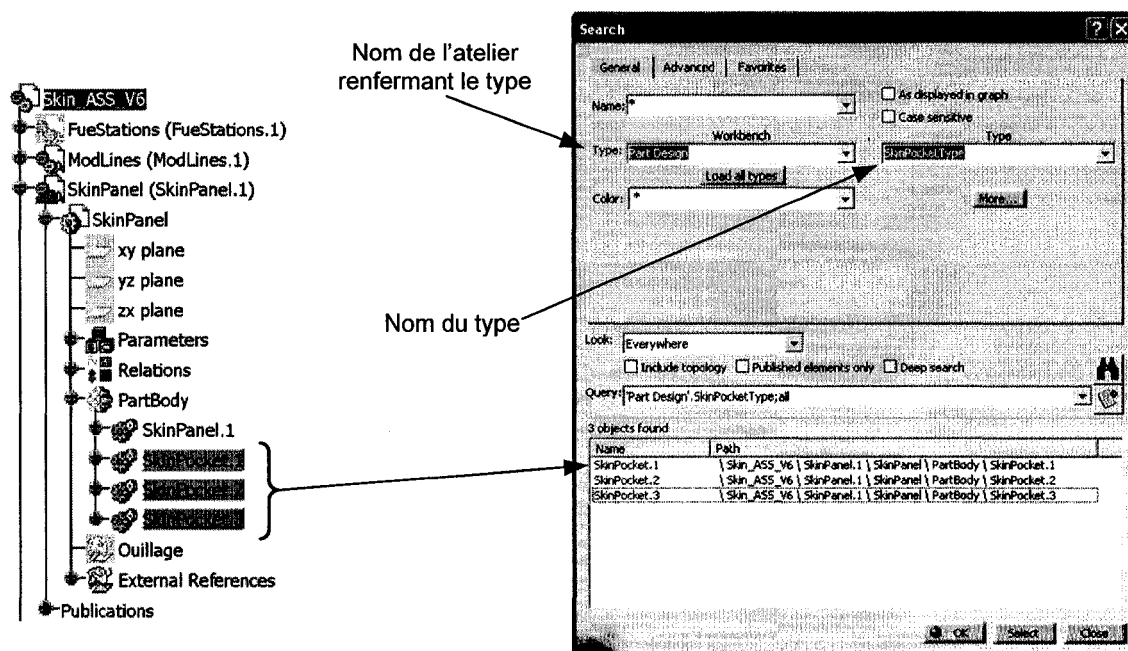


Figure 67 Interface de recherche d'entités dans Catia V5

Il est possible d'utiliser cet outil à l'aide de code VB et, comme nous avons utilisé des UDFs typés pour définir notre caractéristique poche ainsi que les lisses, les cadres et la tôle de revêtement, il est très facile de les trouver dans l'assemblage pour ensuite les manipuler à notre guise. Cette méthode simplifie considérablement le processus de développement, car elle évite de naviguer dans l'arbre de spécifications pour trouver les pièces dont nous avons besoin pour l'insertion des poches dans l'APR. Il nous suffit de lancer une recherche pour trouver toutes les instances spécifiées dans le modèle.

Lorsque l'application détecte la présence de poches dans l'APR, elle demande à l'utilisateur s'il veut les supprimer. Par la suite, une recherche de tous les cadres et de toutes les lisses est effectuée pour calculer le nombre total de poches à insérer. L'application demande ensuite à l'utilisateur de confirmer le lancement de la commande, voir figure 68 ci-dessous.

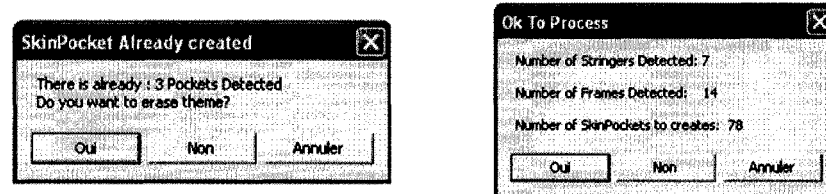


Figure 68 Boîtes de dialogue informant l'utilisateur sur le nombre de poches

Une fois le processus lancé, l'application insère une poche à chaque quadrilatère formé par les pièces de référence et lie les paramètres de la poche aux paramètres « maîtres » du modèle de la tôle. Il sera par la suite facile de les personnaliser, si nécessaire, pour une ou plusieurs poches.

L'APR type utilisé pour nos tests est constitué de 7 lisses et de 14 cadres, ce qui fait un total de 78 poches à insérer. L'application développée requiert 5 minutes et 55 secondes pour faire le travail, ce qui fait une moyenne de 4,6 secondes par poche. Ce test a été effectué sur un ordinateur de table muni d'un processeur d'architecture AMD et d'une

mémoire vive de 1024 Mo. Il est à noter qu'un petit délai a été inséré afin de permettre à la carte graphique d'afficher chaque poche pendant le processus.

La figure 69 ci-dessous, montre les deux types de panneaux avec les poches d'allègement.

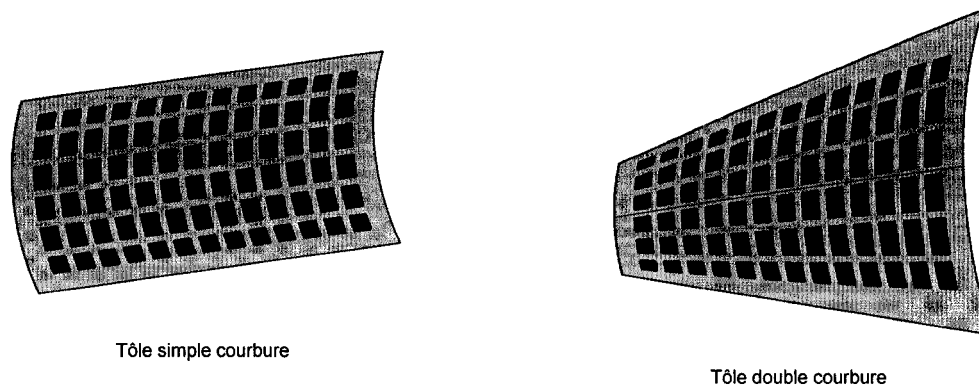


Figure 69 Tôles de revêtement avec les poches d'allègement

De la même façon que décrit précédemment, il suffit de modifier la surface pointée par la publication « OML » pour passer d'une configuration d'APR à simple courbure pour une configuration à double courbure. La présence des 78 poches demande 2 minutes de temps de calcul pour le passage d'une configuration à l'autre, au lieu de 30 secondes sans les poches.

4.20 Synthèse et conclusion relative au développement de l'application

L'application de création, d'insertion et de mise à jour automatique des poches d'allègement au sein d'un APR fonctionne. Les résultats sont discutés en détails au chapitre 5.

Nous avons atteint les objectifs spécifiques que nous nous étions fixés à la section 4.1. En effet, l'application insère des poches d'allègement dans des APRs basés sur des

surfaces à simple et à double courbure en mode pas à pas avec l'utilisateur et de façon complètement automatique.

Le savoir-faire associé à la conception des poches d'allègement est encapsulé dans la caractéristique poche sous forme de règles de conception et certains paramètres sont accessibles à l'utilisateur afin qu'il puisse en contrôler la géométrie.

Comme nous le verrons plus loin, le concept d'empreinte et de pseudo-empreinte est implicitement contenu lorsque l'on définit les liens entre les pièces nécessaires à l'établissement des contraintes situées à l'intérieure même de la caractéristique poche qui est personnalisé à l'aide des UDFs. Dans le contexte des soyages et des découpes de Giguère, l'empreinte et la pseudo-empreinte étaient explicitement établies et traitées à l'aide de routines procédurale sous VB. De plus, les associations persistantes exploitées par Catia V5 permettent la propagation automatique des changements au sein de l'assemblage et assure la cohérence des données.

Nous avons capitalisé également sur les outils de Catia V5 en utilisant entre autre, la capacité de «paramétrisation» de l'outil, les UDFs, la possibilité de les typer, la grammaire de recherche de Catia V5, les associations persistantes ainsi que les fonctions de publication des modèles et des UDFs.

Finalement, nous avons minimisé le nombre de lignes de code requises pour notre application car elle ne crée aucun élément géométrique, mais manipule plutôt des caractéristiques, i.e. des UDFs de poche.

Le premier objectif de ce travail, consistant à contribuer à démontrer le caractère générique des concepts de typages des entités et celui d'empreintes et de pseudo-empreintes, est atteint. Le chapitre 5 qui suit traitera de l'analyse et la discussion des résultats.

CHAPITRE 5

ANALYSE DES RÉSULTATS ET DISCUSSION

Ce chapitre est consacré à l'analyse des résultats obtenus à l'aide du prototype d'application. La première partie du chapitre présente les données. Elle est suivie d'une discussion permettant de confirmer que les concepts vus dans les chapitres précédents sont bien supportés par les outils disponibles et qu'ils s'appliquent efficacement à notre cas d'application des poches d'allègement.

La deuxième partie du chapitre traite du potentiel de déploiement de ces outils à l'échelle industrielle en discutant les questions posées au chapitre 1 et, à défaut d'y apporter une réponse définitive, y jette un éclairage nouveau.

5.1 Analyse des résultats obtenus à l'aide du prototype d'application d'insertion automatique des poches d'allègement dans un APR

Après avoir développé notre application, nous avons effectué différents tests de performances reliés aux temps de modélisation des pièces (OML, tôle, lisses, cadres) et de la caractéristique poche en tant qu'UDF. Nous avons également mesuré les temps requis pour construire l'APR type, pour y insérer les poches d'allègement, manuellement et à l'aide de notre application, et nous avons mesuré le temps requis par Catia V5 pour propager dans l'APR des modifications faites sur les pièces et sur les poches.

5.1.1 Clarifications sur les temps calculés

Avant de débiter, quelques clarifications sont nécessaires. Dans les sections qui suivent nous allons procéder à une série de comparaisons démontrant que l'utilisation des concepts énoncés au chapitre 3 permettent d'obtenir des gains de temps et de coûts significatifs durant le processus de conception. Toutefois, afin d'être le plus objectif et le

plus fidèle à la réalité possible, nous tenons à préciser les différentes activités que nous comptabilisons dans nos résultats.

L'ensemble des activités de développement du prototype mentionnées dans ce travail, i.e. le temps d'apprentissage de l'outil Catia V5 en général, le développement des méthodologies de modélisation des pièces, la création des pièces constituant l'APR et de la caractéristique poche à l'aide d'UDFs, ainsi que la programmation de l'application s'est échelonné sur plusieurs mois. Nous ne disposons pas de données objectives concernant les activités de développement supportant les concepteurs lors de leur utilisation de l'outil Catia V4 chez Bombardier Aéronautique. C'est pourquoi, dans les comparaisons qui suivent, nous considérerons uniquement les temps d'exploitation et non les temps requis pour le développement des solutions.

