



**HAL**  
open science

# Modèle de réconciliation des associations et propagation des modifications au sein d'une maquette numérique

Ameni Eltaief

## ► To cite this version:

Ameni Eltaief. Modèle de réconciliation des associations et propagation des modifications au sein d'une maquette numérique. Génie mécanique [physics.class-ph]. Université de Technologie de Troyes; Université de Sousse (Tunisie), 2020. Français. NNT : 2020TROY0015 . tel-03808683

**HAL Id: tel-03808683**

**<https://theses.hal.science/tel-03808683>**

Submitted on 10 Oct 2022

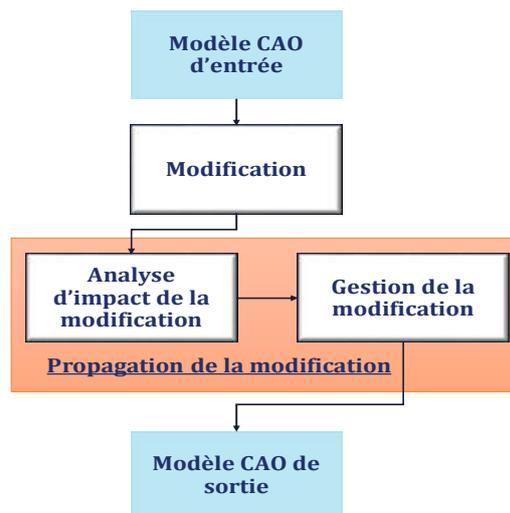
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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

Thèse  
de doctorat  
de l'UTT

**Ameni ELTAIEF**

# Modèle de réconciliation des associations et propagation des modifications au sein d'une maquette numérique



**Champ disciplinaire :**  
Sciences pour l'Ingénieur

2020TROY0015

Année 2020

Thèse en cotutelle avec Université de Sousse - Tunisie



---

---

# THESE

*pour l'obtention du grade de*

## DOCTEUR

de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

EN SCIENCES POUR L'INGENIEUR

**Spécialité : MATERIAUX, MECANIQUE, OPTIQUE, NANOTECHNOLOGIE**

*présentée et soutenue par*

**Ameni ELTAIEF**

*le 5 octobre 2020*

---

---

**Modèle de réconciliation des associations et propagation  
des modifications au sein d'une maquette numérique**

---

---

## JURY

M. Chokri BOURAOUI	PROFESSEUR DES UNIVERSITES DE TUNISIE	Président
M. Frédéric DEMOLY	MAITRE DE CONFERENCES - HDR	Rapporteur
Mme Lilia GZARA	PROFESSEURE ASSOCIEE INP GRENOBLE - HDR	Rapporteur
M. Zouhaier AFFI	PROFESSEUR DES UNIVERSITES DE TUNISIE	Examineur
M. Benoît EYNARD	ENSEIGNANT CHERCHEUR - HDR	Examineur
M. Abdelfattah MLIKA	PROFESSEUR DES UNIVERSITES DE TUNISIE	Examineur
M. Borhen LOUHICHI	MAITRE DE CONFERENCES	Directeur de thèse
M. Sébastien REMY	MAITRE DE CONFERENCES - HDR	Directeur de thèse

## ***Dédicaces***

Afin d'être reconnaissant envers ceux qui m'ont appuyé et encouragé à effectuer ce travail de recherche, je dédie ce mémoire :

À ceux qui sont toujours mon meilleur exemple dans la vie, ma raison d'être : mes chers parents, ***Salem*** et ***Cherifa***, pour les sacrifices qu'ils ont consentis pour mon éducation, pour leur amour et affection.

À mon mari ***Hamdy***, L'homme de ma vie, mon âme sœur et la lumière de mon chemin. Ton soutien, ta gentillesse sans égal, ton profond attachement m'ont permis de réussir mes études.

À toute ma famille, A mes frères Aymen et Marwan, ma sœur Mariem, mes belles sœurs Amel, Sonia et Hanen.

À mes petits anges, mon neveu Omar, ainsi que ma petite nièce Mayar qui ne cessent d'illuminer ma vie par leurs innocences et leurs amours

À tous les enseignants auxquels je dois ce que je suis maintenant.

***Ameni ELTAIEF***

# Remerciements

Cette thèse de Doctorat a été réalisée dans le cadre d'une convention de cotutelle entre l'École Nationale d'Ingénieurs de Sousse (**ENISo**) et l'Université de Technologies de Troyes (**UTT**). Les recherches qui font l'objet de ce mémoire ont été réalisées en Tunisie dans le Laboratoire de Mécanique de Sousse (**LMS**) et en France dans le Laboratoire des Systèmes Mécaniques et d'Ingénierie Simultanée (**LASMIS**) où j'ai passé des périodes d'alternances. Par ces quelques lignes, je tiens donc à remercier toutes les personnes qui ont participé au bon déroulement de cette thèse de près ou de loin, espérant n'avoir oublié personne.

Je tiens à remercier spécialement mon directeur de thèse Français Monsieur **Sébastien Remy**. Je vous remercie d'avoir cru en mes capacités, pour le temps et la patience que vous m'avez accordés. J'ai beaucoup apprécié travailler à vos côtés tant sur le plan scientifique que sur le plan humain.

Je tiens à remercier mon directeur de thèse Tunisien Monsieur **Borhen Louhichi** d'avoir bien assuré la direction et l'encadrement de mes travaux de thèse. Merci pour votre gentillesse, vos qualités humaines, votre patience et vos précieux conseils.

Je souhaite également remercier Monsieur **Abdelfatah Mlika** directeur du Laboratoire de Mécanique de Sousse (LMS) à qui j'adresse mes remerciements les plus sincères pour sa disponibilité, ses conseils, ses encouragements et ses qualités humaines.

Je remercie Monsieur **Frederic Demoly** et Madame **Lilia Gzara** d'avoir accepté de rapporter cette thèse. Vos remarques pertinentes et vos conseils précieux m'ont beaucoup aidé à améliorer la qualité de ce travail. Soyez assurés, de mon estime et de ma profonde gratitude.

Je remercie très chaleureusement tous **les membres de labo LMS** pour la sympathie et l'aide qu'ils m'ont témoignées durant ces années de dur labeur.

Un grand merci aussi pour mes amis de Troyes : *Amani Lamine* et *Monther Rtimi* pour leur aide et leur soutien avec mes sincères vœux de réussite.

# Résumé

Un produit mécanique est essentiellement un ensemble de pièces interconnectées. Ainsi, une modification de l'une des pièces de l'assemblage entraîne des modifications affectant également les autres pièces. Cela peut entraîner des retards importants, voire des erreurs, ce qui augmente les coûts de conception/ fabrication du produit. Le développement d'un outil de propagation automatique des modifications permettrait donc de réduire les coûts et les délais qui peuvent survenir dans un modèle de propagation manuelle. L'objectif principal de cette thèse est d'améliorer et de faciliter la gestion des modifications pendant la phase de conception/ré-conception. En particulier, on s'est positionné à l'intérieur des systèmes CAO utilisés par le designer pour concevoir le modèle du produit. On part d'un modèle CAO existant (contenant tous les paramètres et éléments) et on extrait tous les données (en particulier les données de contraintes d'assemblage). Toutes ces données sont alors utilisées pour établir des associations entre les différentes entités géométriques. Ensuite, si une pièce de l'assemblage est modifiée, l'approche proposée permet de capturer les entités géométriques qui sont susceptibles d'être affectées et de vérifier l'impact prévisible de cette modification ainsi que l'arbre de propagation de celle-ci vers le reste de l'assemblage. Finalement, l'approche présentée est mise en œuvre à l'aide de Visual Studio et cela donne un outil de gestion et de propagation des modifications intégré dans SolidWorks. Différents cas d'études sont utilisés pour tester la validité de notre outil.

## Mots clés

Conception Assistée par Ordinateur (CAO) / Modèles entité-association / Graphes de liaison / Propagation / Évaluation des risques.

# Abstract

A mechanical product is essentially an assembly of interconnected parts. As a result, a modification that affects only one part of the assembly causes modifications affecting other parts as well. This can lead to significant delays or even errors, which increases the costs of the product's design and manufacture. The development of an automatic change propagation tool would therefore reduce the costs and delays associated with the risk of errors that may occur in a manually propagation model. The main objective of this thesis is to improve and facilitate the management of engineering changes during the design or re-design phase. We are interested in the CAD systems used by the designer to design the product model. We will start from an already existing CAD model (containing all the parameters and elements of a given product) and extract existing data (particularly assembly constraints data). Afterwards, all this data is used to establish associations between the different geometrical entities. Then, if a part of the assembly is modified by one of the partners, our proposed approach captures the geometric entities that are likely to be affected and allows us to verify the expected impact of this modification as well as the propagation tree of the intended modification towards the whole assembly. Eventually, the presented approach is implemented using Visual Studio and the result is a simple change management and propagation tool that is integrated into SolidWorks. Different case studies are used to test the validity of our tool.

## Key Words

Computer-aided design / Entity-relationship modeling / Bond graphs/ Propagation / Risk assessment.

# Notations ET Symboles

<b>AIAG</b>	Automotive Industry Action Group
<b>API</b>	Application Programming Interface
<b>BRep</b>	Boundary representation
<b>CAO</b>	Conception Assistée Par Ordinateur
<b>CSG</b>	Constructive Solid Geometry
<b>CPM</b>	Cambridge Advanced Modeler
<b>CAM</b>	Change Prediction method
<b>DSM</b>	Design Structure Matrix
<b>DMM</b>	Domain Mapping Matrix
<b>EC</b>	Engineering Change
<b>CV</b>	Change Vector
<b>ECO</b>	Engineering Change Order
<b>ECM</b>	Engineering Change Management
<b>ECP</b>	Engineering Change Propagation
<b>ECR</b>	Engineering Change Request
<b>ECO</b>	Engineering Change Order
<b>ECT</b>	Engineering Change treatment
<b>EDO</b>	Engineering Design Change
<b>FBS</b>	Function, Behavior, and Structure
<b>GCVP</b>	Gestion du Cycle de Vie du Produit
<b>PDM</b>	Product Data Management

<b>PLM</b>	Product Lifecycle Management
<b>SGDT</b>	Système de Gestion Données Techniques
$\Psi_{ab}$	Dépendance entre deux variables a et b
$\Psi_{ij}^s$	Dépendance Surfaccique entre deux variables i et j
$\Psi_{ij}^V$	Dépendance Volumique entre deux variables i et j
$\Psi_{ij}^{Nb}$	Dépendance Relationnel entre deux variables i et j
<b>IFM</b>	Influence Matrix
<b>IM</b>	Interference Matrix
<b>RM</b>	Relationship Matrix
<b>PM</b>	Probability Matrix
<b>RIM</b>	Risk Matrix

# Table des matières

Résumé .....	iii
Abstract .....	iv
Notations ET Symboles.....	v
Table des matières .....	vii
Liste des Figures.....	xii
Liste des Tableaux.....	xv
Introduction générale.....	1
Chapitre 1 : Contexte et Problématique .....	6
1. Introduction .....	6
2. Motivation industrielle .....	6
3. Problématique de la thèse.....	9
4. Structure de la thèse .....	11
5. Définitions importantes .....	13
5.1 Conception et développement de produit.....	13
5.1.1 Définition .....	13
5.1.2 Les différentes étapes de la conception de produit.....	13
5.2 Conception Assistée Par Ordinateur (CAO) .....	14
5.2.1 Historique .....	14
5.2.2 Définitions.....	15
5.3 Les différents modèles de représentation en CAO.....	17
5.3.1 La modélisation filaire .....	17
5.3.2 Les modélisations surfaciques.....	18
5.3.3 La modélisation volumique .....	18
5.3.3.1. La représentation par occupation spatiale.....	19
5.3.3.2. La représentation par les frontières (BRep) .....	20
5.3.3.3. La représentation CSG.....	21
5.3.3.4. La représentation hybride.....	21