Nous présenterons les mesures de temps effectuées sur nos activités de développement. Ils se traduisent par une économie de temps substantielle lors de l'usage de l'application. De plus, les connaissances des outils de Catia V5 acquises durant ce processus sont facilement transposables à d'autres contextes permettant de capitaliser largement sur le temps investi.

Un autre aspect à souligner est que le travail décrit dans ce mémoire est l'œuvre d'une seule personne laissée à elle-même avec Catia V5 et la documentation d'aide disponible et que ce travail s'est déroulé dans un contexte académique. Lors de notre stage chez Bombardier Aéronautique à l'été 2003, nous avons constaté que les ressources mises à la disposition des gens responsables du soutien aux usagers des outils CAO étaient beaucoup plus importantes que celles dont nous disposions lors de notre développement. Nous pensons donc que le temps requis pour développer une application fonctionnelle dans un contexte industriel diminuerait significativement. Cela rendrait l'utilisation des UDFs et l'automatisation de tâches de modélisation encore plus rentables.

Finalement, notre séjour chez Bombardier Aéronautique nous a permis d'apprendre que très peu de processus de modélisation sont automatisés à l'aide de Catia V4 et par le fait même, la grande majorité des activités de modélisation et des changements d'ingénierie se font manuellement [8]. La raison évoquée pour ce choix est le coût lié au maintien des programmes maison développés sous Catia V4, qui serait trop élevé en fonction des gains obtenus.

5.1.2 Adaptation de l'assemblage type

Pour pouvoir comparer nos résultats avec ceux obtenus par Michaud [6], nous avons adapté notre assemblage type afin que l'on y retrouve le même nombre de pièces structurales qu'il a utilisé dans son travail. Initialement, notre APR contenait 7 lisses et 14 cadres totalisant 78 poches. Le tableau VII, ci-dessous, détaille la composition de notre assemblage type avant et après l'adaptation.

Tableau VII

Nombre de pièces composant l'assemblage type avant et après l'adaptation

Types de pièces	Avant adaptation	Après adaptation
APR	1	1
Nombre de tôles	1	1
Nombre de lisses	7	9
Nombre de cadres	14	11
Nombre de poches	78	80

5.1.3 Temps de création des UDFs et de l'APR type

Nous avons relevé les temps requis pour la construction des différents modèles constituant l'APR type, soit ceux des surfaces maîtresses, des cadres, des lisses, de la tôle de revêtement et celui de la poche d'allègement. Rappelons que cette construction est faite une seule fois de façon manuelle. Une fois les pièces complétées, elles sont encapsulées dans des UDFs qui par la suite sont instanciées pour former l'APR type. Le tableau VIII ci-dessous résume les temps de création de ces modèles et ceux nécessaires à la création de leur UDF.

Cette information donne une idée du temps nécessaire à la création de modèles pouvant être encapsulés dans des UDFs une fois que l'on maîtrise ce type d'outil.

Tableau VIII

Temps de création des modèles et des UDFs de l'APR type

Modèles	Temps de création des modèles (mn)	Temps de création des UDFs (mn)
Surface maîtresse	35	5
Cadre	65	10
Lisse	55	10
Tôle de revêtement	15	5
Poche d'allègement	35	10
Total	205	40

Une fois que les modèles des pièces et que les UDFs de Catia V5 ont été créés, on construit l'assemblage type. Cette opération prend 23 minutes, incluant le temps nécessaire pour insérer les plans de F.S. et de positionnement pour les lisses, les surfaces OML ainsi que le panneau de revêtement (cette étape pourrait éventuellement être éliminée).

Un fait à souligner est que lors de l'instanciation des UDFs tôle, lisse et cadre dans l'assemblage type, aucune entité géométrique de bas niveau n'a été manipulée. De plus, grâce à la commande de copier / coller, compatible avec les UDFs, il nous était très facile de les multiplier. Nous n'avons qu'à remplacer un seul lien contextuel d'entrée, celui déterminant la position de la lisse ou du cadre en l'occurrence, et de renommer les sorties appropriés. Cette fonction nous fit économiser beaucoup de temps en mode manuel.

Notons également que notre application d'insertion automatique des poches pourrait être facilement adaptée pour automatiser l'instanciation des pièces de structure telle que les lisses et les cadres.

5.1.4 Comparaisons des temps de création des poches

Dans son travail de méthodologie unifiée de modélisation, Michaud [6] a comparé les temps requis pour insérer 80 poches d'allègement dans un APR selon sa propre méthode de soustraction. Ainsi, il prend 43 minutes pour insérer 80 poches sous Catia V4 et 115 minutes sous Catia V5, sans modifier n'y adapter sa méthode aux nouvelles fonctionnalités de Catia V5 [6].

Nous nous sommes prêtés à un exercice semblable et avons utilisé la méthodologie de conception de poches d'allègement en six étapes que nous avons mise au point pour ce travail. Rappelons qu'elle consiste à utiliser des plans, délimitant les pièces structurelles et permettant d'obtenir la forme géométrique des poches, pour découper un solide issu d'une surface qui sera soustraite ensuite du volume de la tôle.

L'insertion des 80 poches dans notre APR type à l'aide d'instance d'UDFs prend 45 minutes, tandis que l'utilisation de notre application d'insertion automatique prend 4,7 minutes pour réaliser le même travail. En comparant le temps requis pour insérer automatiquement les 80 poches à celui nécessaire pour les insérer après avoir

personnalisé l'outil CAO, par l'utilisation d'instances d'UDFs, nous obtenons une réduction de temps de l'ordre de 90%, voir le tableau IX.

En comparant maintenant nos résultats, sous Catia V5, à ceux de Michaud, le simple fait d'adapter la méthodologie d'insertion de poche et de personnaliser cette tâche à l'outil CAO nous permet d'obtenir un gain de temps de 61%. En utilisant l'application automatique nous obtenons un gain de temps de 95,9%.

En comparant nos résultats à ceux obtenus par Michaud sous Catia V4, nous constatons que nous sommes arrivés sensiblement au même résultat. Toutefois, l'utilisation de notre application automatique nous permet d'économiser 89% du temps par rapport à sa méthode et à l'utilisation de Catia V4.

Notre stage chez Bombardier [8] nous a permis d'établir qu'un avion de transport Régional, tel qu'un CRJ-700 de 70 places, contenait au total 2500 poches d'allègement dans les panneaux à simple et double courbure recouvrant le fuselage central, avant et arrière, mais aussi dans les ailes et les empennages horizontaux et verticaux. Nous avons appris que les concepteurs en place prenaient environ 5 minutes pour modéliser une poche. Ce temps est basé sur la modélisation d'un APR appartenant à la première famille de tôle classifiée par Michaud [6], voir le tableau I, dont les éléments de structure sont droits formant ainsi des poches aux profils rectangulaires. Ce type de panneau est le plus simple que l'on peut retrouver dans l'ensemble des APRs de l'aéronef. Il est également à noter que la modélisation d'une poche d'allègement dans le cas de la troisième famille répertoriée par Michaud [6], dont le contour est complexe, peut prendre jusqu'à 45 minutes.

Tableau IX

Comparaison des résultats de Bombardier,
Michaud et Lamarche

	Bombardier	Michaud		Lamarche	
	Catia V4	Catia V4	Catia V5	Catia V5	
	Manuel	Manuel		Person- nalisation.	Auto.
	min	min	min	min	min
Insertion des 80 poches	320	43	115	45	4,7
Gains obtenus par Lamarche (mode manuel)	86%	-	61%	-	-
Gains obtenus par Lamarche (mode auto.)	98,5%	89%	95,9%	90%	-

En comparant les résultats obtenus à l'aide de la personnalisation des outils de Catia V5 (utilisation des UDFs entre autre) à ceux de Bombardier sous Catia V4, nous constatons un gain de temps de l'ordre de 86%. L'utilisation de notre application automatique, quand à elle, nous permet d'obtenir des gains de l'ordre 98,5%, voir le tableau IX pour le résumé des résultats.

En comparant les résultats que nous avons obtenus sous Catia V5 par rapport à ceux de Michaud sous Catia V4 (45 min. versus 43 min.), on constate que l'utilisation efficace des outils de Catia V5 donne des résultats similaires à Catia V4. En comparant nos résultats à ceux de Michaud sous Catia V5 (45 min. versus 115 min.), on constate qu'en laboratoire, l'utilisation efficace des concepts sous la forme d'UDFs permet un gain significatif.

Finalement, nos résultats confirment l'une des conclusions de Michaud [6] au sujet des méthodologies de conception lors de la migration vers un nouvel outil CAO. En effet, celles-ci doivent être adaptées à la nouvelle plateforme afin de bénéficier de la migration.

5.2 Temps requis pour apporter certaines modification à l'APR

Nous avons également procédé à une série de modifications sur l'assemblage type contenant les poches d'allégement afin de démontrer les performances de Catia V5 lors de la propagation de corrections et de modifications au sein de l'APR. Cet exercice est réalisé en mode interactif et repose sur la technologie des liens persistants établis au sein de l'assemblage.

Nous ne disposons pas de données numériques provenant de chez Bombardier pour la propagation de changements sous Catia V4. Toutefois, compte tenu du faible usage de liens persistants sous Catia V4, les modifications influençant la géométrie d'une pièce ou d'une caractéristique engendrent souvent la « re-modélisation » complète ou partielle de celle-ci. C'est ce que Michaud [6] a posé comme hypothèse dans son travail sur le développement d'une méthodologie de modélisation unifiée pièce-outillage en CAO aéronautique, appliquée au cas des gabarits de découpes pour les tôles de revêtement. Il reprend donc les temps de modélisation complet sous Catia V4 pour les comparer aux changements qu'il a effectués sous Catia V5.

Le tableau X, à la page suivante, expose les résultats que nous avons obtenus lors de différents scénario de modification de l'APR.

Tableau X

Temps requis pour propager certaines modifications dans l'APR type sous Catia V5

Types de modifications	Nombres d'éléments impactés							Temps sec.
	Lisses	Cadres	Tôle	Poches	OML	F.S.	Total	
Angle de la OML	9	11	1	80	1	0	102	55
Surface OML (simple à double courbure)	7	11	1	80	1	0	100	92
Position d'un cadre (F.S.)	0	1	1	8	0	1	11	155
Espacement entre les cadres (F.S.)	0	11	1	80	0	11	103	57
Position d'une lisse (plans)	1	0	1	10	0	1	13	110
Espacement entre les lisses (plans)	9	0	1	80	0	8	98	101
Dimension localisation								
Une poche	0	0	1	1	0	0	2	40
Toutes les poches	0	0	1	80	0	0	81	50
Rayon de congé de raccordement								
Une poche	0	0	1	1	0	0	2	30
Toutes les poches	0	0	1	80	0	0	81	47
Profondeur des poches								
Une poche	0	0	1	1	0	0	2	35
Toutes les poches	0	0	1	80	0	0	81	40
Épaisseur de la tôle	9	11	1	80	0	0	101	84
								Moyenne : 69

Comme nous pouvons le remarquer en analysant le tableau X ci-haut, ce ne sont pas nécessairement les modifications se répercutant sur le plus grand nombre d'éléments qui prennent le plus de temps à être complétées. En effet, la modification de la position d'un seul cadre requiert 155 secondes tandis que de modifier la distance entre l'ensemble des cadres nécessite 57 secondes. Cela s'explique par le fait que pour modifier la position d'un seul cadre, il nous faut créer un nouveau plan et modifier la publication. Par la suite, l'application recalcule le nouveau cadre et la forme des poches touchées par ce changement. La partie la plus longue de cette modification est celle qui dépend de l'interaction avec l'utilisateur. En contre partie, le temps nécessaire pour modifier l'espacement entre tous les cadres prend 3 fois moins de temps, bien que le nombre total de pièces impactées soit dix fois plus grand. Ceci s'explique par le fait que l'intervention de l'utilisateur se limite à modifier la valeur d'un seul paramètre.

Globalement les temps de modification obtenus sont très courts, en moyenne 69 secondes. Comme le mentionne Michaud [6], l'investissement au niveau de la modélisation initiale sous Catia V5 est rapidement rentabilisé dès qu'il y a des changements. Rappelons que les changements d'ingénierie sont inhérents au processus de conception et que seulement durant l'année 2001, Bombardier Aéronautique a enregistré plus de 13000 demandes de changements d'ingénierie [2].

Cet exercice nous permet de saisir le potentiel de gains que peut apporter la délégation de tâches de modélisation à la machine et à l'outil CAO. Dès qu'il y a intervention de l'utilisateur, le temps d'exécution est plus long et le risque d'erreurs est plus grand.

5.3 Les UDFs et la programmation VB comme outil de mise en œuvre des concepts

Cette section démontre de quelles façons les concepts d'associations, de typage d'entités, de liens technologiques, d'empreinte et de pseudo-empreinte sont mis en œuvre dans l'outil CAO de nouvelle génération Catia V5.

Rappelons qu'une association nous informe sur l'existence d'une dépendance entre deux objets techniques et qu'elle se présente selon trois niveaux d'abstraction, soit la relation, le lien et la contrainte, voir le tableau IV.

Dans notre cas d'application, nous nous intéressons particulièrement à la relation «Contexte d'assemblage – Tôle», car les caractéristiques poches sont instanciées dans le modèle de la tôle et elles ont besoin de tirer de l'information provenant du contexte d'assemblage afin de s'y adapter. Il est à noter que la structure de l'APR type nécessite une étape supplémentaire de décomposition au niveau de cette relation. En effet, les pièces cadres et lisses ont été agrégées dans deux sous-assemblages sous Catia V5 afin de faciliter la programmation de l'application automatique. Généralement, la règle « une pièce par modèle » est adoptée en industrie [8].

Au premier niveau d'abstraction de la relation «Contexte d'assemblage – Tôle», la tôle de revêtement est reliée à quatre sous-assemblages : «CFramesGroup», contenant les pièces cadres, «TStringersGroup», contenant les pièces lisses, «Mold Lines – OML», contenant les surfaces maîtresses et «SkinPanel», contenant la tôle de revêtement (celle-ci est reliée à elle-même car les instances de poches utilisent son épaisseur nominale), voir la figure 46 et la figure 70 a). Rappelons que cette structure est propre à notre projet d'automatiser la modélisation des poches d'allègement.

La cardinalité de ce premier niveau de la relation «Contexte d'assemblage – Tôle» correspond au cas numéro 2 (m X 1) répertorié par Tremblay [53], soit 4 X 1, voir la figure 32.

Cette relation établie entre objets techniques du niveau des sous-assemblages peut-être considérée comme établie au niveau des pièces qu'elle relie en décomposant les sous-assemblages en pièces impliquées, voir la section «b) Relation Contexte d'assemblage – Tôle (niveau d'abstraction 2)» de la figure 70. La cardinalité de la relation à ce niveau d'abstraction des objets techniques est alors de 6 X 1.

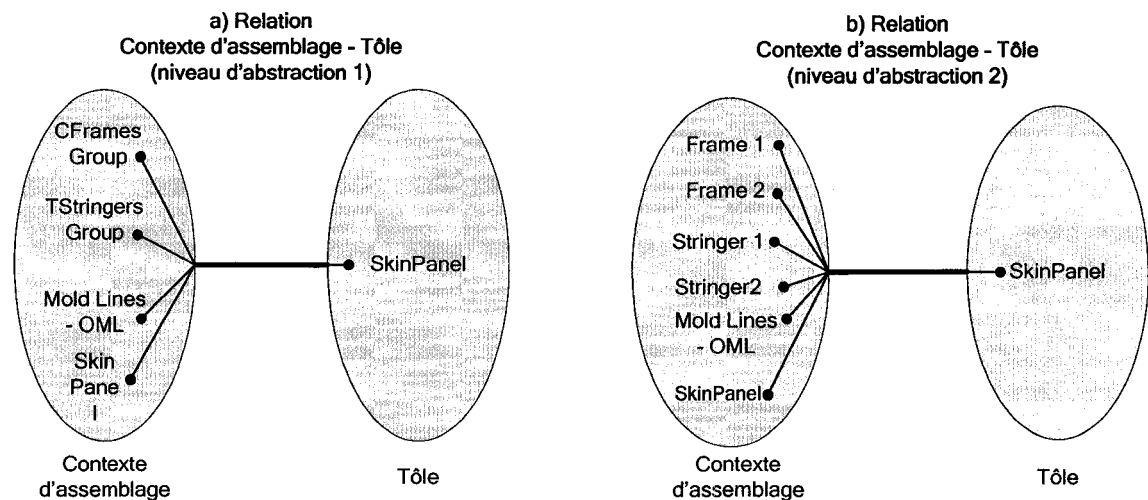


Figure 70 Les deux niveaux d'abstraction de la relation «Contexte d'assemblage - Tôle»

L'application automatique de modélisation des poches que nous avons programmée en VB opère au niveau de la relation, «Contexte d'assemblage – Tôle» en l'occurrence, pour tirer l'information dont elle a besoin afin d'accomplir sa tâche. En effet, lorsqu'on lance l'application, en mode assisté par exemple, elle détecte automatiquement la présence d'une OML et d'un « SkinPanel » puis l'utilisateur n'a qu'à pointer les deux lisses et les deux cadres entourant la poche à créer. L'application gère ensuite, dans un premier temps, la décomposition de la relation en un lien entre caractéristiques, puis,

dans un deuxième temps, la décomposition du lien en contraintes afin de relier les entités géométriques de bas niveau du contexte à ceux de la caractéristique «poche» (dans notre cas, l'instanciation d'une caractéristique poche se fait à l'aide d'UDF). Lancée en mode automatique, l'applicationinstanciera une poche à chacun des quadrilatères formés par deux cadres et deux lisses ; la relation est dans ce cas décomposée automatiquement en plusieurs liens.

La relation «Contexte d'assemblage – Tôle» se décompose en un lien «Contexte d'assemblage – Poche» qui implique des objets techniques de type caractéristique plutôt que de modèles d'assemblage et de pièces. Dans notre cas d'application des poches d'allègement, c'est un lien de dérivation qui est en fait un lien technologique spécialisé.

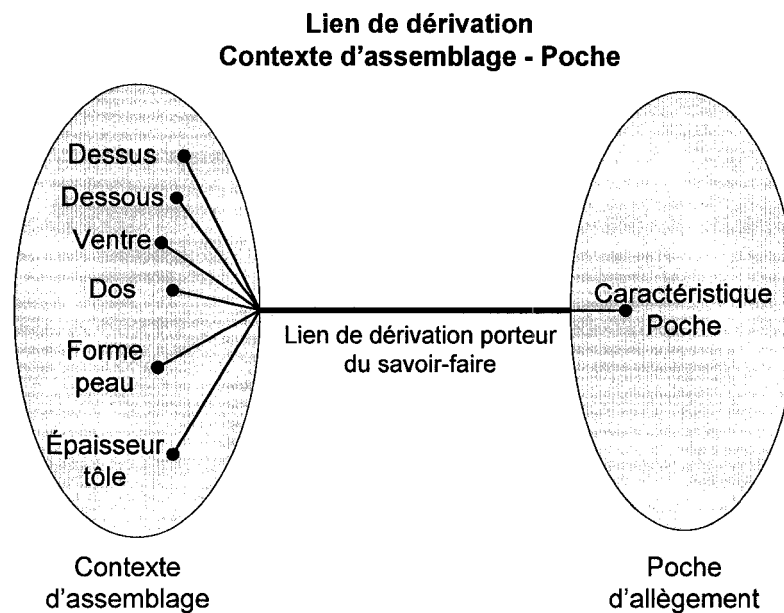


Figure 71 Lien de dérivation «Contexte d'assemblage – Poche»

Les termes contenus dans la partie «Contexte d'assemblage» de la figure 71, soit : «Dessus, Dessous, Ventre, Dos, Forme peau et Épaisseur tôle », correspondent aux caractéristiques du contexte d'assemblage permettant d'instancier une caractéristique

poche et auxquelles cette dernière sera reliée. On remarquera que la cardinalité de ce type d'association, au niveau des caractéristiques, est de 6 X 1.

Le lien de dérivation, permettant de «dériver» l'empreinte en pseudo-empreinte grâce au savoir-faire qu'il porte sous forme de règles de conception, est mis en œuvre à l'aide de l'outil UDF (User-Defined Feature) de Catia V5. En effet, sans aller à l'intérieur de l'UDF pour l'instant, on voit qu'en manipulant un UDF de poche, on manipule en fait un lien de dérivation, voir la figure 72. Ce lien vient «brancher» les entrées du UDF au contexte d'assemblage pour adapter la forme de la caractéristique poche.

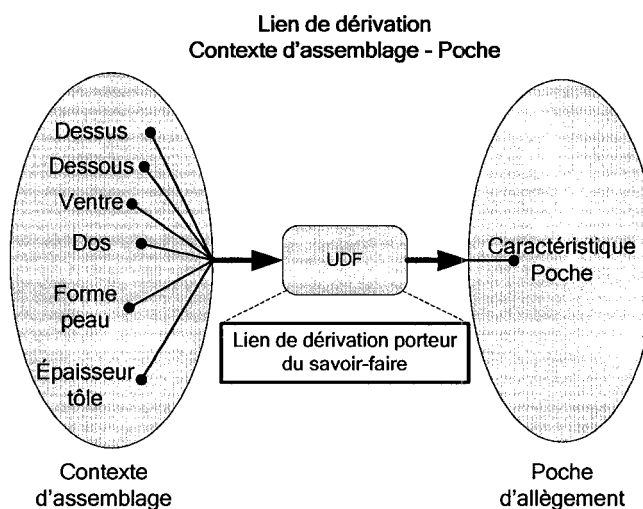


Figure 72 L'UDF comme outil de mise en œuvre du lien de dérivation

On remarquera également qu'en manipulant une relation, on manipule en fait une agrégation de liens. En effet, dans notre cas d'application des poches et de l'APR type, la relation «Contexte d'assemblage – Tôle» agrège 80 liens de dérivation (et 80 UDFs) définissant 80 poches d'allègement reliées au contexte d'assemblage.