5.3.4	Synthèse sur les modèles de représentation numérique de produit .....	22
5.4	Les différents modes de conception .....	22
5.4.1	Par assemblage, ou Bottom-Up .....	22
5.4.2	Dans l'assemblage, ou Latéral .....	23
5.4.3	<i>Piloté, ou Top-Down</i> .....	24
5.4.4	Utilisations et limitation des différents modes .....	24
5.5	Changement d'ingénierie (EC).....	25
5.6	Propagation des changements (ECP) .....	28
6.	Conclusion.....	31
Chapitre 2 : État de l'art .....		33
1.	Introduction .....	33
2.	Gestion des changements d'ingénierie (ECM).....	33
2.1	Product Lifecycle Management (PLM).....	33
2.1.1	Définition et historique.....	33
2.1.2	Limitations .....	35
2.2	La conception collaborative .....	36
2.2.1	Définition .....	36
2.2.2	Les changements d'ingénierie dans un environnement de conception collaborative	36
2.3	Le management des changements d'ingénierie.....	37
2.3.1	Le processus des changements d'ingénierie.....	38
2.3.2	Caractérisation des changements.....	40
2.3.3	La gestion des changements d'ingénierie.....	41
2.3.3.1	Méthodes traditionnelles .....	42
2.3.3.2	Méthodes multi-compagnies .....	47
2.3.3.3	Méthodes multicouches .....	48
2.4	La propagation des changements .....	50
2.4.1	Méthodes matricielles .....	50
2.4.2	Méthodes nécessitant une base de données .....	53

2.4.3	Méthodes basées sur d'autres concepts sous-jacents .....	54
2.5	Modélisation des dépendances .....	55
2.6	Conclusions par rapport à l'état de l'art de l'ECM .....	55
3.	Les Outils de Comparaison des modèles CAO .....	56
3.1	L'utilité de la comparaison des modèles CAO.....	56
3.2	La comparaison des modèle 3D dans la littérature .....	59
3.3	La comparaison avec les systèmes de CAO 3D .....	62
3.3.1	La comparaison sous SolidWorks .....	63
3.3.2	La comparaison sous Catia V5 .....	64
3.3.3	Conclusion par rapport aux outils proposés dans CATIA et SolidWorks.....	65
4.	Analyse des associations et des objets associés .....	66
4.1	Utilisation des termes dans la littérature .....	66
5.	Conclusion.....	68
Chapitre 3 : Algorithmes proposés pour la propagation des modifications au sein du modèle CAO .....		70
1.	Introduction .....	70
2.	Mise en situation .....	70
3.	Approche proposée.....	71
3.1.	Algorithme de dépendance.....	74
3.1.1.	Extraction des données du modèle CAO.....	74
3.1.1.1.	La Matrice des Relations .....	77
3.1.1.2.	La Matrice des Interférences.....	78
3.1.1.3.	Le graphe des liaisons .....	78
3.1.2.	Arbre de Propagation .....	79
3.1.3.	La Matrice des influences .....	81
3.1.3.1.	Caractérisation d'un changement .....	81
3.1.3.2.	Coefficient d'influence au sein d'un assemblage .....	82
3.1.3.2.1	Coefficient d'influence direct.....	84
3.1.3.2.2	Coefficient d'influence indirecte.....	86

3.1.3.2.3	Coefficient d'influence mixte.....	87
3.1.3.3.	Matrice des Influences .....	88
3.1.4.	Probabilité de Propagation d'une modification.....	89
3.1.4.1.	Probabilité de Propagation directe .....	89
3.1.4.2.	Propagation indirecte.....	91
3.1.5.	Risque de Propagation d'une modification .....	94
3.2.	Algorithme de Comparaison .....	95
3.2.1	Importance de l'outil de comparaison .....	95
3.2.2	Algorithme générale de la comparaison de deux pièces CAO.....	96
3.2.2.1	Extraction des Features .....	97
3.2.2.2	Comparaison des données extraites .....	98
3.2.2.3	Reconnaissance des faces.....	100
3.3.	Algorithme de Réconciliation .....	101
3.3.1	Mise en œuvre des associations initiales .....	101
3.3.2	Mise en œuvre des correspondances .....	102
3.3.3	Réconciliations des associations.....	105
3.4.	Algorithme de Propagation .....	106
3.4.1	Analyse d'impact d'une modification .....	106
3.4.2	Arbre et chemin de propagation de point de vu « Feature » .....	110
3.4.3	Arbre pondéré de propagation des modifications.....	113
4.	Conclusion.....	113
Chapitre 4 : Implémentation et validation.....		115
1.	Introduction .....	115
2.	Environnement de développement .....	115
2.1	Logiciels & bibliothèques utilisés .....	115
2.1.1	SolidWorks.....	115
2.1.1.1	Les API du SolidWorks .....	116
2.1.1.2	Types d'applications de l'API SolidWorks.....	117
2.1.1.2.1	Les macros du SolidWorks.....	117

2.1.1.2.2	Les add-ins.....	117
2.1.1.2.3	Les stand-alone .....	118
2.1.1.3	SDK de l'API SolidWorks .....	118
2.1.1.4	Personnalisation de l'interface utilisateur SOLIDWORKS .....	119
2.2	Langage de programmation.....	120
3.	Implémentation de l'outil « CAD Change propagation Tool ».....	120
4.	Implémentation de l'outil « CAD Part Comparaison » .....	123
5.	Validation .....	126
5.1	Premier cas d'étude : Roue de Remorque .....	126
5.1.1	L'extraction des données.....	127
5.1.2	Comparaison et localisation des modifications .....	130
5.1.3	Réconciliation des Associations.....	131
5.1.4	Risque et Chemin de Propagation de la modification .....	132
5.2	Deuxième cas d'étude : Mécanisme d'un micromoteur.....	135
5.2.1	L'extraction des données.....	136
5.2.2	Comparaison et localisation des modifications .....	138
5.2.3	Réconciliation des associations.....	140
5.2.4	Risque et Chemin de Propagation de la modification .....	141
6.	Conclusion.....	142
	Conclusion générale .....	145
	Références .....	149
	Annexes .....	159

# Liste des Figures

Fig.1. 1. Structure générale de la thèse.....	12
Fig.1. 2. La progression temporelle de la CAO. ....	15
Fig.1. 3. Ingénierie Assistée par Ordinateur.....	16
Fig.1. 4. Modélisation surfacique vs fil de fer. ....	18
Fig.1. 5. Modélisation par occupation spatiale[18]. ....	20
Fig.1. 6. Modèle BREP [12].....	20
Fig.1. 7. La représentation CSG [12]. ....	21
Fig.1. 8. Modes de conception en CAO « Bottom-Up » [19]. ....	23
Fig.1. 9. Modes de conception en CAO « Latéral » [19]. ....	23
Fig.1. 10. Modes de conception en CAO « Top-Down » [19]. ....	24
Fig.1. 11. Les différents niveaux de dépendance. ....	29
Fig.2. 1. Les 3 grandes phases de l'origine des systèmes PLM.....	35
Fig.2. 2. Le processus de gestion des modifications proposées par Jarratt et al.[20].....	39
Fig.2. 3. L'approche du CPM proposé par Clarkson et al. [44]. ....	44
Fig.2. 4. Implémentation de l'outil proposé par Reddi et al. (2009) [66] ....	46
Fig.2. 5. Design structure Matrix. ....	51
Fig.2. 6. Méthode proposée par Flanagan et al. [89].....	52
Fig.2. 7. La relation entre le niveau de détail exigé et la cardinalité et les Fonctions de base qui en résultent [73].....	58
Fig.2. 8. Les différents types de signatures existantes. ....	61
Fig.2. 9. Visualisation des différences entre les 2 modèles en utilisant l'outil « Comparer » du SolidWorks.....	64
Fig.2. 10. Visualisation des différences entre les 2 modèles en utilisant l'outil « Compare parts » sous Catia. ....	65
Fig.3. 1. Le processus général. ....	72
Fig.3. 2. L'algorithme général du modèle proposé. ....	73
Fig.3. 3. L'Algorithme de Cohérence. ....	75
Fig.3. 4. Les différents types d'interfaçages. ....	76
Fig.3. 5. Exemple d'un graph des liaisons. ....	79
Fig.3. 6. Construction d'un arbre de propagation.....	81

Fig.3. 7. Les différents types d'influence qui peuvent exister au sein d'un assemblage. ....	84
Fig.3. 8. Phénomène d'influence entre une Pièce i et Pièce j indirectement connectées. ....	86
Fig.3. 9. Phénomène d'influence mixte au sein d'un assemblage.....	87
Fig.3. 10. La non réciprocity du phénomène d'influence entre une Pièce i et Pièce j. ....	89
Fig.3. 11. Exemple de système avec différents nombre de liens. ....	91
Fig.3. 12. Arbre de probabilités de propagation.....	92
Fig.3. 13. Construction de la matrice de Probabilité.....	94
Fig.3. 14. Matrice de Risque de Propagation d'une modification proposée. ....	95
Fig.3. 15. Algorithme général de Comparaison. ....	97
Fig.3. 16. Algorithme de comparaison des Features. ....	99
Fig.3. 17. Reconnaissance des faces. ....	100
Fig.3. 18. Algorithme de détermination des Associations de Features. ....	102
Fig.3. 19. Exemple des listes de Features associées.....	101
Fig.3. 20. Le principe de correspondances utilisées par Louhichi et al. [102].....	103
Fig.3. 21. Algorithme de détermination de correspondances entre les Features.....	104
Fig.3. 22. Principe de correspondance entre les Features. ....	104
Fig.3. 23. Schéma explicatif de la réconciliation des associations. ....	105
Fig.3. 24 . Exemple d'un diagramme d'araignée. ....	109
Fig.3. 25 . Algorithme de création de l'arbre de propagation. ....	111
Fig.3. 26 . Exemple d'un arbre de propagation. ....	112
Fig.4. 1. Communication de deux applications via les API. ....	118
Fig.4. 2. Première interface utilisateur de l'outil « CAD CPT ». ....	121
Fig.4. 3. Interface de l'extraction des données de l'outil « CAD CPT ». ....	121
Fig.4. 4. L'interface « Extracted Data » de l'outil « CAD CPT ». ....	122
Fig.4. 5 : L'interface « Associations » de l'outil « CAD CPT » .....	122
Fig.4. 6. Interface d'analyse d'impact de la modification proposée de l'outil « CAD CPT ». ....	123
Fig.4. 7 : Interface de démarrage de l'outil de Comparaison des Pièces CAO « CAD Part Comparison ». ....	124
Fig.4. 8. Interface de visualisation des résultats de l'outil de Comparaison des Pièces CAO « CAD Part Comparison ». ....	125
Fig.4. 9. Modèle CAO d'une Roue de Remorque. ....	126
Fig.4. 10. Modification apporté à l'Essieu fixe. ....	127

Fig.4. 11. Les données extraites. ....	128
Fig.4. 12. Graphe des liaisons du système Roue de Remorque et son matrice d'adjacence...	128
Fig.4. 13. Arbre de propagation des modifications du Composant C7 (Essieu fixe).....	129
Fig.4. 14. Matrice d'influence/ Matrice de Probabilité.....	129
Fig.4. 15. Résultats de la comparaison des deux versions de l'essieu fixe (Avant/Après modification). ....	130
Fig.4. 16. Les listes des Associations.....	131
Fig.4. 17. Risque de propagation de modification au sein du système Roue de remorque....	132
Fig.4. 18. Analyse d'impact de la modification.....	132
Fig.4. 19. Capture de l'exécution de l'outil "CAD Propagation Tool". ....	133
Fig.4. 20. Arbre de propagation de la modification de l'Essieu fixe vers le reste du système. ....	134
Fig.4. 21. Modèle CAO d'un micromoteur.....	135
Fig.4. 22. Schéma explicatif de la modification apportée au Piston. ....	136
Fig.4. 23. Graphe des liaisons du Micromoteur et son matrice d'adjacence. ....	136
Fig.4. 24. Arbre de propagation des modifications du Piston. ....	137
Fig.4. 25. Matrice des Influences et Matrice de Probabilité de Propagation des modifications au sein du mécanisme micromoteur. ....	138
Fig.4. 26. Les résultats de comparaison des deux versions du Piston-7. ....	139
Fig.4. 27. Correspondance des Features.....	140
Fig.4. 28. La liste des Features Associées du système Micromoteur.....	141
Fig.4. 29. Risque de propagation de modification au sein du système d'un micromoteur. ....	142
Fig.4. 30. Arbre de propagation d'une modification qui a touché le piston vers le reste du système micromoteur. ....	144

# Liste des Tableaux

Tab.1. 1. Les définitions d'EC existantes dans la littérature.....	25
Tab.2. 1. .Les fonctions de base de la comparaison des modèle 3D proposé par[74].....	59
Tab.3. 1.Outil d'évaluation des conditions du changement. ....	110

# Introduction générale

Nous soulignons que les travaux présentés dans ce mémoire sont effectués dans le cadre d'une cotutelle internationale entre le Laboratoire de Génie Mécanique à l'École Nationale d'Ingénieurs de Sousse (LMS-ENISO) et le Laboratoire de Systèmes Mécaniques et Ingénierie Simultanée à L'université de technologie de Troyes (LASMIS -UTT).