Le lien de dérivation se décompose finalement en contraintes où les objets techniques impliqués sont des entités géométriques. La contrainte est le plus bas niveau

d'abstraction des associations, tandis que la formalisation du savoir-faire est au plus haut niveau, voir le tableau IV. Dans notre cas d'UDF poche, ce dernier contient des entités géométriques et des paramètres, mais surtout des contraintes entre ces entités pilotées par les paramètres (ces paramètres correspondent aux règles de conception des poches et ils sont accessibles à l'utilisateur via les publications de Catia V5).

La figure 73 représente le lien de dérivation décomposé en cinq contraintes. La cardinalité de ces contraintes est de 1 X 1 ou de 2 X 1 selon le cas. On peut voir que l'empreinte est mise en œuvre dans l'outil Catia V5 par l'extraction de quatre plans (limites formées par les deux cadres et les deux lisses), d'une surface (l'OML) et d'un paramètre (l'épaisseur nominale de la tôle). L'extraction de l'empreinte est matérialisée par le branchement des entrées de l'UDF poche aux entités géométriques du contexte d'assemblage.

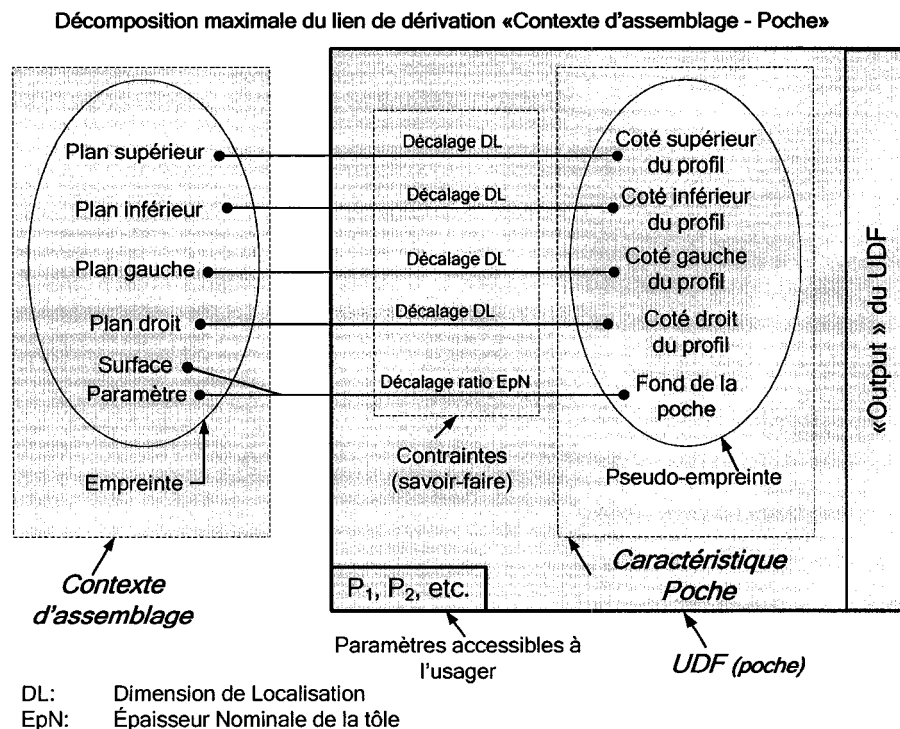


Figure 73 Décomposition du lien «Contexte d'assemblage – Caractéristique Poche» en contraintes

La pseudo-empreinte est mise en œuvre par l'UDF dans lequel on applique le savoir-faire relatif à la modélisation des poches sous forme d'une structure de contraintes. En effet, les quatre plans (représentant les limites des éléments de structures) de l'empreinte sont donc décalés de la dimension de localisation, ce qui permet de déterminer le profil de la poche, tandis que la surface (l'OML) est décalée d'un ratio calculé à partir de l'épaisseur nominale de la tôle, ce qui permet de déterminer la forme et le fond de la poche.

L'UDF est donc une agrégation de contraintes et opère à ce niveau. En effet, lorsque l'on instancie un UDF poche au sein du modèle de la tôle, manuellement ou par programmation, on doit lui fournir les entrées nécessaires. Ces entrées sont des entités géométriques de bas niveau tirées du contexte d'assemblage. Par la suite, bien que n'apparaissant pas sur la figure 73, l'UDF gère le branchement des entrées ainsi que la forme et l'emplacement de la caractéristique poche dans la tôle qui en résulte.

En résumé, l'application automatique de modélisation des poches programmée en VB agit à un haut niveau d'abstraction en décomposant la relation en un ou plusieurs liens. Elle appelle ensuite l'UDF «poche» et lui fournit les entrées nécessaires tirées du contexte d'assemblage. Par la suite, l'UDF, qui agit au niveau du lien, le décompose en contraintes et s'occupe de leur fournir en entrées les entités géométriques adéquates de l'UDF afin qu'une caractéristique poche soit instanciée au bon endroit et avec la bonne forme, voir la figure 74.

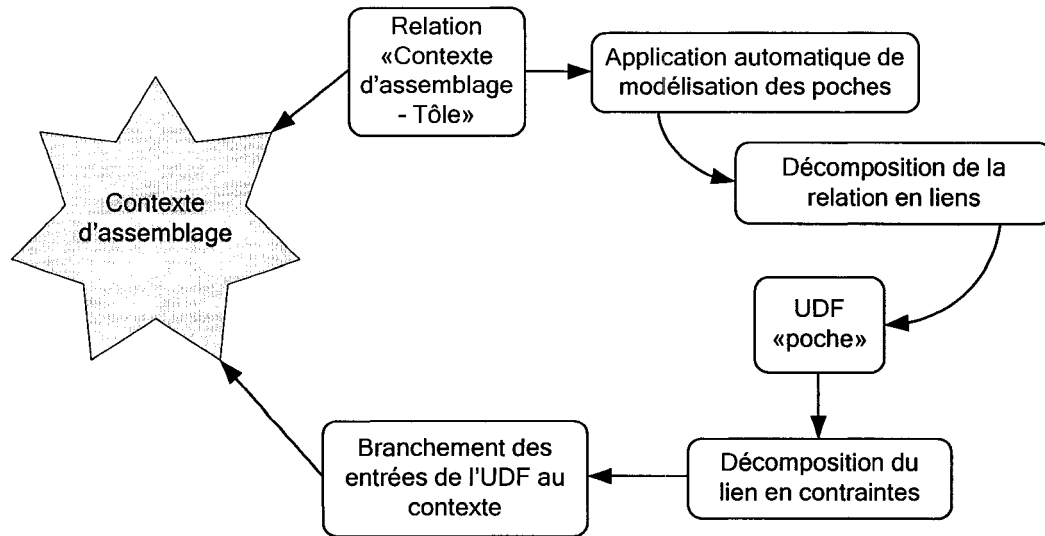


Figure 74 Processus complet d'instanciation de l'UDF poche

5.4 Capacité de déploiement à l'échelle industrielle d'outils CAO permettant d'automatiser des processus de conception

Cette section est dédiée à la discussion des questions fondamentales auquel ce mémoire tente d'apporter un éclairage.

Nous voulons donc alimenter la réflexion sur la valeur de l'automatisation de tâches de modélisation sous forme d'outils orientés métier en industrie. Ce projet nous a permis de faire ressortir certains aspects qui influencent directement la rentabilité de ces projets en entreprise, à savoir :

- Faut-il adapter les méthodologies de conception aux nouveaux outils CAO tel que Catia V5 ?
- Les retours sur l'investissement obtenus en personnalisant des outils de nouvelle génération tel que Catia V5 (UDFs, liens persistants, etc.) ne sont-ils pas plus importants que l'automatisation complète des tâches ?
- Quels sont les gains réels de l'automatisation de tâches de modélisation sous forme d'applications orientées métier ?

- Peut-on assurer la pérennité des applications orientés métier à mesure que les plateformes CAO évoluent ?

5.4.1 Est-il rentable pour une entreprise de modifier ses méthodologies de conception lorsqu'elle migre d'un outil CAO vers un autre?

Nos résultats sur ce point vont dans le même sens que ceux obtenus par Michaud [6]. Non seulement il est rentable de modifier les méthodologies, mais c'est une exigence directe de la migration vers un outil CAO, tel que Catia V5. En effet, si une entreprise veut profiter d'une telle migration, elle doit mettre à jour ses méthodologies et former, bien entendu, les concepteurs au nouvel outil. D'ailleurs, comme le démontre le tableau IX, il faut 43 minutes à Michaud [6] pour modéliser 80 poches sous Catia V4, tandis que le même travail sous Catia V5 lui demande 115 minutes. La maîtrise et la personnalisation de l'outil ainsi que le développement de nouvelles méthodologies de conception adaptées à l'outil nous permettent de modéliser 80 poches en 45 minutes dans un APR. Par rapport à l'utilisation actuelle de Catia V4, il n'y a pas vraiment de gains. Toutefois, en comparant nos résultats à ceux de Michaud [6] sous Catia V5, nous obtenons un gain de 61% en adaptant les méthodologies. Pour un avion complet de type Régional, cette économie représente 3431 minutes (2500 (poches dans l'avion) * 5 min. / poche * 45% (première famille de tôle) * 61% (économie de temps)), soit environ 57 heures. Il est important de noter que cette économie de temps concerne seulement les tôles de la première famille.

Tout comme Michaud [6], nous avons également constaté que les gains de temps associés au passage de Catia V4 à Catia V5 se font surtout au niveau des modifications apportées à l'APR une fois qu'il est modélisé.

5.4.2 Est-il rentable pour une entreprise d'investir dans la personnalisation d'outils CAO permettant d'encapsuler et de réutiliser les connaissances ?

Les résultats que nous avons obtenus lors de la construction de l'APR type et de l'instanciation des poches d'allègement, tant au niveau de la personnalisation à l'aide d'UDF, qu'en mode automatique par un programme VB, démontrent clairement le potentiel de gains réalisables lors de l'utilisation d'outils tel que les UDFs de Catia V5. En effet, ils permettent d'encapsuler dans l'outil les connaissances propres à une tâche et de capitaliser sur le travail déjà fait en réutilisant de la géométrie adaptable au contexte. Plusieurs arguments nous mènent à conclure qu'il est très rentable pour une entreprise d'investir dans ce type d'outil.

Tout d'abord, l'emploi de liens persistants, établis lors de l'instanciation, entre les entrées des UDFs et le contexte de conception, constitue la majeure partie des gains de temps obtenus. En effet, ces liens permettent aux UDFs de se connecter au contexte, d'adapter leur forme et lorsqu'il y a un(des) changement(s), de les mettre à jour automatiquement. L'un des avantages principal relié à la capacité de propager des changements très rapidement au sein d'un assemblage concerne la possibilité d'explorer plusieurs solutions lors du processus de conception. Lederman [48] exploite d'ailleurs cette possibilité pour effectuer des calculs d'optimisation de poids dans une maquette numérique commune à plusieurs domaine d'expertise afin de déterminer le nombre de pièces structurelles composant un APR d'avion.