Le philosophe grec Héraclite, qui est connu pour sa doctrine du changement en tant que phénomène central de l'univers, a affirmé vers l'année 500 avant J.-C. que *"la seule constante est le changement"*. Eh bien, dans le monde de l'ingénierie, c'est certainement vrai et encore plus aujourd'hui que jamais ! Ainsi, on ne peut jamais s'opposer voire même arrêter le processus de changement. Mais d'autre part, il y'a une autre citation qui dit qu'*"Il n'y a rien de négatif dans le changement, si c'est dans la bonne direction"*. Par conséquent, comme les changements ne peuvent pas être arrêtés, ils doivent être bien gérés afin de réduire au maximum leurs effets éventuels et de les rendre bien évidemment dans la bonne direction.

Pour les produits d'ingénierie, les changements peuvent être définis au sens large comme des modifications apportées à des produits qui visent à améliorer, perfectionner ou adapter le produit à de nouvelles exigences, ou à supprimer des erreurs. Ces changements sont essentiels dans le développement de produits complexes, et nous pouvons dire même qu'il n'y a pas d'amélioration de produit sans changement d'ingénierie (Engineering Change EC). Des tendances telles que la réduction du temps de développement, la personnalisation et la complexité croissante ont accentué le rôle important des EC. Par conséquent, la gestion appropriée des EC est devenue une discipline cruciale [1].

Cette thèse rentre dans ce contexte et présente une nouvelle approche de la gestion et la propagation des changements d'ingénierie au sens Conception Assistée par Ordinateur (CAO). Nous s'intéressons dans le cadre de cette thèse aux assemblages CAO. Un assemblage CAO est défini comme un ensemble de pièces interconnectées par des contraintes afin de réaliser la fonctionnalité désirée. Notre approche consiste à retirer une pièce (ou un sous-assemblage) d'un assemblage, apporter de la valeur ajoutée et l'insérer de nouveau dans l'assemblage initial. Cette dernière étape présente plusieurs défis : (i) La gestion des contraintes initialement

présentes entre les pièces CAO ; (ii) La détection des formes (Features) impliquées dans les changements ; (iii) La propagation des changements de la pièce/sous assemblage vers l'assemblage ; (iv) La vérification des cohérences entre les différentes pièces de l'assemblage et validation des modifications apportées. Pour présenter l'approche développée, ce mémoire s'articule autour de quatre chapitres :

Le premier chapitre est constitué de deux grandes parties dont la première consiste à décrire la motivation industrielle du travail ainsi que la problématique abordée durant cette thèse et les questions de recherche que nous allons y répondre. À la fin de cette partie un schéma explicatif de la structure générale de la thèse est présenté. La deuxième partie met en place un état de l'art détaillé sur quatre aspects clés dans nos travaux, à savoir, (1) le développement et la conception d'un produit et en particulier la conception mécanique assistée par ordinateur ainsi que les différents modes des modélisations tridimensionnelles utilisés (2) les Changements d'ingénieries, (3) la propagation des changements d'ingénieries et finalement (4) les différents niveaux de dépendances entre les composants d'un système. Dans une première étape, les différents stades de la conception de produit seront détaillés. Puis la conception mécanique assistée par ordinateur (CAO) sera définie. Ensuite, l'aspect modélisation géométrique en CAO est introduit. Nous mettons l'accent sur trois types de modélisation géométrique, à savoir : la modélisation fil de fer, la modélisation surfacique et la modélisation volumique, tout en montrant les avantages et les inconvénients de chacun des modèles. Ce qui nous a conduits à choisir la technique de modélisation volumique comme outil de modélisation des systèmes CAO que nous adoptons par la suite. En deuxième étapes le terme changement d'ingénieries (EC : engineering change) sera définie. Nous essayons de relever les différentes définitions existantes dans la littérature. Un tableau récapitulatif sera présenté. De même pour le terme propagation d'une modification ou d'un changement. Les définitions les plus intéressantes seront présentées. La dernière partie dans ce chapitre sera consacrée à la présentation des différents niveaux de dépendances entre des éléments ou entités regroupés ensemble. On a bien défini les termes : association, relation, contrainte et lien. Car par la suite de notre travail on a choisi d'utiliser le terme association.

Après avoir clairement précisé l'objectif de notre travail de thèse ainsi que les notions les plus importantes que nous allons aborder, l'objectif du deuxième chapitre est de mener une synthèse bibliographique sur les méthodes de gestion des changements d'ingénieries les plus reconnues. Ce qui a nécessité un passage bref sur le développement progressif des systèmes PLM dès la

première génération des SGDT et PDM jusqu'à aujourd'hui. Les limitations de ces systèmes-là ont mené plusieurs chercheurs à trouver des méthodes de gestion des modifications qui sont détaillées dans la première partie de ce chapitre. Nous avons classifié ces méthodes en traditionnelles, Méthodes et Méthodes multicouches. La deuxième partie est consacrée aux différentes méthodes de propagation des changements qui se divise entre : Méthodes matricielles, Méthodes nécessitant une base de données, et Méthodes basées sur d'autres concepts sous-jacents. La section suivante de ce deuxième chapitre est dédiée aux outils de Comparaison des modèles CAO. On a essayé de mettre l'accent sur l'utilité de la comparaison des modèles CAO, les différentes méthodes de comparaison des modèles 3D existantes dans la littérature ainsi que les outils proposés dans CATIA et SolidWorks. La dernière partie est réservée à l'analyse des associations et des objets associés aussi que l'utilisation de ces termes dans la littérature. Enfin, ce chapitre va permettre de recenser les principales techniques de gestion et propagation des modifications existantes dans la bibliographie ainsi que les des outils de comparaisons des modèles CAO. Ces dernières feront l'objet d'une base solide à nos travaux de recherche relatifs à la propagation d'une modification qui 'a touché l'un des composants d'un système CAO. Ce qui nous a permis de tracer nos algorithmes proposés, objet du chapitre suivant.

Le troisième chapitre fera l'objet du développement de l'algorithme général proposé durant cette thèse qui va nous permettre par la suite de propager une modification affectant une pièce d'un assemblage CAO tout en conservant la cohérence existante initialement. Cet algorithme se compose principalement de quatre sous-algorithmes dont le premier est dédié à la partie la plus importante de ce travail qui est l'extraction et le traitement des données du système CAO initiale. Ces données sont présentées sous une forme matricielle, puis ces matrices sont utilisées afin de concevoir un graphe de liaison. Ce dernier est parcouru en largeur pour déterminer les différents chemins de propagation en changeant à chaque fois du nœud source ou initiateur. Le second algorithme est l'algorithme responsable de la partie comparaison : on compare les deux versions de la pièce source de modification initiale/modifiée afin de préciser les Features et les entités qui sont affectés. Le troisième algorithme est celui qui va nous permettre de déterminer les associations existantes initialement pour que nous puissions finalement les réconcilier. Finalement le dernier algorithme est consacré à la partie propagation des modifications. C'est l'algorithme responsable de l'étude d'impact d'une modification proposée, le calcul des probabilités et risques de propagation et finalement de donner les chemins de propagation.

Notre approche proposée est développée en utilisant des outils informatiques pour obtenir un add-in intégrable dans SolidWorks et bien fonctionnel. Le quatrième chapitre montrera clairement les différents outils informatiques et langages de programmation utilisés pour la mise en œuvre du modèle. Et finalement, en vue d'une validation de l'outil proposé, ce même chapitre montrera l'étude de cas de gestion et propagation des modifications au sein d'un système CAO représentant un modèle mécanique ayant un nombre important de composants reliés entre eux avec des liens et des contraintes bien solides. L'analyse des résultats obtenus permettra de valider le fonctionnement de notre outil proposé et de mettre en évidence les principaux avantages de notre approche, mais aussi de souligner les difficultés rencontrées. Ces difficultés feront l'objet des perspectives de recherches issues de ce travail dans le cadre de mastères ou de thèses.

# **Chapitre 1 :**

## **Contexte & Problématique**

# Chapitre 1 : Contexte et Problématique

## 1. Introduction

Les industriels dans le secteur mécanique sont en quête de moyens pour renforcer leur compétitivité. Il s'agit de réduire les coûts et le temps associés à la conception ainsi qu'à la fabrication de nouveaux produits. Cette compétitivité est fortement liée aux processus internes de chaque entreprise. Lors du processus de conception, la définition d'un produit est en constante évolution et elle continue de se modifier, même après que le produit est mis en service. Cela est expliqué par la valeur ajoutée au produit afin de satisfaire le client qui est devenu de plus en plus exigeant. Un produit mécanique est généralement un assemblage de pièces interconnectées. Il en résulte qu'une modification apportée à une pièce induit également des modifications affectant d'autres pièces. Cela peut engendrer des délais, voire, des erreurs, ce qui augmente les coûts de conception et de fabrication des produits. La mise au point d'un système de propagation automatique des changements réduirait donc les coûts et les délais associés au risque d'erreurs pouvant survenir dans un modèle produit dans le cadre d'une propagation manuelle. Dans ce chapitre on va détailler plus la motivation industrielle de notre sujet de thèse, ainsi, présenter une étude bibliographique qui nous permet de bien définir les termes « Changement d'ingénierie » et « Propagation des modifications ».

## 2. Motivation industrielle

Les processus de développement des produits et la rentabilité sont aujourd'hui très liés et ils ont un impact direct sur la croissance financière de l'entreprise. De plus, le marché d'aujourd'hui est caractérisé *par une concurrence importante, par des groupes de travail géographiquement dispersés, sans oublier les exigences des clients*. En effet, le produit doit satisfaire les besoins des clients en ce qui concerne les différents aspects esthétique, économique, confort, etc. Ces aspects récemment cités représentent les véritables défis, qui concernent à la fois le nombre croissant des parties impliquées et la nécessité de faire face à des changements de conception qui deviennent plus fréquents ou de respecter les réglementations industrielles rigoureuses. Il est donc essentiel de les prendre en compte et de les traiter.

- Le premier facteur, c'est la concurrence qui devienne de plus en plus féroce et qui oblige les entreprises à raccourcir au maximum le cycle de développement de leurs produits. De ce fait, toute modification apportée à un produit dans un tel marché nécessite une gestion efficace de l'information au travers de toutes les strates de l'entreprise. Le grand défi est donc d'introduire de nouveaux produits ou des produits améliorés sur le marché *aussi rapidement que possible* en atteignant au plus tôt la production de masse. Sachant que la prise en compte d'une modification peut même engendrer des modifications de la totalité des solutions logiques et /ou physiques. D'après Martinet al. [1], il est désormais beaucoup plus intéressant financièrement pour les entreprises de lancer en premier un produit sur le marché (afin de gagner autant de parts de marché que possible), plutôt que de se contenter simplement un produit optimal en termes de qualité et de rentabilité.
- Le deuxième facteur, consiste aux groupes de travail géographiquement dispersés. En fait, la définition complète d'un produit nécessite souvent la collaboration des différents partenaires qui ne se trouvent pas nécessairement dans un même périmètre. Le partage et l'échange de données entre ces partenaires est donc devenu une tâche importante tout au long du cycle de vie d'un produit. Louhichi et al.[2] ont souligné l'importance de garder la cohérence des données échangées dans des telles conditions.
- Aussi, avec la diversification des besoins des consommateurs (le troisième facteur), le monde industriel doit s'adapter aux changements de produit qui se manifestent de manière plus fréquente mais surtout plus rapide. Les entreprises ont donc été amenées à faire de nombreux retours en arrière dans le cycle de développement des produits, pour les pouvoir prendre en compte. Cette tâche est devenue de plus en plus complexe avec le nombre accru des intervenants car les demandes de changements sont réalisées par le client (personnalisation), par le concepteur lui-même (amélioration) ou d'autres parties prenantes, et avec l'augmentation importante de flux de données. Par exemple, pour les produits aussi complexes que les véhicules et les aéronefs, nous pourrions simplement comprendre la quantité astronomique de données à gérer en modifiant la conception de ces produits pour les optimiser ou simplement pour offrir une configuration personnalisée, dans le but de donner plus de souplesse aux désirs du client, car aujourd'hui chaque produit demande un certain degré de personnalisation[3], [4].

Il est toutefois important de noter que les changements peuvent généralement se produire dans deux contextes différents [3] :

- L'amélioration de produit qui désigne le fait de changer un système ou un produit afin de le mettre à niveau ou de le faire évoluer comme l'ajout des nouvelles fonctionnalités (wifi à bord d'un bus, vitre fumée, etc.).
- La ré-conception de produit qui correspond au fait d'appliquer un ensemble de modifications qui sont relatives à sa structure de base, son comportement, ou ses fonctions afin d'obtenir une nouvelle version. Par exemple, la ré-conception d'une voiture ou bien la création d'une nouvelle génération ou nouvelle version. Tandis que le premier cas de modification peut être considéré comme la création d'une pièce personnalisée sans changer la structure de base du système.

La gestion efficace et rationnelle des changements techniques est un objectif stratégique important. Il en découle que, quel que soit le contexte du changement demandé, les conséquences économiques et techniques d'une mauvaise gestion de ce dernier sont importantes : Des pertes de marché, des retards de livraison, des incompatibilités de montage, les non-respects des exigences fonctionnelles, les non-respects des exigences des clients etc...

D'une part, économiquement parlant, les changements techniques ou d'ingénieries sont très coûteux. En se référant à l'exemple du secteur automobile nord-américain, environ 35 millions changements sont demandés et réalisés par an avec un coût de traitement (sans la prise en compte des coûts des outils et matériaux achetés) de 50000 USD par changement d'après l'Automotive Industry Action Group (AIAG)[5]. Aussi, Une enquête menée en août 2011 par L'Université Technique du Danemark (DTU)/ L'Université Technique Munich (TUM)[6] : montre que les modifications techniques consomment près d'un tiers de la capitale totale de la Recherche et développement (R&D). Dans le même contexte, Potoczak[7] a fait une estimation du coût des changements qui montre que celui-là est allant de 100 000 \$ à près de 500 000 \$ par usine au cours d'une année typique. Également Fleurent et al.[8], ont mentionné qu'il a été possible de constater (des observations menées en entreprise ABC-construction) que l'augmentation de l'envergure du projet a induit des demandes de changements qui ont mené à des dépenses d'heures d'ingénierie de 300 % supérieures à la planification effectuée lors de la signature du contrat avec les clients.

D'autre part, il a été constaté que le délai de mise en œuvre des changements dans les systèmes de gestion réactifs des changements d'ingénierie est très long, la durée moyenne d'une modification technique en phase d'analyse d'impact du processus étant de 126 jours[9]. Les

conséquences d'un changement ont également été examinées. Il a été constaté que certaines modifications techniques ont eu des incidences dramatiques. Parmi ces effets, 20,1 % ont affecté la nomenclature, 19,6 % la base de référence et 9,6 % la structure, représentant les entités à effet le plus important, soit 49,3 % des effets totaux. En mesurant l'importance des effets des changements on peut conclure l'importance de la gestion de ces derniers. En effet, FRICKE et al.[10]ont montré dans leur étude que 30% des travaux journaliers des ingénieurs dans l'industrie allemande sont reliés directement aux changements d'ingénierie. Maier et al [6]l'ont aussi étudié sur la base d'une enquête menée auprès de plus de 90 sociétés d'ingénierie de différents secteurs industriels et de différentes tailles au Danemark.

Certaines entreprises ont considéré les systèmes **PLM** (Product Life cycle Management, littéralement « gestion du cycle de vie des produits ») comme une solution potentielle. De nos jours pour les entreprises, le «système PLM» est considéré comme un facteur clé de succès[11].En effet, les entreprises qui s'appuient sur des systèmes PLM peuvent être plus compétitives tout en répondant aux besoins des clients dans un environnement innovant (qui nécessite une commercialisation rapide, une collaboration interentreprises, des produits de haute qualité et des politiques de coûts agressives). Aussi l'ingénierie concurrente ou l'ingénierie simultanée peut être une solution efficace. Elle peut fortement réduire les délais pour lancer un produit sur le marché (time-to-market). Cependant, cette approche nécessite une gestion parfaitement de données échangées et un compromis doit être trouvé entre les gains fournis par cette approche (l'approche concurrente) et les coûts engendrés par l'organisation et le management de données, d'après Loch et al. [12]. Par ailleurs, les dépendances qui existent, avec la possibilité que des modifications importantes soient nécessaires tout au long du cycle de vie du produit, fait peser un risque financier (coût du changement) ou technique (par exemple, défaut de qualité si le changement n'est pas adopté).

Dans le cadre de ce mémoire, nous tenterons de proposer une nouvelle approche pour gérer Les modifications qui affectent un modèle de Conception Assistée par Ordinateur (CAO) à travers son évolution.

### **3. Problématique de la thèse**

La gestion des changements occupe une place importante tout au long de la phase de conception et de ré-conception car une gestion adéquate implique nécessairement des résultats bénéfiques qui aident à améliorer le coût, la qualité et même le délai de mise en marché ou bien le TTM

(time to market). Les travaux de recherche sur ce sujet se sont principalement concentrés sur la minimisation des effets négatifs des Engineering Changes (EC), tels que les retards et les dépassements budgétaires, ce qui a entraîné une perception négative des EC. Cependant, les EC ne sont pas seulement considérés comme un problème mais aussi comme une opportunité [6] : elles permettent aux entreprises bien organisées qui développent leurs produits de répondre rapidement à l'évolution des besoins des clients et d'obtenir de meilleurs résultats que leurs concurrents. Les effets des EC peuvent être bénéfiques lorsque la qualité du produit augmente ou lorsqu'une réduction des coûts est obtenue. En fait, une gestion efficace et efficiente de la chaîne d'approvisionnement (La chaîne d'approvisionnement comprend tous les processus impliqués dans la fabrication d'un produit depuis sa conception jusqu'à sa livraison au client.) procure aux entreprises un avantage concurrentiel. Les entreprises qui adoptent des processus, des outils et des techniques appropriés pour contrôler et mettre en œuvre les EC améliorent leur compétitivité dans les trois aspects du coût, de la qualité et du délai. Ainsi, les avantages d'un bon Engineering Change Management (ECM) sont doubles : ils évitent les coûts engendrés par les EC et génèrent des gains supplémentaires.

Cela étant posé, la gestion en elle-même est une opération très complexe et bien évidemment très coûteuse. Ajoutons à cela un effet boule de neige de la propagation des modifications, la gestion d'un changement donné va engendrer la gestion d'autre changement et par la suite des impacts importants sur la totalité du produit.

Comme nous l'avons déjà dit, la gestion des documents techniques est une partie très importante dans le cycle de vie d'un produit et elle est assurée par les systèmes PLM/PDM. Dans la suite de ces travaux de thèse, nous nous sommes concentrés sur des documents de type CAO. L'objectif principal de cette thèse est d'améliorer et faciliter la gestion des changements d'ingénierie. Au regard de la complexité du cycle de vie d'un produit et des multiples phases de celui-ci, nous nous concentrons uniquement sur la gestion de ces changements durant la phase de conception ou de ré conception. En particulier, nous nous positionnons à l'intérieur des systèmes CAO utilisés par le concepteur afin de concevoir le modèle du produit. Notre point de départ est un modèle CAO déjà existant contenant tous les paramètres et éléments d'un produit qui sont par la suite assemblés sous forme de sous-ensemble à l'aide des différents liens imposés ou définies.

#### 4. Structure de la thèse

Cette recherche a pour objectif général, comme indiqué plus haut, d'améliorer et faciliter l'ECM (la gestion des changements d'ingénierie). Cet objectif est très global et admet de nombreuses questions de recherche potentielles. Afin de réduire le champ de la recherche et de se concentrer sur les domaines les plus intéressants, l'hypothèse suivante, fondée sur la discussion de la partie motivation industrielle a été utilisée :

**Hypothèse de recherche :** La capacité prédictive des méthodes de l'ECM peut être améliorée par une modélisation plus détaillée des liens existants et des associations entre les différents composants du modèle.

##### *Q.1 Quelle est la signification actuelle des EC et des ECM ?*

On a répondu à cette question en examinant les publications les plus importantes sur les EC et les ECM et en exposant en détail les principaux éléments. La connaissance acquise à partir de la réponse à Q1 a permis de déterminer la formulation de la deuxième question de recherche :

##### *Q.2 Quelles sont les avancées les plus récentes en matière d'ECM et quelles sont les méthodes d'ECM existantes ?*

La réponse à cette question a été obtenue par une recherche bibliographique et une catégorisation méthodique de la littérature qui a permis l'identification des méthodes ECM disponibles. Alors que la réponse à Q1 a permis de mieux comprendre les EC et les ECM, la réponse à Q2 a fourni les publications et la base de données des méthodes ECM pour cette recherche. Les deux réponses ont guidé le cours de cette recherche vers le développement d'une méthode ECM et ont déterminé les autres questions de recherche :

##### *Q.3 Quels sont les éléments de base de notre modèle et quelles sont les données du modèle CAO nécessaires à la réalisation d'un modèle d'ECM ?*

La question à cette réponse est la partie la plus importante, en effet nous devons déterminer les éléments de base de notre modèle, le type de modélisation utilisé par le système CAO choisi, ainsi que les données CAO nécessaires pour la réalisation du modèle. Après avoir répondu à cette question, on se trouve devant la nécessité de préciser les outils utilisés pour la récupération

de ces données ainsi que devant la construction et l'implémentation du modèle ce qui a permis la formulation de la quatrième question de recherche :

**Q.4 Quels sont les outils utilisés afin de développer le modèle ?**

Une fois l'identification des données nécessaires réalisée et la formulation de base de l'algorithme du modèle finie, nous passons vers l'implémentation qui nécessite bien évidemment de définir les outils, les logiciels et le langage de programmation utilisé pour réaliser le modèle ainsi que l'outil informatique désiré.

**Q.5 Quelle est la performance du modèle développé dans une étude de cas réel ?**

Enfin, la réponse à cette question exigeait une application à la pratique et l'évaluation de la méthode développée.