Un autre aspect très intéressant des liens persistants manipulés sous forme d'UDFs est qu'ils permettent d'intégrer le savoir-faire d'une organisation, sous forme de règles de conception à même les modèles tridimensionnels. Actuellement, lorsqu'un concepteur quitte l'entreprise, il amène avec lui toutes les connaissances qu'il a acquises durant ses années de travail au sein de l'organisation. L'utilisation d'UDFs permet de récupérer une partie de ces connaissances, ce qui rentabilise l'usage de ce type d'outil à un autre niveau.

En troisième lieu, l'utilisation des UDFs force les concepteurs à respecter les méthodologies de conception et permet d'homogénéiser les pièces. En effet, en utilisant une instance d'UDF plutôt que de construire un nouveau modèle, le concepteur s'assure que la pièce respecte la méthodologie de l'entreprise. D'ailleurs, en industrie, une minorité de concepteurs suivent les méthodologies de conception, ce qui entraîne parfois des délais et des coûts supplémentaires. De plus, en homogénéisant le plus de pièces et de caractéristiques possibles, l'entreprise diminuera encore le temps liés aux changements d'ingénierie en réduisant le temps nécessaire aux concepteurs à la compréhension d'un modèle qu'ils n'ont pas créé.

Quatrièmement, un autre gain apporté par l'utilisation des UDFs est la diminution de l'encombrement de l'arbre des spécifications. En effet, la géométrie composant un UDF n'est pas visible dans l'arbre, ce qui rend la navigation plus facile surtout lors de modifications effectuées par un concepteur ne connaissant pas le modèle. Cette simplicité devrait se traduire par une économie de temps supplémentaire.

En cinquième lieu, l'utilisation des UDFs peut éviter l'utilisation des contraintes d'assemblage, comme dans notre cas des poches, ce qui diminue le temps passé à construire un assemblage et contribue à rentabiliser l'outil.

Finalement, Nous pensons que chaque pièce et/ou caractéristique dupliquée, selon un seuil à déterminer, devrait faire l'objet d'une encapsulation à l'aide d'un outil tel que les UDFs. En effet, chaque instanciation d'UDF fait économiser un temps considérable, comparativement à la « re-modélisation » complète ou partielle.

Il n'y a pas que des avantages à utiliser les UDFs et bien que Catia V5 évolue rapidement, nous avons noté certains inconvénients et/ou limitations lors de l'utilisation de cet outil.

Tout d'abord, sous la version R14 de Catia V5, lorsqu'un UDF était créé dans le modèle servant à sa définition, il n'était plus possible de le modifier. Sa définition était un processus à sens unique et aucun lien n'était établi entre l'UDF et les éléments géométriques du modèle ayant servi à sa définition. Cette limitation était très irritante, car il est très rare de réussir la création d'un UDF lors du premier essai, surtout pendant la période d'apprentissage de l'outil. Si l'UDF ne fonctionnait pas correctement ou selon les critères voulus, il fallait ouvrir à nouveau le modèle servant à sa définition, effacer l'UDF et en créer un autre. Cette limitation est toutefois corrigée dans la version R16 de Catia V5 où il est maintenant possible de modifier les entrées, les sorties, les paramètres publiés par l'UDF et même son type.

Deuxièmement, il n'existe aucun lien entre le prototype d'un UDF et ses instances, de telle sorte que les modifications apportées au prototype ne sont pas propagées aux instances déjà présentes dans les différents modèles. Lorsque le prototype d'un UDF est modifié, l'état actuel de l'outil (Catia V5 R17), oblige l'utilisateur à procéder à une nouvelle instanciation dans les modèles contenant déjà des instances. En effet, il n'existe pas de liens persistants entre le prototype d'un UDF et ses instances.

Il est à noter qu'une application de modélisation automatique, tel que notre prototype de définition automatique des poches d'allègement, pallie à cette limitation en permettant d'instancier rapidement de nouveaux UDFs au sein d'un modèle. D'ailleurs, le développement d'une application d'instanciation automatique d'UDFs, capable de gérer plusieurs prototypes, seraient, à notre avis, très utile dans le contexte d'utilisation de cet outil.

Finalement, le dernier inconvénient que nous avons noté lors de l'utilisation des UDFs, concerne le nombre de fonctions supportées par l'outil. En effet, sous les versions R14 à R16, il n'était pas possible d'utiliser des fonctions telles que le «Loft», le «Shell» ou les

«Edges fillets» dans le prototype d'un UDF. Toutefois, comme nous l'avons démontré lors du développement de notre application, il est possible de pallier à cette limitation en étant créatif et en utilisant les outils de bas niveaux. L'évolution de Catia V5 laisse toutefois croire que ces fonctions seront intégrées progressivement dans les nouvelles versions de Catia V5.

Nous pensons tout de même que les avantages à utiliser les UDFs sont plus nombreux et plus importants que les inconvénients. Le potentiel d'utilisation de l'outil UDF est, à notre avis, considérable et il ne fait aucun doute dans notre esprit quand à sa rentabilité en entreprise.

5.4.2.1 Solutions apportées aux contextes de Giguère et Michaud à l'aide d'UDFs

Pour démontrer que l'apprentissage et l'utilisation d'outils tel que les UDFs sont rentables et performants, nous avons repris la problématique de définition automatique de soyages et de découpes de François Giguère [3-5] ainsi que celle de la création d'un gabarit de découpe pour les poches d'allègement de Michel Michaud [6].

Giguère a passé environ 6 mois à programmer son application sous Catia V5 R6 à cause du degré de maturité de l'outil disponible en 2001. Rappelons que son application automatisait l'insertion de soyages et de découpe à l'intersection d'un cadre et d'une lisse, qu'elle propageait automatiquement ou manuellement les changements aux pièces impactées dans l'assemblage et qu'elle intégrait dans l'outil (sous forme de code) les connaissances liées au contextes des soyages et des découpes (voir section 3.5 pour plus de détails).

La formalisation des concepts d'empreinte et de pseudo-empreinte que Giguère a faite nous a été très utile dans notre travail et suite à l'expérience acquise lors du développement de notre prototype d'application automatisant les poches d'allègement, nous avons repris les contextes de Giguère et de Michaud pour les solutionner à l'aide

des outils actuels et de notre approche utilisant, entre autre, les UDFs. Le but de cet exercice est de démontrer qu'il est profitable d'utiliser ces outils dans d'autres contextes que celui des poches d'allègement.

Nous avons donc débuté par le contexte de Giguère. L'objectif étant de créer un UDF contenant les soyages et les découpes d'un cadre à l'intersection d'une lisse. Comme le démontre le tableau XI à la page suivante, la construction du modèle utilisé pour l'UDF prend 55 minutes, la création de l'UDF en soit prend 10 minutes et l'instanciation manuelle du UDF requiert seulement 2 minutes. Cet UDF de soyages et de découpes s'adapte automatiquement au contexte à l'aide de liens persistants assurant ainsi la propagation de changements apportés à la lisse ou au cadre. De plus, l'UDF intègre dans l'outil CAO la connaissance spécifique reliée à la tâche de modélisations de soyages et de découpes.

Afin de pouvoir comparer nos résultats à ceux de Giguère, nous avons transposé les six mois de travail qu'il a investis dans le développement de son application en minutes. Pour ce faire, nous avons supposé une semaine de travail de 40 heures, ce qui nous donne 57600 minutes (6 mois * 4 semaines/mois * 40 heures/semaines * 60 minutes/heures). Ce qui nous permet de calculer un gain de 100% de temps par rapport à la personnalisation des outils à l'aide d'UDFs.

Il est à noter que notre programme d'insertion et de mise à jour automatique des poches d'allègement pourrait être facilement adapté afin d'insérer automatiquement tous les soyages et toutes les découpes contenues dans un APR. En termes de chiffres, il s'agit de 99 découpes et 198 soyages pour l'APR ayant servie pour le développement de notre prototype d'application. Giguère dénombra 4302 soyages et découpes dans un avion de type régional et 2962 dans un avion de type affaire [3-5].

Dans le cas du contexte de gabarit de découpe de poches d'allègement dans les APRs d'un avion utilisé par Michaud, il lui faut 44 minutes pour créer le modèle du gabarit sous Catia V4 (ce temps est basé sur un APR contenant 80 poches) et 36 minutes sous Catia V5 pour créer le même modèle [6]. Nous avons repris ce contexte et créé un UDF de gabarit de découpe de poches d'allègement pour une tôle de revêtement. C'est un UDF très simple qui consiste tout d'abord à créer une surface entre l'OML et l'IML, pour intégrer une provision pour l'attaque du métal par l'acide, puis d'épaissir cette surface afin de créer le modèle solide dans un deuxième temps. Cette technique permet de réutiliser l'effort de modélisation effectué lors des poches d'allègement de l'APR. Comme le démontre le tableau XI, la construction du modèle prototype de l'UDF prend 15 minutes, la création de l'UDF en soit prend 5 minutes et l'instanciation manuelle nécessite moins d'une minute. Le gain de temps relié à l'instanciation d'un UDF de gabarit de découpe par rapport à la modélisation complète est de plus de 97%

Tableau XI

Temps requis pour créer et instancier des UDFs
des contextes de Giguère et Michaud

	Étapes de construction d'un UDF			Temps total de création (min.)	Gains %
	Construction du modèle de définition (min.)	Création du prototype (min.)	Instanciation manuelle (min.)		
Giguère (soyages et découpes)	-	-	-	57600	100%
Lamarche (soyages et découpes)	55	10	2	65	
Michaud (gabarit de découpes)	-	-	-	36	97,2%
Lamarche (gabarit de découpes)	15	5	1	20	

En comparant le temps nécessaire à la création de l'application de modélisation automatique des soyages et des découpes développée à l'époque par Giguère [3-5] avec celui que nous avons obtenu en personnalisant cette même tâche à l'aide d'UDF, on peut conclure que l'utilisation des UDFs dans ce contexte est très rentable. On peut voir également que sous Catia V5, l'automatisation en 2001 coûtait plus cher que la personnalisation en 2006.

De même, en comparant le temps nécessaire à Michaud [6] pour créer un modèle de gabarit de découpe avec celui que nous avons obtenu en personnalisant cette tâche à l'aide d'UDF, nous obtenons un gain de temps de l'ordre de 97% qui représente en fait 35 minutes. Le gain est plutôt marginal par rapport à celui obtenu pour l'UDF de soyages et de découpes.

Globalement on peut donc conclure qu'une fois maîtrisé, l'utilisation des UDFs est très rentable pour une entreprise.

5.4.3 Est-il rentable pour une entreprise d'investir dans la programmation d'application orientée métier afin d'automatiser certaines tâches de modélisation ?