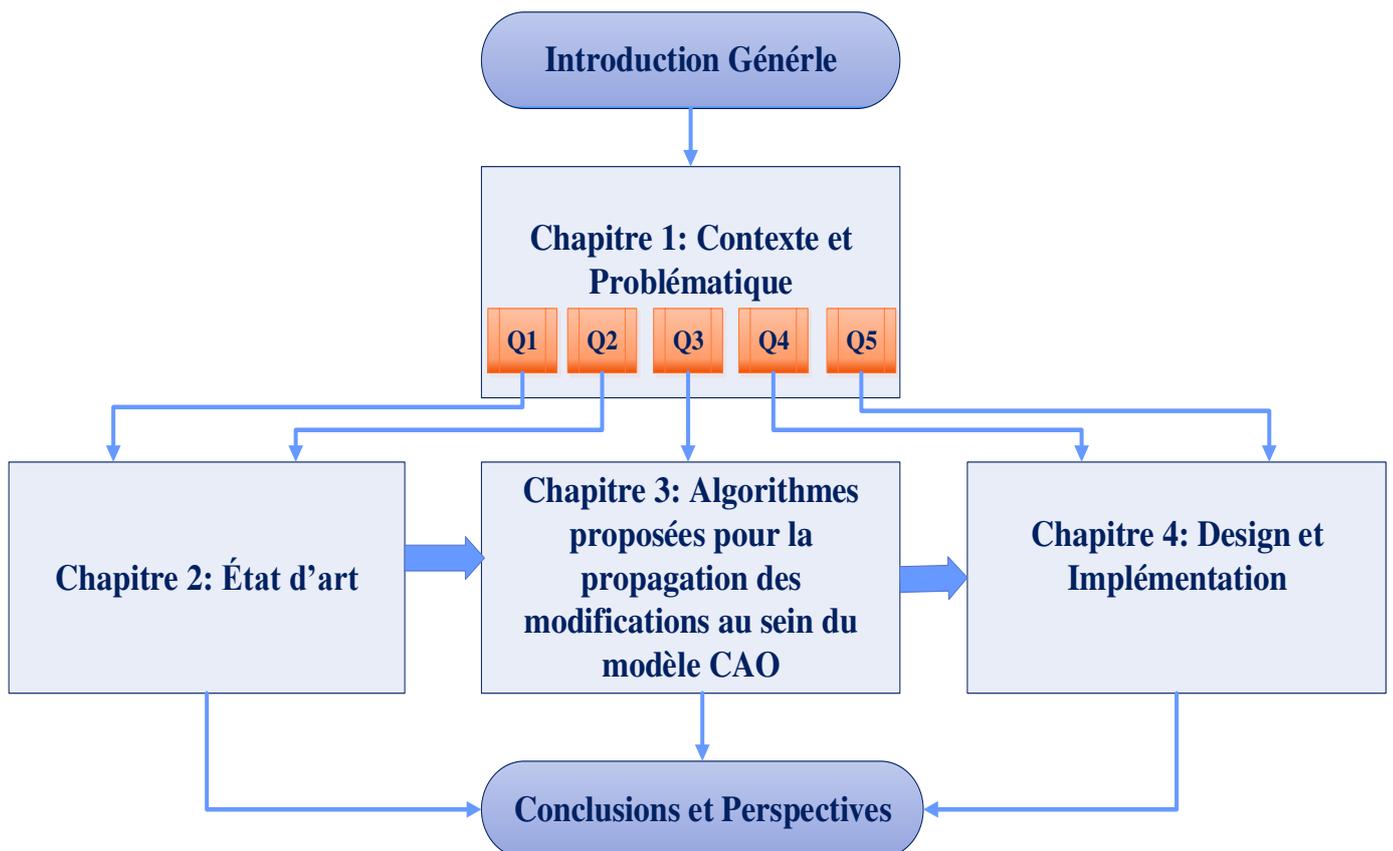


Fig.1. 1.Structure générale de la thèse.

## 5. Définitions importantes

### 5.1 Conception et développement de produit

#### 5.1.1 Définition

« La conception et le développement » est un ensemble d'activités créatrices qui, partant des besoins exprimés et des connaissances existantes, aboutissent à la définition d'un produit satisfaisant ces besoins et qui seraient par la suite industriellement réalisables. La conception de produits peut être considérée comme un processus complexe, une suite de nombreuses étapes de travail, de l'idée à la réalisation pratique, au travers de la création, de la simulation, optimisation, tests, etc. Dans le monde industriel, la conception de produit consiste à inventer, développer et commercialiser sur un marché, un bien ou un service nouveau, le temps mis pour ce processus étant appelé "Time to market"[13].

La norme ISO 9000:2005[14] définit la conception et le développement de produits comme un ensemble de processus qui transforme des exigences en caractéristiques spécifiées ou en spécification d'un produit, d'un processus ou d'un système.

#### 5.1.2 Les différentes étapes de la conception de produit

Le développement de produit passe par plusieurs étapes nécessitant un ensemble de compétences et des moyens spécifiques. Parmi ces différentes étapes, dans l'ordre chronologique, on peut noter[15] :

- La définition du cahier des charges fonctionnel (expression du besoin, analyse fonctionnelle, analyse de la valeur).
- La recherche de solutions innovantes pour répondre au besoin (méthode Cetinnoy, méthodes de créativité, veille technologique et concurrentielle, ...)
- La conception : conception du design 3D du produit (CAO), définition des différents composants, choix des matériaux, choix des techniques d'assemblage, ...
- La simulation numérique : validation ou optimisation du dimensionnement, optimisation des fonctionnalités du produit, validation de la CAO et/ou du choix des matériaux utilisés...
- La réalisation de prototypes : maquette de validation esthétique, prototype fonctionnel, prototype de présérie (bonne matière et bon procédé), ...

- *Les essais de validation sur prototype*, dans des conditions réelles d'utilisation ou dans des conditions représentatives spécifiquement définies : validation fonctionnelle, validation du dimensionnement, validation des simulations numériques, ... Les résultats de cette dernière étape peuvent rendre nécessaires des modifications ou des améliorations sur la conception du produit, comme un redimensionnement ou un changement de matériau.

Parmi toutes ces étapes citées précédemment, c'est la phase de conception qui nous intéresse et sur laquelle nous allons nous concentrer dans la section qui suit.

## **5.2 Conception Assistée Par Ordinateur (CAO)**

La conception assistée par ordinateur (CAO) comprend notamment tous les logiciels et techniques de modélisation géométrique utilisés pour concevoir, tester virtuellement - à l'aide d'un ordinateur et de techniques de simulation numérique - ainsi que pour fabriquer les produits manufacturés et les outils permettant de les fabriquer [16].

### **5.2.1 Historique**

La CAO est le fruit de travail de deux personnes en particulier, Patrick Hanratty et Ivan Sutherland, qui ont largement participé au développement de ce que nous appelons aujourd'hui la CAO ou Conception Assistée par Ordinateur. On peut également citer Pierre Bézier, ces travaux sont les bases de nombreux logiciels utilisés aujourd'hui.

Patrick J. Hanratty est largement crédité comme "le Père de CAD". En 1957, il a développé pour la société General Electric un programme PRONTO (Program for Numerical Tooling Operations) c'était le premier langage de programmation de contrôle numérique.

Quant à Yvan Sutherland, il présente en 1963 sa thèse de doctorat intitulée « Sketchpad » dans laquelle il développe le précurseur des programmes de conception assistée par ordinateur. Ce premier programme a comme caractéristique d'utiliser un crayon optique pour manipuler des objets affichés sur un écran cathodique. C'est le début de l'interface graphique et de la modélisation 3D. Avec l'émergence des stations de travail UNIX au début des années 80, les systèmes de CAO se développent énormément et sont surtout utilisés par l'aérospatiale, le secteur automobile et les industries. Les recherches de Pierre Bézier aboutirent à un logiciel, appelé Unisurf, breveté en 1966. Il est à la base de nombreux logiciels de CAO et de CFAO créés par la suite, comme CATIA.

La CAO décolla dans les années 70-90, lorsque le coût de mise en place d'un poste se rapprocha du coût annuel d'un dessinateur (Fig.1. 2). Les premiers travaux sur les systèmes de Dessin Assisté par Ordinateur (DAO) étaient en 1963 par Ivan Sutherland ; puis on a vécu l'évolution du DAO vers la CAO en y intégrant des fonctions de calculs exploitant la maquette virtuelle ainsi que l'évolution des interfaces de communication et de visualisation avec les systèmes de CAO. Le domaine de la mécanique est historiquement l'un des premiers à s'être doté de logiciels de CAO.

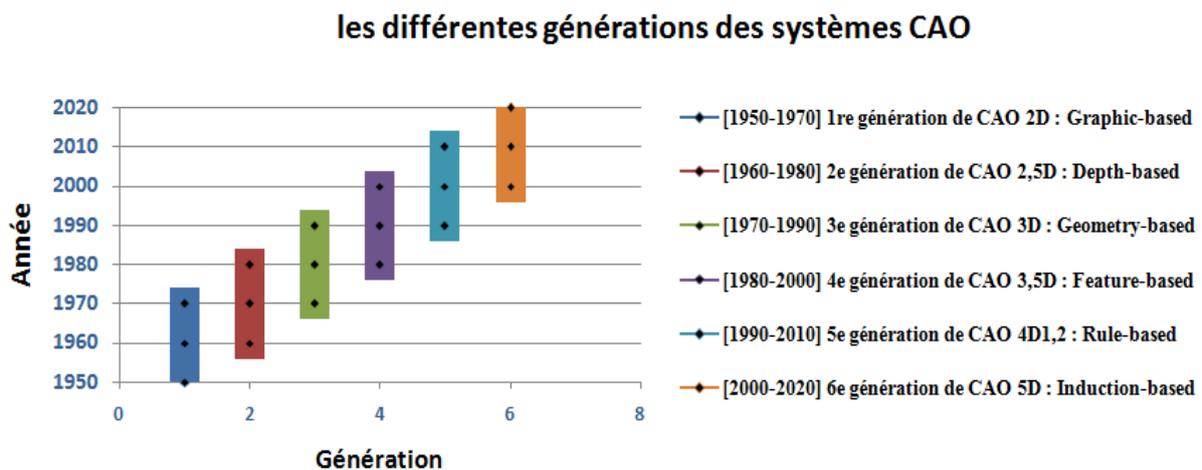


Fig.1. 2. La progression temporelle de la CAO.

La CAO a considérablement influencé la vitesse de développement des produits conçus. Les modifications appliquées aux produits s'effectuent maintenant beaucoup plus rapidement qu'autrefois. Le dessin assisté par ordinateur en deux dimensions était essentiellement utilisé comme moyen de documentation statique pour la description de pièces. Maintenant, le modèle géométrique 3D contient une foule de propriétés et d'informations rattachées dès sa création.

### 5.2.2 Définitions

En général, la Conception Assistée par Ordinateur (CAO) comprend l'ensemble des outils logiciels visant à assister dans son travail tout concepteur, tous domaines confondus (mécanique, électronique, informatique, etc.). Tout ceci entre dans le cadre de l'IAO, on parle d'Ingénierie Assistée par Ordinateur (ou bien L'ingénierie numérique) (Fig.1. 3). Cette dernière comprend toutes les ressources numériques et logicielles habituellement utilisées par les ingénieurs de bureau d'études afin de concevoir, simuler et valider des nouveaux produits. Les solutions d'ingénierie assistée par ordinateur (IAO) permettent aux concepteurs, ingénieurs et

analystes de concevoir (CAO), de simuler la performance des produits et processus dans l'analyse d'éléments finis (FEA), les calculs de dynamique des fluides (CFD) et l'analyse dynamique multi corps (MBD). Elles intègrent également des outils d'estimation de coûts et d'optimisation de la conception (GPAO).

La CAO est une technique qui consiste à rassembler les qualités de l'homme et de l'ordinateur afin que l'équipe ainsi formée travaille mieux que chacun séparément. La CAO est une méthode qui consiste à modéliser les systèmes physiques, ce qui permet dans un premier temps l'analyse interactive et automatique de différentes variantes d'un concept, et dans un second temps, l'expression de ce concept dans une forme adéquate pour sa fabrication. La CAO permet de concevoir des systèmes complexes dont le niveau de complexité est plus important que la capacité de l'être humain. La conception permet de faire une évaluation intégrale des performances de l'objet créé avant même qu'il n'existe. En CAO, on ne dessine pas, on construit virtuellement tout un modèle qui est capable de réagir dans son environnement non-réel selon des lois régies par le logiciel. Le résultat, qu'on appelle maquette numérique, est alors un véritable prototype en constante évolution.

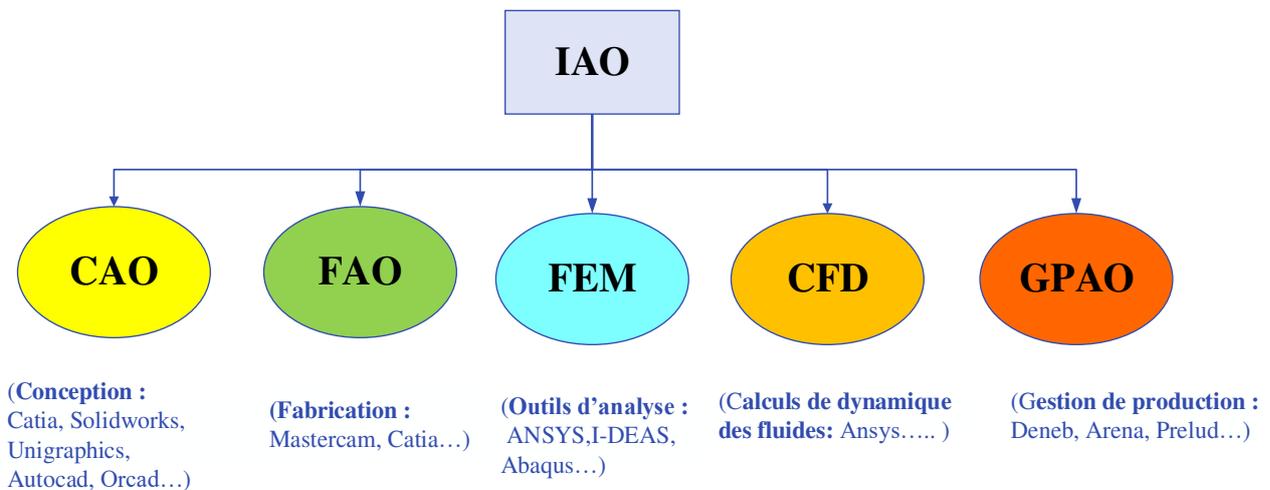


Fig.1. 3. Ingénierie Assistée par Ordinateur.