L'utilisation d'application automatique semble être très rentable pour une entreprise car elle améliore l'économie de temps obtenue par l'utilisation manuelle des données structurées en permettant de capitaliser sur le travail déjà fait. Le tableau XII, ci-dessous, reprends les résultats obtenus par François Giguère. La première colonne, intitulée «Tâch», désigne la tâche effectuée, la deuxième et la troisième colonne, intitulées «Manuel» et «Assistée», indique de qu'elle façon la tâche est accomplie et finalement, la dernière colonne, intitulée «Gain», représente les gains de temps obtenus en utilisant l'application développée par Giguère par rapport à la méthode manuelle.

Comme on peut le constater, le gain de temps obtenu par l'utilisation de l'application est en moyenne de 91%, tandis que le gain de temps lié à la propagation d'un changement

apporté à la géométrie est de 72% et celui lié à un changement apporté à la position est de 85%.

Tableau XII

Gains apportés par l'application de soyages et de découpe de Giguère [3-5]

Tâche	Manuel	Assisté	Gain
Modélisation d'un soyage et d'une découpe	6.5 min.	0.4 min.	94 %
Modélisation d'une découpe	3.2 min.	0.4 min	88 %
Propagation d'une modification géométrique	4.3 min.	1.2 min.	72 %
Propagation d'une modification de position	4.0 min.	0.6 min.	85 %
Modélisation de toutes les caractéristiques contextuelles d'un avion régional	403 heures	29 heures	93 %
Modélisation de toutes les caractéristiques contextuelles d'un avion d'affaires	265 heures	19 heures	93 %

Ces pourcentages sont impressionnants, mais lorsqu'on les transpose en temps et qu'on les compare à ceux investis pour développer l'application, la rentabilité de l'exercice n'est pas évidente. En effet, comme le démontre le tableau X, Giguère prit 960 heures (57600 min.) pour développer son application automatique de soyages et de découpe, tandis qu'elle permet de sauver 375 heures de travail pour un avion de type régional (403 * 93%) et 246 heures (265 * 93%) pour un avion de type affaire. Basé sur cette

seule application, même en additionnant les temps épargnés pour les deux types d'avion, il est plus rentable pour une entreprise de demeurer en mode manuel.

Dans le même ordre d'idée, le développement de l'application de modélisation automatique des poches d'allègement nous pris également environ 6 mois. Basé sur le même calcul de temps que pour le cas de Giguère, nous obtenons aussi un investissement de temps de l'ordre de 960 heures. En comparant ce chiffre avec les économies obtenues par l'application pour un avion complet, soit 57 heures (voir la section 5.4.1), il est encore une fois plus rentable pour une entreprise de demeurer en mode manuel.

Pris individuellement, ces deux exemples nous poussent à conclure que l'automatisation de tâches de modélisation n'est pas rentable et demeure un outil de laboratoire.

Il est toutefois à noter, que dans le cas des poches d'allègement, seulement 45% des 2500 poches contenues dans un avion de type régional sont modélisables automatiquement et que l'application serait facilement adaptable au 55% restant. De plus, il serait possible de capitaliser sur le temps de développement investi en adaptant l'application pour qu'elle puisse être réutilisée dans d'autres contextes.

Un fait important à souligner est que les programmes VB libèrent complètement le concepteur et que pendant ce temps, il peut effectuer d'autres tâches de modélisation. Cette économie supplémentaire de temps n'est pas comptabilisée dans nos données, mais doit, à notre avis, être considérée.

5.4.4 Existe-t-il d'autres activités de modélisation du domaine aéronautique qui pourraient potentiellement bénéficier de la personnalisation des outils CAO et des applications automatiques ?

Le domaine aéronautique offre une multitude de contextes dans lesquels l'encapsulation de géométrie et de caractéristiques à l'aide d'UDFs, jumelée à l'instanciation

automatique, peuvent être appliquée. Nous avons déjà testé les cas de soyages et de découpes (aux intersections cadre-lisse), celui des poches d'allègements dans les APRs, celui de la création de gabarits de découpes pour les poches d'allègement des tôles devant être usinées chimiquement et notre collègue du LIPPS, Omar Msaaf [57;58], a adapté la méthode à la création automatique de pièces constituant une aile d'avion (des longerons et des lisses) afin de supporter l'optimisation multidisciplinaire.

Nous distinguons donc au moins deux types d'activités de modélisation potentielles dans lesquelles la personnalisation des outils CAO et les applications automatiques permettraient d'obtenir des gains de temps considérables. La première catégorie regroupe les activités de modélisation de caractéristiques, telles que les poches, les soyages et les découpes. La deuxième catégorie, regroupe plutôt des pièces complètes telles que des cadres et des lisses.

En considérant l'anatomie d'un cadre, voir la figure 11, nous notons deux autres caractéristiques susceptibles d'être encapsulées à l'aide d'UDFs et instanciées automatiquement à l'aide d'un programme VB tel que le nôtre. En effet, les trous d'allègements, «lightening holes», servant à diminuer le poids des cadres tout en maintenant leur rigidité sont positionnés entre deux découpes. À l'aide des données collectées par Giguère [3-5] chez Bombardier Aéronautique, nous pouvons établir le nombre de trous d'allègement, pour les cadres seulement, à environ 2000 pour un avion de transport régional. De même, les embossages, «beads», servant à rigidifier le cadre près d'une découpe, se dénombrent également à environ 2000 dans le même type d'aéronef.

Ainsi, sur une seule pièce structurelle, le cadre présente déjà deux autres caractéristiques auxquelles la méthode présentée dans ce travail pourrait facilement s'appliquer. Le nombre de chacun des deux types de ces caractéristiques est sensiblement le même que pour les poches d'allègement et nous pouvons entrevoir un gain de temps de l'ordre de

90% en automatisant l'insertion et la mise à jour de ces caractéristiques selon les mêmes méthodes que pour les poches d'allègement. En effet, les règles régissant la définition des trous d'allègements et des embossages sont inscrites dans des méthodologies de conception bien définies de la même façon que pour les poches d'allègement et les soyages et les découpes. Il est donc envisageable d'encapsuler le savoir-faire propre à la définition de ces caractéristiques dans un UDF afin de les instancier dans le contexte approprié au même titre que les autres caractéristiques mentionnées précédemment. De plus, d'autres types de caractéristiques se retrouvent un peu partout dans les pièces de structure de l'avion tel que dans les longerons, dans les poutres, dans les nervures, etc.

Nous pensons également qu'il serait possible de créer une application de création, d'insertion et de mise à jour automatique pour la construction des poches contenues dans les pièces usinées qui se comptent par milliers dans un avion. En effet, la modélisation de ce type de poches est régie par des règles bien définies et ces poches se rattachent au contexte, entre autre, à l'aide de plans et de surfaces. On peut supposer que les gains de temps sur ces tâches seraient du même ordre que pour les poches d'allègement, les soyages et les découpes.

Dans la catégorie des pièces, nous pensons qu'il serait relativement facile d'appliquer notre méthode de réutilisation de modèles à l'aide d'UDFs et d'insertion automatique pour les nervures se retrouvant dans les ailes et dans certaines parties des fuselages. Ces méthodes pourraient s'appliquer également à tous les différents types de pièces d'attache, « brackets » et « clips », tout comme nous l'avons fait pour les lisses, les cadres, la tôle, etc., qui se comptent par milliers dans un avion. Cependant, il est à noter que les fonctions « part-template » et « product-template » du module BKT (Business Knowledge Template) de Catia V5 sont également à considérer pour la réutilisation de modèles de pièces complètes plutôt que l'outil UDF. En effet, les UDFs sont destinés à être utilisés au niveau des caractéristiques se trouvant à l'intérieur des pièces. Toutefois,

l'un des avantages intéressant des UDFs est la réduction de l'encombrement de l'arbre des spécifications, ce qui n'est pas le cas pour les fonctions mentionnées ci-haut.

Le domaine de l'outillage nécessaire à la fabrication de pièces est également très intéressant à considérer pour implanter la personnalisation des outils CAO et les applications automatiques de modélisation. Comme l'explique Michaud [6], le concepteur d'outillage doit définir, à partir du modèle de la pièce à fabriquer, des dispositifs permettant de positionner la pièce, des systèmes de localisation de l'outillage, etc. De plus, le secteur de l'aéronautique entraîne des difficultés additionnelles à cause des formes complexes et des tolérances exigées. Ce qui amène le concepteur d'outils à modéliser des pièces devant être fabriquées pour assurer la fabrication des pièces composant un aéronef. Comme par exemple, le gabarit de découpe nécessaire pour l'usinage chimique des poches d'allègement logées dans la tôle des APRs d'avion. Il existe un lien direct entre certaines pièces devant être fabriquée et l'outillage nécessaire à sa fabrication. Des outils comme les liens persistants, les UDFs et les programmes VB ou C++ peuvent procurer des gains de temps considérables dans la modélisation de l'outillage et la propagation des modifications lorsque l'outillage est impacté.

Globalement, on évalue que le domaine aéronautique est propice au développement de telles applications à cause du grand nombre de pièces semblables devant s'adapter au contexte. C'est pourquoi nous pensons que l'approche de personnalisation des tâches CAO à l'aide d'outils tels que les liens persistants, les UDFs et l'instanciation automatique est généralisable.

5.4.5 Peut-on assurer la pérennité des outils CAO de personnalisation et d'automatisation à mesure que la plateforme évolue?

Durant le développement de notre prototype d'application utilisant Catia V5, nous avons noté que Dassault Système développe et met à jour continuellement son logiciel et ce de deux façons : une « Release » est mise sur le marché environ une fois par année tandis

que des « Services Pack » (S.P.) sont délivrés environ tous les deux mois. Les versions « Release » sont des changements majeurs apportés au logiciel, comme par exemple, la modification de l'interface usager, l'amélioration des pilotes vidéo, la modification ou l'ajout de nouveaux ateliers, la modification ou l'ajout de nouvelles fonctions, etc. Les S.P. quant à eux, sont des correctifs corrigeant des blocages ou des dysfonctionnement du logiciel qui se manifestent de plusieurs façons comme, par exemple, par l'abandon prématuré d'une fonction et même par la fermeture du logiciel.

Lors de notre séjour chez Bombardier Aéronautique, nous avons appris que la stabilité des applications fournies aux concepteurs constituait l'une de leur préoccupation majeure. En effet, les mises à jour faites par Dassault Système sur cette version de Catia occasionnaient un travail de déploiement de 6 à 8 mois, réalisé au niveau des méthodologies d'utilisation de l'application et sur la vérification du bon fonctionnement du logiciel. En conséquence, le développement d'applications automatiques dans ce contexte n'est pas rentable à cause des coûts de maintien très élevés.

Notons également que l'application développée par notre collègue François Giguère fonctionnait uniquement sous Catia V5 R6 SP-2.

Bien que Catia V5 ait atteint aujourd'hui un niveau de maturité très acceptable (version R16), nous avons profité du contexte de développement de notre application automatique sous Catia V5 pour vérifier la pérennité des UDFs, des liens persistants de notre APR type, ainsi que de notre programme VB lors du passage d'une « Release » à l'autre, dans cinq cas distincts, soit de R12 à R16 ainsi qu'en appliquant deux correctifs S.P. pour chaque « Release ».

La construction de notre APR type, la modélisation de nos pièces et de notre caractéristique poche sous forme d'UDFs, ainsi que la programmation de notre application se sont faits initialement sous Catia V5 R14 sans S.P. appliqués. Il ne nous était pas possible de rétrograder l'ensemble de notre développement, en particulier le

programme VB. Toutefois, nous avons refait la modélisation de l'APR et des UDFs sous Catia V5 R12 afin de pouvoir tester la migration des modèles vers R16.

Les tests de migration pour les modèles consistaient tout d'abord à charger le modèle de l'APR en mémoire et effectuer un changement majeur dans l'assemblage tel que la modification de l'OML simple courbure en OML double courbure. Cette manipulation permettait de vérifier le fonctionnement des liens persistants de l'assemblage ainsi que la pérennité des modèles. Par la suite, nous avons effectué l'instanciation d'UDFs au sein de l'assemblage pour chacune des pièces, soit une lisse, un cadre, une tôle et une caractéristique poche. Finalement, pour les versions R15 et R16 nous avons lancé notre programme d'insertion automatique de poches sur l'APR.

Le tableau XIII présente les résultats de nos tests de migration sur nos modèles, ainsi que sur notre programme VB, pour les différentes versions « R » de Catia V5 et les correctifs appliqués.

Tableau XIII

Pérennité de l'application à travers l'évolution de Catia V5

Types de tests	R12			R13			R14		
	SP0	SP4	SP6	SP0	SP3	SP6	SP0	SP3	SP7
<i>Liens persistants de l'assemblage</i>	√	√	√	√	√	√	√	√	√
<i>Instanciation d'UDFs</i>	√	√	√	√	√	√	√	√	√
<i>Exécution du programme VB</i>	NT	NT	NT	NT	NT	NT	√	√	√

Tableau XIII (suite)

Types de tests	R15			R16		
	SP0	SP2	SP3	SP0	SP2	SP3
<i>Liens persistants de l'assemblage</i>	√	√	√	√	√	√
<i>Instanciation d'UDFs</i>	√	√	√	√	√	√
<i>Exécution du programme VB</i>	√	√	√	√	√	√

Le seul problème que nous avons relevé lors de ces migrations est la perte occasionnelle des liens entre les pièces de l'APR. Il est toutefois très facile à régler et demande peu de temps, car il suffit de réinitialiser un seul lien et la mise à jour complète du modèle réinitialise les autres liens. Ce problème nous semble lié au noyau de Catia V5 compte tenu du fait que la réinitialisation d'un seul lien permet de réinitialiser tous les autres, ce qui est surprenant. Nous pensons donc que le problème sera éventuellement corrigé dans les prochaines « Releases » de Catia V5.

Les tests de pérennité que nous avons effectués nous permettent de comprendre que le travail de maintien des modèles et de codes relatif à des programmes d'application automatique sous Catia V5 est relativement léger, bien qu'absolument nécessaire. En effet, l'utilisation des liens persistants est à la base de notre méthode et donc de l'économie de temps qui en résulte. Il est donc crucial de maintenir toutes les associations établies à l'intérieure et entre les différents modèles.

5.5 Synthèse de l'analyse des résultats

L'exploitation des liens persistants de Catia V5 dynamise les modèles et leur confère une réactivité leur permettant de se mettre à jour lorsqu'un changement survient au sein du modèle ou de l'assemblage. La mise en place de l'approche de conception par caractéristiques à l'aide d'outils d'encapsulation de la géométrie, tel que les UDFs de Catia V5, procure un gain appréciable par rapport à l'investissement.

La programmation d'applications automatiques de tâches de modélisation permet des gains de temps appréciables lorsqu'elle est comparée avec des méthodes de modélisation manuelles. Toutefois, lorsque les applications automatiques sont comparées aux outils d'encapsulation, tel que les UDFs, les gains obtenus ne valent pas l'investissement. Ce qui n'exclut pas le recours aux applications automatiques, mais l'emploi des outils d'encapsulation devrait être priorisé.

Des nouvelles méthodes de travail nécessitent souvent une restructuration des tâches. La mise en place et l'utilisation des méthodes discutées ici ne font pas exception et, compte tenu de l'importance des liens persistants dans cette approche, nous pensons qu'il serait très judicieux de mettre en place une structure ou un mécanisme qui permettra d'assurer la continuité de tous les maillons constituant la chaîne de liens persistants entre les modèles.

CONCLUSION

Ce travail visait deux objectifs principaux. Le premier consistait à franchir un pas de plus dans la généralisation des concepts de typage d'entités, permettant d'attribuer un comportement spécifique à une entité ou à un groupe d'entités géométriques dans un contexte donné, et celui d'associations établies entre ces entités et leur empreintes et pseudo-empreintes au sein d'une caractéristique contextuelle. Cette dernière, permet entre autre, d'intégrer une partie du savoir-faire d'ingénierie propre à une tâche de modélisation au sein même de l'outil CAO dans le but de le réutiliser ultérieurement.

Nous avons atteint ce premier objectif en appliquant ces concepts, dans un premier temps, au contexte industriel de modélisation des poches d'allègement contenues dans les panneaux de revêtement des fuselages d'un avion dont la tôle doit être usinée chimiquement, et, dans un deuxième temps, en les mettant en œuvre au sein de l'outil CAO de nouvelle génération Catia V5.

La CAO manipule par défaut des contraintes et des entités de bas niveau. Les UDFs (User-Defined Features) forment un mécanisme d'agrégation de contraintes qui constitue, en soi, l'encapsulation d'un savoir-faire spécifique à une tâche. L'instanciation, même manuelle, d'un UDF force quand même l'utilisateur à sélectionner les objets de bas niveau qui alimentent les entrées de ce dernier. En effet, à ce niveau, il n'y a pas d'agrégation des objets techniques. La mise en œuvre des concepts en personnalisant la modélisation des poches à l'aide des UDFs et des liens persistants de Catia V5 nous ont permis d'obtenir des gains de temps de l'ordre de 86% par rapport à la méthode manuelle actuellement utilisée en industrie.

Nous avons également développée une application orientée métier qui libère le concepteur de tâches de modélisation longues et pénibles et lui permet de travailler sur d'autres tâches requérant toute son attention. Le recours à un programme

(automatisation) mène, d'une certaine manière, à traiter les objets techniques de manière agrégée. En effet, c'est l'application automatique qui décompose les entrées présentées sous formes de pièces afin d'en extraire les objets techniques de bas niveau pour les transmettre ensuite aux entrées correspondantes de l'instance d'UDF.

Les gains de temps obtenus à l'aide de cette application, par rapport à la méthode manuelle sous Catia V4, sont de l'ordre de 98,5% et par rapport à la méthode manuelle sous Catia V5, les gains sont de l'ordre de 90%.

Le deuxième objectif consistait à analyser et à statuer sur le potentiel de déploiement industriel de la personnalisation des outils CAO et de l'automatisation de certaines tâches de modélisation. Cette analyse nous conduit à certaines conclusions et observations. Tout d'abord, comme le conclut Michaud [6], lors de la migration d'un outil CAO vers un autre, l'adaptation des méthodologies de modélisation est incontournable afin d'exploiter adéquatement le nouvel outil. En effet, la simple transposition des méthodologies sur la nouvelle plateforme s'est avérée inefficace, voire même plus coûteuse en temps, car elles n'exploitaient pas les nouveaux outils disponibles sous la nouvelle plateforme.

En deuxième lieu, l'utilisation d'un outil d'encapsulation de géométries et de connaissances, tel que les UDFs, permet de diminuer substantiellement le temps de modélisation répétitives. De plus, nous avons ciblé plusieurs autres cas du domaine aéronautique où l'utilisation de ces outils serait potentiellement rentable.

Troisièmement, les tests de pérennités que nous avons effectués sont concluants et nous permettent de présumer que le travail de maintien des modèles et de codes relatif à des applications automatiques sous Catia V5 est relativement léger.

Finalement, la cardinalité des différentes associations impliquées entre les objets techniques n'est pas un problème sous Catia V5.

Globalement, le recours aux UDFs comme outils de personnalisation semble faisable et souhaitable. Par contre, la pertinence du recours à l'automatisation complète de tâches de modélisation, à l'aide de programmes VB, bien que fournissant un gain supplémentaire de l'ordre de 90% par rapport à l'utilisation des UDFs dans notre cas, nécessite un investissement considérable de temps au niveau de l'apprentissage de l'outil et de la programmation des applications par rapport aux gains réels qu'elle apporte. Avant de se lancer dans un projet de développement de programmes dédiés, une analyse complète et minutieuse des besoins en automatisation, des compétences requises pour la programmation des applications et de la disponibilité des outils est nécessaire afin que les gains en économie de temps surpassent les coûts engendrés par le projet.

RECOMMANDATIONS

Ce travail de maîtrise confirme que les outils de personnalisation supportent adéquatement les concepts permettant l'exploitation des outils CAO modernes. Il met également un bémol sur le déploiement systématique des outils visant à automatiser complètement des tâches de modélisation. Malgré le travail réalisé, il demeure bien des pistes à explorer et, au fur à mesure que les outils évoluent, d'autres pistes se dévoilent.

Voici les principales recommandations que nous pouvons formuler suite à nos travaux :

- a. Dans ce travail, le sens de l'information circulant à l'aide d'associations est unidirectionnel, i.e. de la référence vers la cible. Il serait intéressant d'explorer les concepts et les outils sur un cas d'application nécessitant que le flux d'information via les associations soit bidirectionnel.
- b. L'intégration des concepts d'empreinte et de pseudo-empreinte s'est faite naturellement dans l'outil haut de gamme Catia V5 à l'aide d'UDFs et des liens persistants. Il serait intéressant d'explorer les plateformes CAO de milieu de gamme, utilisées majoritairement par les PME, tels que SolidEdge, SolidWorks, TopSolid, etc, afin d'évaluer le niveau d'intégration des concepts et les outils natifs disponibles dans ces plateformes.
- c. Les concepts ont été validés à l'aide de cas d'application tirés du domaine aéronautique. La poursuite de la généralisation pourrait s'étendre à d'autres cas tirés de domaines divers tels que la fabrication d'automobiles, de trains, de bateaux, mais aussi de la fabrication de produits de consommation.

- d. Dans ce travail, les associations et la propagation des changements sont circonscris à l'intérieur même de la plateforme CAO. Il serait pertinent d'étendre leur application à d'autres types de documents impliqués dans le processus de fabrication de produits, tels que les gammes de fabrication, les listes de matériel et même les documents liés à la comptabilité et à la planification des quarts de travail, par exemple.