Les logiciels de CAO aident non seulement à créer des pièces mécaniques, ou à les fabriquer, mais aussi à simuler leur comportement, et donc à valider les solutions retenues. Dès que la création est terminée, l'exécution automatique de la mise en plans (tout en respectant les conventions de représentation), est devenue triviale. Le passage en mode 2D ne pose plus de problème. La réalisation de plans techniques permettant de comprendre le fonctionnement demeure plus complexe tandis que la cotation fonctionnelle et la cotation selon le principe de

l'indépendance sont maintenant plus faciles. Ce qui a été conçu lors de la conception de la pièce est ramené à la bonne cote au moment du dessin automatique. Alors que ce travail est supplémentaire pour le technicien utilisant un outil de Dessin assisté par ordinateur (DAO).

### **5.3 Les différents modèles de représentation en CAO**

Il faut distinguer trois types de modélisation 3D : filaire, surfacique et volumique. Ces modélisations offrent une représentation « réelle » des objets. Elles fournissent une assistance extrêmement intéressante lorsque la complexité des formes et/ou des représentations de ces formes devient trop importante pour la technique de dessin 2D classique.

L'une des contraintes les plus importantes que les modèles tridimensionnels doivent résoudre est la nécessité d'une représentation des objets, le plus réel possible, et d'un temps de réponse le plus réduit possible (pour la visualisation). On peut notamment faire varier la modélisation et/ou la représentation utilisée en fonction de l'état d'avancement des travaux où elle est exécutée. Par exemple, dans le même logiciel, « le couplage » des modélisations filaire et surfacique ou surfacique et volumique, ou « le couplage » de différentes représentations : traits ou images (ombrées, colorées).

#### **5.3.1 La modélisation filaire**

Cette modélisation est appelée aussi linéique, treillis ou fil de fer. Il s'agit du premier niveau de modélisation dans l'espace. Elle reprend les mêmes entités géométriques que la représentation 2D, en y ajoutant une troisième dimension. Par conséquent, cette méthode est basée sur des points et des lignes, où les lignes représentent les arêtes et les points représentent les sommets. L'objet est donc décrit par ses sommets et ses arêtes.

Ce type de modélisation permet une représentation "réelle" d'un objet dans l'espace. Les erreurs d'interprétation sont réduites par comparaison avec le modèle 2D (grâce aux informations supplémentaires fournies par la troisième dimension). Cela permet donc de traiter des géométries plus complexes. Pourtant, cette dernière ne comprend pas les notions de surface et de volume, la visualisation obtenue peut nous donner « une idée ». Il n'y a pas d'élimination systématique des arêtes invisibles. Il en résulte cependant des ambiguïtés au niveau de la compréhension de la géométrie de la pièce. La lecture de dessin devient difficile au-delà d'une certaine densité de traits.

### 5.3.2 Les modélisations surfaciques

Cette modélisation est le premier moyen du concepteur de formes car elle a pris en compte la notion de surface dont elle permet aussi bien la représentation et la manipulation. Un objet est défini par, ses surfaces-frontières (son enveloppe). Elle gère les intersections de surface tout en appliquant des règles de contrôle topologique (ouvert/fermé, intérieur/extérieur) pour « créer » des objets.

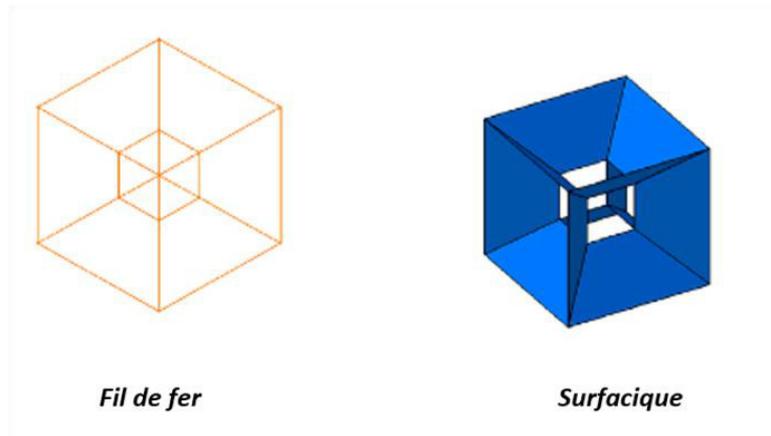


Fig.1. 4. Modélisation surfacique vs fil de fer.

Il existe principalement deux types de modélisation surfacique : *par facettes planes* et *par surfaces gauches*. Lorsque l'objet est représenté par des facettes. On habille une structure filaire avec ces facettes polygonales planes (ou « carreaux »). Il est possible d'obtenir une visualisation améliorée lorsqu'on utilise un grand nombre de facettes. Lorsque la surface à définir est trop complexe, la méthode de la surface gauche est utilisée (plane, cylindrique, sphérique, conique...). On définit la surface grâce à des fonctions polynomiales paramétrées. L'utilisation de telles fonctions donne une très bonne approximation de la surface réelle.

### 5.3.3 La modélisation volumique

Il s'agit de la modélisation la plus complète car elle englobe les deux modélisations précédentes (arêtes et surfaces). Elle permet la représentation dans l'espace, en y intégrant la notion de matière. Pour créer un objet, le 3D volumique : utilise des primitives volumiques, les assemble par des opérateurs logiques, les manipule par des opérateurs géométriques tout en conservant les étapes de la construction.

Les primitives volumiques sont des volumes basiques stockés dans une bibliothèque, dans laquelle chaque utilisateur « pioche » ce qui correspond à ses propres besoins. Les primitives les plus répandues sont parallélépipède, cylindre, sphère, cône, pyramide, tore, polyèdres.... De plus on peut en créer de nouvelles (selon les besoins spécifiques). Elles sont stockées en bibliothèque et sont utilisables au même titre que les primitives initiales.

Les opérateurs logiques correspondent aux opérations booléennes typiques : union, intersection, différence. Ils permettent de combiner les primitives pour créer des solides plus complexes. Tandis que les opérateurs géométriques sont des transformations géométriques classiques : translation, rotation, symétrie, homothétie...

Le solide est caractérisé par son aspect 3D homogène, ses limites définies (le solide occupe un espace fini) et ses frontières qui précisent un intérieur et un extérieur au volume. Nous introduirons dans les paragraphes qui suivent les principaux modèles solides : la représentation par occupation spatiale, la représentation par les frontières, la représentation CSG et la représentation hybride.

### **5.3.3.1. La représentation par occupation spatiale**

L'objet est donc représenté par un ensemble de cellules qui occupent le volume du corps dans l'espace[17]. Ces dernières peuvent être de n'importe quel volume, mais l'utilisation de cubes de taille fixe est très courante. Ces cubes que l'on appelle voxels sont de petits éléments de volume discret. Ils sont le plus souvent positionnés grâce aux coordonnées d'un point 3D. Tout objet est représenté par un tableau de triplets  $(x, y, z)$  appelé tableau spatial, le tableau étant ordinairement ordonné et son arbre correspondant bien à l'ordre dans lequel l'objet est construit. Cette méthode est simple à implémenter. Elle simplifie le calcul des propriétés physiques de l'objet et la mise en œuvre des opérations booléennes (union, intersection et différence). Elle demeure invariante en translation, sans pour autant être en rotation. L'inconvénient majeur de la modélisation de l'occupation spatiale est le fait qu'elle ne donne qu'une approximation plus ou moins grossière de l'objet, en fonction du niveau de résolution qui a été choisi, dont dépend aussi la capacité de la mémoire.

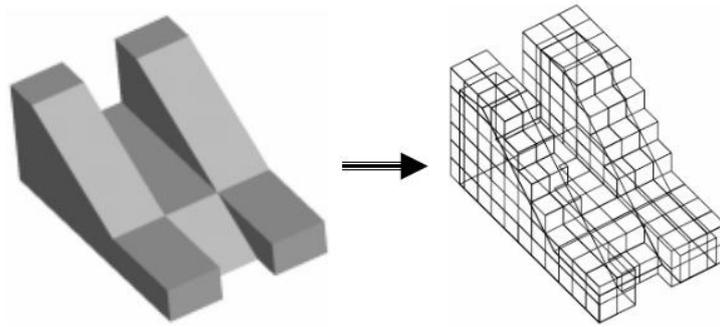


Fig.1. 5. Modélisation par occupation spatiale [18].

### 5.3.3.2. La représentation par les frontières (BRep)

La représentation BRep veut dire “*Boundary representation*”, où un objet est représenté par sa surface : *sommets*, *segments* et *faces*, pas de notion de volume. Le BRep contient des informations géométriques et topologiques importantes sur le solide.

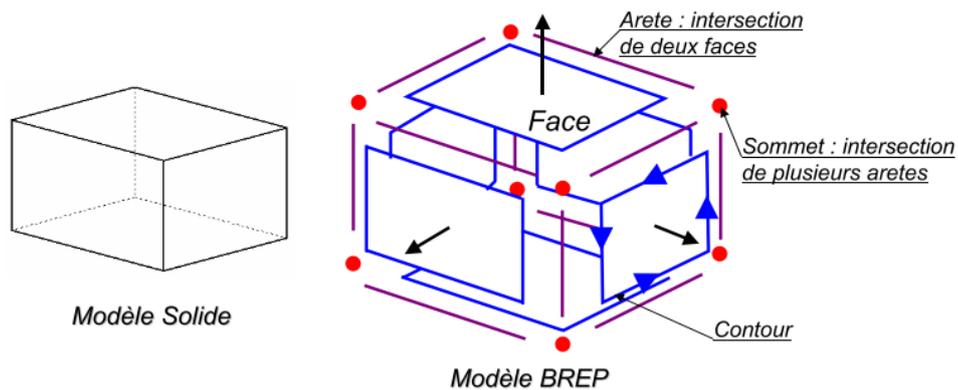


Fig.1. 6. Modèle BREP [12].

Pour qu'un modèle BRep soit valide, chaque face doit en outre vérifier un ensemble de propriétés garantissant la reconnaissance de l'objet comme un volume occupant une partie de l'espace ainsi que son homogénéité. En fait, le modèle BRep définit les frontières comme une juxtaposition de plusieurs faces orientées. Chacune d'entre elles (les faces) est formée d'une surface (décrire sa géométrie) et est délimitée par au moins un contour fermé et orienté. Chaque contour est composé d'un ensemble d'arêtes. Chaque arête comprend une courbe porteuse et limité par deux sommets.

### 5.3.3.3. La représentation CSG

Ce modèle est le plus utilisé en CAO (*Constructive Solid Geometry*) il combine des solides avec des opérations booléennes. On dispose d'un ensemble de volumes élémentaires, tels que des parallélépipèdes, des sphères, des cônes, des tores, etc. On paramètre leurs dimensions, (telles que largeur, longueur, hauteur, rayon, etc...) puis en utilisant les transformations géométriques (translations, rotations, etc.) et de l'ensemble d'opérateurs de composition (union, intersection et différence), on combine les objets de base afin de construire des objets spatiaux plus complexes. La représentation de l'intérieur d'un objet généré selon ce modèle est un graphe acyclique binaire appelé souvent arbre de construction CSG, où chaque nœud non-terminal est un opérateur de composition et où les feuilles sont des objets élémentaires. Ce modèle a évolué par la suite avec les logiciels de CAO vers l'arbre des caractéristiques qui contient l'historique de création de produit.

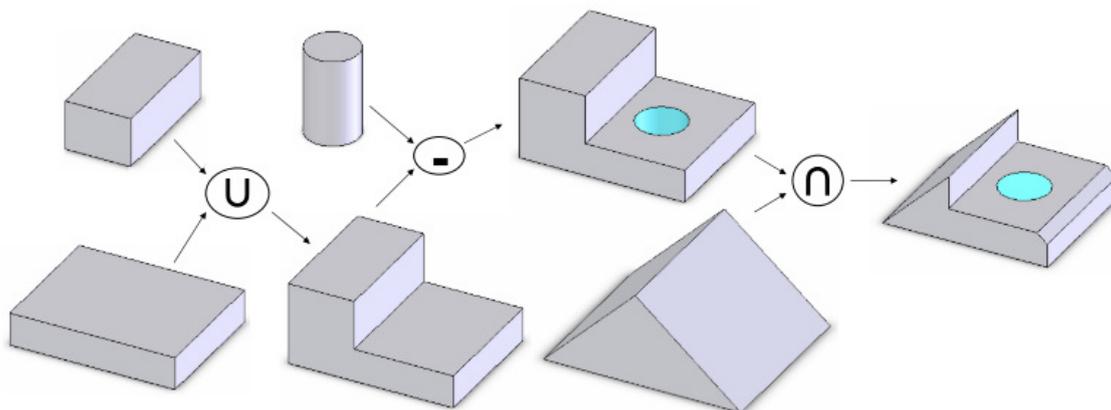


Fig.1. 7.La représentation CSG [12].

### 5.3.3.4. La représentation hybride

Le modèle hybride permet de combiner les avantages de la représentation du BRep et des CSG tout en réduisant leurs inconvénients respectifs. Les primitives de base utilisées dans ce type de modéleur sont soit celles du modèle CSG (cône, pyramide, cube), soit encore des solides qui sont définis par leurs frontières. À chaque intervention sur l'arbre de construction, les éléments de frontière sont réalisés. Ce modèle est le modèle le plus fréquemment utilisé aujourd'hui. Le modèle hybride nous permet : de construire un solide en toute simplicité, d'effectuer facilement des opérations complexes (calcul du volume d'un solide, appartenance d'un point au solide,

etc.). De plus, ce modèle nous donne également la possibilité de placer des contraintes géométriques et d'assemblage.

### **5.3.4 Synthèse sur les modèles de représentation numérique de produit**

La technique du fil de fer est utilisée par les logiciels de CAO tridimensionnels juste pour réaliser les lignes de construction et les contours d'un modèle 3D complexe. Cette technique est cependant peu utilisée dans les logiciels de CAO. Quant à la modélisation surfacique elle est encore très utilisée notamment pour les pièces de carrosserie et les pièces qui possèdent des faces de formes complexes. Les logiciels disponibles actuellement (Catia, SolidWorks, TopSolid...) comportent des modules surfaciques qui peuvent être utilisés pour générer des surfaces de plusieurs façons (surface passant par des courbes imposées et comportant des critères de tangence, surface approchant au mieux un nuage de points...). Cela permet de contrôler et de définir d'une façon plus précise le modèle produit.

La représentation solide permet de modéliser les pièces volumiques. La modélisation solide comporte trois principaux types de représentation (BREP, CSG et hybride), présentant tous des avantages et des inconvénients. Il est à noter que les modèles de représentation solide intègrent dans leurs techniques de création de solides les opérations classiques d'extrusion, de balayage, de révolution, de lissage, etc. rencontré dans les logiciels de CAO d'aujourd'hui. La modélisation solide présente des avantages par rapport aux autres techniques de modélisation à savoir l'accès aux propriétés massiques du solide constitué (centre de gravité, matrice d'inertie et position des axes d'inertie, volume et masse) et aussi aux propriétés topologiques et géométriques. L'extraction de ces données-là est également plus facile.

## **5.4 Les différents modes de conception**

### **5.4.1 Par assemblage, ou Bottom-Up**

Il s'agit de créer chaque pièce en dehors d'un contexte d'assemblage puis les assembler pour constituer une maquette fonctionnelle. C'est un mode séquentiel où la définition des pièces devance la construction du mécanisme.

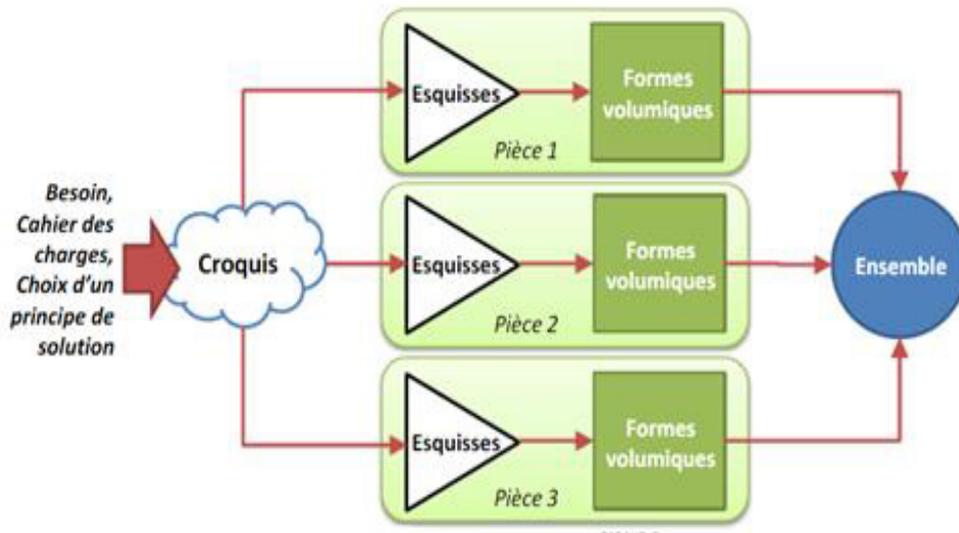


Fig.1. 8. Mode de conception en CAO « Bottom-Up » [19].

#### 5.4.2 Dans l'assemblage, ou Latéral

Ce mode se repose sur l'alternance entre les phases de conception de pièces et les phases de construction de l'assemblage. C'est un mode itératif s'appuyant sur la construction d'une première pièce qui est souvent la pièce la plus représentative et « au cœur » de l'assemblage ou la pièce de base de l'assemblage.

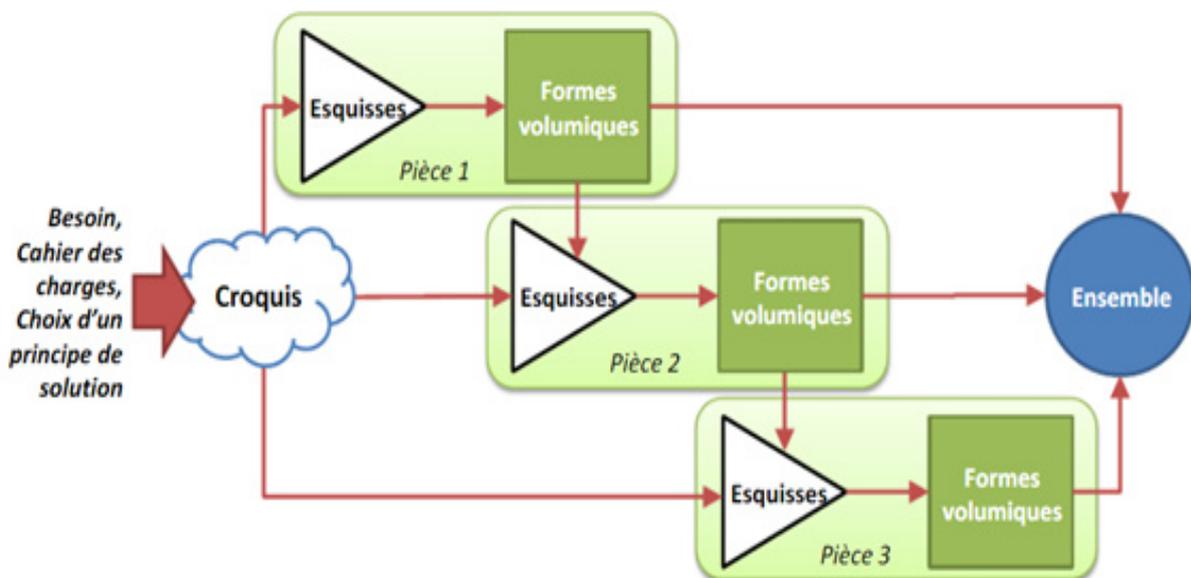


Fig.1. 9. Mode de conception en CAO « Latéral » [19].

### 5.4.3 Piloté, ou Top-Down

Il s'agit de travailler directement à partir d'un « guide fonctionnel » qui peut être un schéma, une esquisse particulière, dite pilotante, un squelette de mécanisme, etc., sur laquelle pour s'appuyer la définition des formes et dimensions des pièces constitutives.

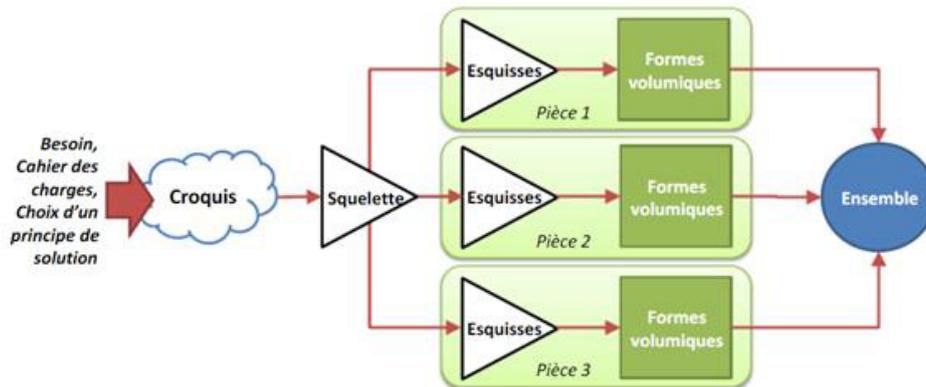


Fig.1. 10. Mode de conception en CAO « Top-Down » [19].

### 5.4.4 Utilisations et limitation des différents modes

La méthode « Bottom-up » est une méthode de conception robuste et performante car il n'y a aucun lien entre documents. Elle est aussi simple à comprendre et à mettre en place car elle est très basique. C'est la méthode de travail la plus utilisée par les logiciels de conception assistée par ordinateur CAO généralistes. Cette méthode de conception ascendante possède néanmoins des inconvénients. Elle nécessite plus de temps et est plus fastidieuse, car le dessinateur doit « réfléchir » ses pièces pour garantir le montage des unes par rapport aux autres (report de côtes, de formes ...). Mais surtout, le risque d'erreur est important surtout en cas de modification / évolution du modèle. *En effet, des contrôles ou des modifications doivent être effectués sur chacune des pièces concernées par une évolution.* La conception « Bottom-up » est pertinente dans le cas des :

- Assemblages mécaniques basiques
- Assemblages standards non paramétriques
- Assemblages où les pièces n'ont pas de contraintes fortes les unes par rapport aux autres
- Assemblages où les modifications vont être limitées

Tandis que la méthode « Top-Down » est plus facile à utiliser et aide spécialement à faire gagner du temps. Elle est idéale lorsqu'il s'agit d'étudier très rapidement un nouveau concept, optimiser un mécanisme en testant de nombreuses variations dimensionnelles, travailler dans le cadre de l'ingénierie simultanée.

Mais lorsqu'il s'agit de réutiliser les mêmes composants dans plusieurs mécanismes, il sera impossible. Aussi le concepteur se trouve souvent devant la nécessité de poser des questions et de travailler en groupe (impossible d'utiliser le logiciel dans son coin sans se poser de question). Le principal inconvénient c'est la difficulté de créer une gamme de produits fortement configurables.

On peut donc conclure que les différentes méthodes de conception présentent des limitations importantes en présence des changements d'ingénierie (quel que soit le contexte de la modification).

## 5.5 Changement d'ingénierie (EC)

Le changement d'ingénierie est devenu de plus en plus important à la fois pour l'industrie et pour le monde académique en tant que domaine de recherche et de développement actif [20]. Il s'avère indispensable de définir convenablement la notion des changements d'ingénierie (CI), ou engineering changes en anglais (EC), avant de commencer à citer les différentes méthodes de gestion existant ainsi que les différentes autres notions qui en découlent. Les trois définitions de Wright et al.(1997)[21], Terwiesch et al.(1999)[22] et Huang et al. ( 2001)[23] sont celles qui sont les plus citées dans la littérature. En se basant sur ces définitions, d'autres définitions plus complètes sont proposées et le tableau ci-dessous (Tab.1. 1) montre les définitions du terme Changement d'ingénierie les plus pertinentes qui existent dans la littérature.