BIBLIOGRAPHIE

- [1] Maranzana, R. (1999). *Les liens technologiques: De la modélisation géométrique à la modélisation fonctionnelle*. Rapport de recherche: Bombardier Aéronautique - École de Technologie Supérieure.
- [2] Holding, J. (2002). *Séminaire sur le Partage des données électroniques sur les produits: Solutions client-fournisseur*, Institut des matériaux industriels du CNRC, Ministère de l'Industrie du Québec.
- [3] Giguère, F. (2002). *Application des liens multi-modèles à la conception mécanique*. Mémoire de maîtrise, Université de Sherbrooke et École de Technologie Supérieure, Sherbrooke.
- [4] Giguère, F., Rivest, L., & Desrochers, A. (2001). Improving Design Productivity and Product Data Consistency ;From Design Intent to Solid Models through Technological Links. *Proceedings of the International IFIP Conference FEATS 2001*.
- [5] Giguère, F., Rivest, L., Desrochers, A., & Maranzana, R. (2001). Les caractéristiques contextuelles : une solution pour accroître la productivité en CAO. *Revue Internationale de CFAO et d'Informatique Graphique.*, 17/1-2.
- [6] Michaud, M. (2004). *Méthodologie de modélisation unifiée pièce-outillage en CAO aéronautique : application aux tôles et gabarits de découpe*. Mémoire de maîtrise, École de technologie supérieure, Montréal.
- [7] Jenkinson, L. R., Simpkin, P., & Rhodes, D. (1999). *Civil jet aircraft design* (ed.). Reston, Va.: American Institute of Aeronautics and Astronautics.
- [8] Lamarche, B., & Michaud, M. (2003). *Stage en entreprise - L'usage de la CAO reliée à l'usinage chimique chez Bombardier aéronautique*. Montréal: École de Technologie Supérieure.
- [9] Soors, P. (1984). *Dessin d'outillage en aéronautique* (ed.). Outremont: Modulo.
- [10] Cutler, J. (1992). *Understanding aircraft structures* (2nd ed ed.). Oxford, England: Blackwell Scientific.
- [11] Harris, W. T. (1976). *Chemical milling: the technology of cutting materials by etching*. (ed.). Oxford: Clarendon Press.

- [12] Dassault. (2005). http://www.dassault-aviation.com/passion/fr/dassault_a_a_z/histoire_aviation/sites_aviation/argenteuil.cfm. Retrieved 16 février, 2005
- [13] Ullman, D. G. (2003). *The mechanical design process* (3rd ed.). New York, N.Y.: McGraw-Hill.
- [14] Hyman, B. I. (1998). *Fundamentals of engineering design* (ed.). Upper Saddle River, N.J.: Prentice-Hall.
- [15] Suh, N. P. (1990). *The principles of design* (ed.). New York: Oxford University Press.
- [16] Finger, S., & Dixon, J. R. (1989). A Review of research in mechanical engineering design. Part I: Descriptive, prescriptive and computer-based models of design process. *Research in Engineering Design*, 51-67.
- [17] Finger, S., & Dixon, J. R. (1989). Review of research in mechanical engineering design. Part II. Representations, analysis, and design for the life cycle. *Research in Engineering Design*, 1(2), 121-137.
- [18] Reymen, I. (2001). *Improving design processes through structured reflection: A domain-independent approach*. Mémoire de maîtrise, Technische Universiteit Eindhoven, Eindhoven.
- [19] Reymen, I., Hammer, D., Kroes, P., van Aken, J., Dorst, C., Bax, M., et al. (2006). A domain-independent descriptive design model and its application to structured reflection on design processes, *Research in Engineering Design* (Vol. 16, pp. 147-173).
- [20] Warfield, J. N. (1986, May). *Domain of science model: evolution and design*. Paper presented at the Proceeding of the Society for General Systems Research.
- [21] Fitzhorn, P. (1988). A computational Theory of Design. *Design Computing*, 2(2).
- [22] Raymer, D. P., & American Institute of Aeronautics and Astronautics. (2006). *Aircraft design a conceptual approach* (4th ed.). Reston, Virg.: American Institute of Aeronautics and Astronautics.
- [23] Hale, M., Mavris, D., & Carter, D. (1999, 19-21 Oct.). *The Implementation of a Conceptual Aerospace Systems Design and Analysis Toolkit*. Paper presented at the World Aviation Conference, San Francisco, CA.
- [24] Maurino, M. (1993). *La gestion des données techniques technologie du concurrent engineering* (ed.). Paris: Masson.

- [25] ANSI/EIA-649-1998. (1998). *National Consensus Standard for Configuration Management* (ed.): American National Standards Institute/Electronic Industries Alliance.
- [26] Kidd, M. W., & Thompson, G. (2000). Engineering design change management. *Integrated Manufacturing Systems*, 11(1), 74-77.
- [27] Bhuiyan, N., Gatard, G., & Thomson, V. (2006). Engineering change request management in a new product development process. *European Journal of Innovation Management*, 9(1), 5-19.
- [28] Wright, I. C. (1997). A review of research into engineering change management: Implications for product design. *Design Studies*, 18(1), 33-39.
- [29] Shah, J. J., & Mäntylä, M. (1995). *Parametric and feature-based CAD/CAM : concepts, techniques, and applications* (ed.). New York, N.Y.: J. Wiley and Sons.
- [30] Shah, J. J., & Rogers, M. T. (1988). Expert form feature modelling shell. *Computer-Aided Design*, 20(9), 515-524.
- [31] McMahon, C., & Browne, J. (1998). *CADCAM : principles, practice and manufacturing management* (2nd ed.). Harlow, Angleterre: Addison-Wesley.
- [32] Cunningham, J. J., & Dixon, J. R. (1988). Designing with features: The origin of features, *Computers in Engineering 1988 - Proceedings, Jul 31-Aug 4 1988* (pp. 237-243). San Francisco, CA, USA: Publ by American Soc of Mechanical Engineers (ASME), New York, NY, USA.
- [33] Dixon, J. R. (1988). *DESIGNING WITH FEATURES: BUILDING MANUFACTURING KNOWLEDGE INTO MORE INTELLIGENT CAD SYSTEMS*. Paper presented at the Proceedings of Manufacturing International '88. v 1, Symposium on Product and Process Design; v 2, Symposium on Management and Economics; v 3, Symposium on Manufacturing Systems; v 4, Manufacturing Science of Composites., Atlanta, GA, USA.
- [34] Bidarra, R., Idri, A., Noort, A., & Bronsvort, W. (1998, 13-16 September). *Declarative user-defined feature classes*. Paper presented at the Proceedings of the ASME 1998 Design Engineering Technical Conferences, Atlanta, USA.
- [35] Bidarra, R., & Bronsvort, W. F. (2000). Semantic feature modelling, *CAD Computer Aided Design* (Vol. 32, pp. 201-225): Elsevier Science Ltd, Exeter, Engl.

- [36] Deneux, D. (1999). Introduction to assembly features: An illustrated synthesis methodology. *Journal of Intelligent Manufacturing*, 10(1), 29-39.
- [37] Zimmermann, J. U., Haasis, S., & Van Houten, F. J. A. M. (2002). *Applying universal linking of Engineering Objects in the automotive industry - Practical aspects, benefits, and prototypes*. Paper presented at the 28th Design Automation Conference, Sep 29-Oct 2 2002, Montreal, Que., Canada.
- [38] Zimmermann, J. U., Haasis, S., & Van Houten, F. J. A. M. (2002). ULEO - Universal linking of engineering objects. *CIRP Annals - Manufacturing Technology*, 51(1), 99-102.
- [39] Hummel, K. E., & Brown, C. W. (1989). *The role of features in the implementation of concurrent product and process design*. Paper presented at the Symposium on concurrent product and process design, San Fransisco, Californie.
- [40] Deneux, D., & Wang, X. H. (2000). A knowledge model for functional re-design. *Engineering Applications of Artificial Intelligence*, 13(1), 85-98.
- [41] Fruchter, R., & Demian, P. (2002). CoMem: Designing an interaction experience for reuse of rich contextual knowledge from a corporate memory, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM* (Vol. 16, pp. 127-147): Cambridge University Press.
- [42] UGS. (2005). *Systems Libraries in Assemblies* (ed.). Montréal.
- [43] Shah, J. J., & Rogers, M. T. (1994). A testbed for rapid prototyping of feature based applications. In *Advances in features based manufacturing* (Vol. 20, pp. 423-453): Elsevier Science.
- [44] Shah, J. J., Rogers, M. T., Sreevalsan, P., Hsiiao, D., & Mathew, A. (1990). *The ASU features testbed: an overview*. Paper presented at the Proceedings of the ASME 1990 Computers in Engineering Conference, New-York.
- [45] Hoffmann, C. M., & Joan-Arinyo, R. (1998). On user-defined features. *Computer-Aided Design*, 30(5), 321-332.
- [46] Hoffmann, C. M., & Juan, R. (1993). *Erep -- an editable, high-level representation for geometric design and analysis* (P. Wilson, M. Wozny, M. Pratt ed.). North Holland: Geometric and Product Modelling.
- [47] Bidarra, R., & Bronsvort, W. F. (1999). Validity maintenance of semantic feature models. *Proceedings of the Symposium on Solid Modeling and Applications*, 85-96.

- [48] Ledermann, C., Ermanni, P., Hanske, C., Kelm, R., & Wenzel, J. (2005). Associative parametric CAE methods in the aircraft pre-design. *Aerospace Science and Technology*, 9(7), 641-651.
- [49] Subramani, S., & Gurumoorthy, B. (2005). Maintaining associativity between form feature models. *Computer-Aided Design*, 37(13), 1319-1334.
- [50] Wikipédia. (2007). http://fr.wikipedia.org/wiki/Maquette_num%C3%A9rique. Retrieved 10 Janvier, 2007
- [51] American Society of Mechanical Engineers. (1989). *A structural language format for basic shape description* (ed.). New York, N.Y.: American Society of Mechanical Engineers.
- [52] Québec (Province). Office de la langue française. (1998). *Le grand dictionnaire terminologique* (ed.). Outremont, Québec: CEDROM-SNI.
- [53] Tremblay, T. G. (2006). *La propagation du changement dans un contexte PLM: vers la maîtrise du rôle des associations dans le développement de produits et des processus associés*. Mémoire de maîtrise, École de technologie supérieure, Montréal.
- [54] Tremblay, T. G., Rivest, L., & Lamarche, B. (2005). *Vers une application généralisée du concept de lien technologique en développement de produits : Terminologie des associations*. Paper presented at the MICAD, Paris.
- [55] Tremblay, T. G., Rivest, L., Msaaf, O., & Maranzana, R. (2006, 18 - 22 Septembre 2006). *The role of associations in CAD and PLM for handling change propagation during product development*. Paper presented at the 13th ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING: RESEARCH AND APPLICATIONS, Antibes, France.
- [56] Dassault. (2006). <http://www.3ds.com/fr/products-solutions/plm-solutions/catia/overview/>. Retrieved 31 mai, 2006
- [57] Msaaf, O., & Rivest, L. (2006). *La maquette numérique structurée et son courtier pour une exploitation en automatique de la géométrie en optimisation multidisciplinaire*. Montréal: École de Technologie Supérieure.
- [58] Msaaf, O., & Rivest, L. (2005). *Formalisation des requis pour une gestion structurée de la géométrie en optimisation multidisciplinaire, Projet MOSAIC du CRIAQ*. Montréal: École de Technologie Supérieure.