Tab.1. 1. Les définitions d'EC existantes dans la littérature.

---

<b>Source</b>	<b>Définition</b>
Wright et al.(1997)[21]	Un changement d'ingénierie est une modification apportée à un composant d'un produit, lorsque ce produit est entré en production.

---

---

Huang et al.(1999)[24]	Les modifications d'ingénierie sont les changements et les modifications au niveau des formes, ajustements, matériaux, dimensions, fonctionnalités, etc. d'un produit ou d'un composant.
Terwiesch et al.(1999)[22]	Les ordres de changement d'ingénierie sont les modifications apportées à des pièces, des dessins ou des logiciels qui ont déjà été lancés.
Huang et al. (2003)[25]	Les changements d'ingénierie sont des changements et/ou des modifications dans les ajustements, les fonctions, les matériaux, les dimensions, etc. d'un produit et de ses composants après la validation de la conception.
Fricke et al. (2005)[26]	L'expression "changements" (...) englobe toutes sortes de changements, qu'il s'agisse de changements de besoins, d'exigences, de spécifications, de composants déjà construits, de processus, de coûts, de calendrier, etc.
Tavčar et al. (2005)[27]	Une modification technique est une modification d'un produit après son entrée en production.
Jarratt et al.(2011)[20]	Un changement d'ingénierie est une modification apportée à des pièces, des dessins ou des logiciels qui ont déjà été lancés au cours du processus de conception du produit. Le changement peut être de n'importe quelle taille ou de n'importe quel type ; le changement peut impliquer n'importe quel nombre de personnes et prendre n'importe quelle longueur de temps.
Siddiqi et al.(2011)[28]	Un changement de conception est considéré comme un processus de révision d'une maquette qui a déjà été considérée comme achevée ou finalisée.

---

Vianello et al.(2012)[29]	Un changement d'ingénierie est une modification apportée à un composant d'un produit, après que sa conception originale est terminée.
Hamraz et al. (2013)[30]	Les modifications techniques sont des changements et/ou des modifications apportées à : une structure validée (ajustements, formes et dimensions, surfaces, matériaux, etc.), tenue (stabilité, résistance, corrosion, etc.), fonctionnalité (vitesse, performance, efficacité, etc.) ou relations entre fonctionnalité et comportement (principes de conception), ou bien comportement et structure (lois physiques).

---

Il faut souligner que dans toute la littérature le terme « changement d'ingénierie » n'est pas couramment employé et que l'on le remplace souvent par le terme « ordre de changement d'ingénierie », ou engineering change order (ECO). Cette interchangeabilité se justifie par la similitude des deux termes. En pratique, le changement d'ingénierie « CI » induira un ordre de changement d'ingénierie « OCI » ; les intervenants parlent généralement d'un CI lorsqu'une OCI sera mis à exécution. On trouve aussi le terme « changement d'ingénierie de design » ou engineering design change (EDC), ou bien « changement de design » tout court parce que généralement les changements d'ingénieries touchent le design du produit[30].

Cependant, le changement technique est considéré à la fois une opportunité et un obstacle pour les entreprises [20]. Wright et al.[21] ont admis que les modifications techniques sont considérées comme « un mal, imposé sur le fabricant par les ingénieurs de conception qui ont probablement commis une erreur ». En revanche, ces changements peuvent en fait être le principal moyen pour une entreprise de maintenir ou d'accroître son marché, il est donc très rare de trouver une entreprise qui ne recourt pas à la stratégie des innovations incrémentales. En effet, les changements techniques peuvent survenir au cours des différentes étapes du cycle de vie du produit :

- Lors de la conception d'un tout premier produit.
- Pendant la fabrication
- En cours de fonctionnement.

- En vue de la conception des prochaines générations.

Mais, fondamentalement, il n'y a que deux raisons pour modifier un produit : (1) pour éliminer les erreurs ou le faire fonctionner correctement ou (2) pour l'améliorer ou l'adapter [20]. Or, ces modifications apportées aux produits d'ingénierie peuvent être classées selon Eckert et al.[31] en changements initiés ou émergents ou les changements initiés sont les changements qui surgissent de l'extérieur du produit. Les améliorations, perfectionnements ou adaptations d'un produit peuvent prendre différentes formes. Par exemple, un changement peut être initié pour réduire le coût du produit ou pour rendre le produit conforme aux normes et lois des territoires dans lesquels ils souhaitent vendre. Les changements émergents, eux, découlent des propriétés du produit lui-même, par exemple des erreurs qui apparaissent, et le produit doit donc être retiré pour qu'il soit amélioré et/ou corrigé. Il existe donc de nombreuses façons de classer le changement : selon le but ou l'origine, comme nous l'avons vu précédemment, ou selon l'urgence du changement ou sa survenue dans le cycle de vie du produit. DiPrima et al. [32] classe les changements techniques selon le moment où ils doivent être mis en œuvre. Il considère les modifications qui doivent être effectuées immédiatement et qui sont généralement liées à la sécurité ou à des défauts comme « *Immédiat* ». Les modifications qui doivent être prises dès que possible, mais pour lesquelles il y a une certaine souplesse quant au moment de leur exécution sont dites « *Indispensable* ». Enfin, celles qui améliorent le produit, mais ne sont pas essentielles sont considérées comme « *Commodité* ». Reidelbach et al. [33] propose les types de changements d'ingénierie (EC) en fonction du moment du changement dans le processus de développement du produit : Les EC précoces et à faible impact, les EC de mi-production (ont lieu après l'approbation des commandes d'achat et le début de la fabrication. Ils peuvent avoir un impact considérable sur la production) et les EC tardives et accélérés, ils peuvent être très dommageables.

## 5.6 Propagation des changements (ECP)

La propagation efficace du changement est une question économique fondamentale liée aux délais, à la qualité et à la compétitivité. En effet les changements d'ingénierie sont généralement inévitables et leur propagation peut entraîner une augmentation de la complexité de la conception tout en sachant que plus le système est complexe, plus il est difficile de le contrôler et de prévoir son comportement.

La Propagation des changements est définie par le processus par lequel un EC concernant des composants d'un produit entraîne un ou plusieurs EC supplémentaires affectant d'autres composants de ce produit, alors que ces modifications n'auraient pas été normalement nécessaires[34]. D'après Hamraz et al.[30], le phénomène de propagation est défini par le changement amorcé dans une pièce du système qui tend à avoir des effets en cascade, déclenchant ainsi des changements successifs dans les autres pièces. Selon Soufi et al.[35], la propagation des modifications est la détermination précise des entités impactées par cette modification, rapidement et efficacement.

La propagation du changement peut créer un effet boule de neige et, dans le pire des cas, une vague de changements pouvant affecter tout le système. D'autre part, l'impact qui en résulte peut se révéler très grave, car il peut entraîner à la fois une augmentation des coûts et un retard dans les délais de livraison.

### **5.7 Termes en relation avec le projet**

Il est important de bien définir les termes qui vont servir de support pour le travail.

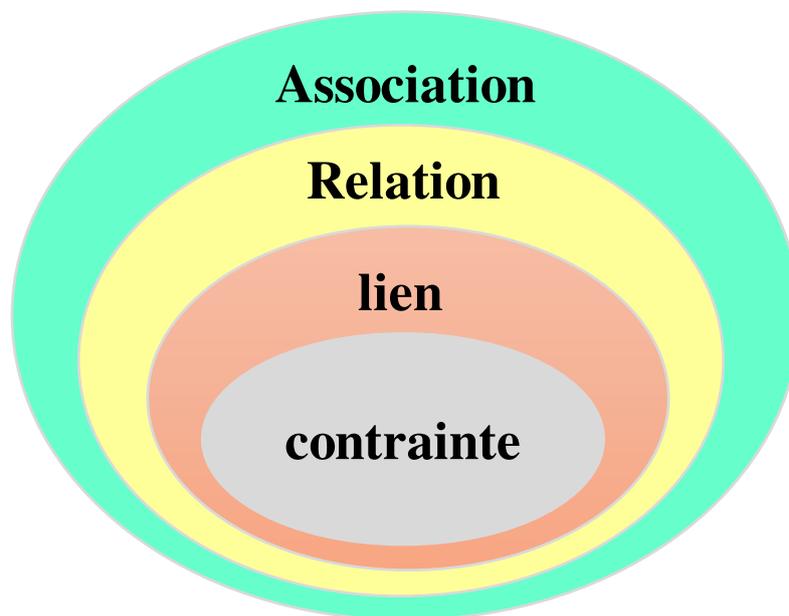


Fig.1. 11. Les différents niveaux de dépendance.

- Dépendance :

Une dépendance est un rapport qui fait qu'une chose dépend d'une autre. Selon Giguère et al.[36], un lien technologique est un lien exprimant une relation de dépendance entre des éléments d'information. Dans le contexte d'un assemblage, plusieurs auteurs ([37][38][39]) utilisent le terme relation de contact (Mating relation) pour décrire une dépendance entre deux composants d'un assemblage. Pour décrire plus précisément le type de dépendance, ils utilisent le terme contraint.

- Association :

L'association est un terme générique qui désigne la famille qui regroupe les termes : relation, contrainte et lien. Une association exprime une dépendance entre des éléments ou des entités.

- Relation :

Une relation établit une dépendance abstraite entre deux objets techniques (sans faire intervenir un savoir-faire précis ou formalisé). L'interprétation et l'utilisation d'une relation nécessite une intervention humaine. Par exemple, le document définissant l'outillage employé dans la fabrication d'une pièce est associé par une relation au document qui définit la pièce. Cependant, le savoir-faire requis pour définir l'outillage à partir de la définition de la pièce n'est pas formalisé dans cette relation. C'est propagé à l'outillage sans intervention humaine.

- Lien :

Le lien peut être défini comme un outil qui attache, unit, établit un rapport logique ou de dépendance. C'est la dépendance entre deux éléments qui font intervenir un savoir-faire formalisé pour réaliser une tâche donnée. Les liens peuvent être gérés automatiquement par logiciels, mais ils peuvent occasionnellement nécessiter une intervention humaine. Le savoir-faire associé à un lien est totalement maîtrisé et formalisé, contrairement à celui sous-entendu par une relation. Les liens au sein des assemblages sont des outils utilisés dans les systèmes CAO. Les liens d'assemblage sont établis entre les différents composants. Entre autres, ils permettent d'orienter et de positionner les composants les uns par rapport aux autres[38].

- **Contrainte :**

La contrainte décrit concrètement et précisément un type de dépendance entre deux entités. Une contrainte est indécomposable, il s'agit d'un savoir-faire de bas niveau. Le niveau d'abstraction d'une contrainte est plus faible que pour un lien. Un lien constitue une agrégation organisée de contraintes. Une contrainte constitue une condition à respecter pour valider le modèle[38]. Par exemple, une contrainte de perpendicularité peut être appliquée entre deux droites, il s'agit d'une opération à réaliser. Les différents niveaux de dépendances sont illustrés par la figure (Fig.1.8).

## **6. Conclusion**

Une fois bien définis, les EC sont plus facilement identifiables. La gestion des changements d'ingénierie doit se faire en ayant toutes les informations nécessaires. Tandis que ces informations dépendent énormément du modèle question de changement, du type de modélisation utilisé afin de concevoir ce modèle ainsi que du type de dépendance utilisé. Ce chapitre a permis, d'une part, de bien définir la motivation industrielle de la thèse ainsi que sa structure. Ainsi, le lecteur peut facilement trouver la réponse aux questions de recherche qui ont été formulés dans ce premier chapitre. D'autre part, les termes clés utilisés par la suite dans le reste du rapport ont été définis. Aussi, les différents types de modélisation ont été présentés dans le but de bien choisir le type du modèle et les entités sur lesquelles se base l'approche proposé dans le reste du travail. Subséquemment, le prochain chapitre tentera donc de recenser les avancements réalisés dans la gestion des changements d'ingénierie, de différents modèles de caractérisation de dépendance existants et des différents outils de comparaison des modèles CAO disponibles.

# **Chapitre 2 :**

## **État de l'art**

# Chapitre 2 : État de l'art

## 1. Introduction

Au cours de son cycle de vie, un produit est amené à évoluer tout en satisfaisant le besoin du client et en assurant le meilleur rapport qualité-prix. L'intervention de différents partenaires (clients, fournisseurs, ...) ajoute le problème de la gestion et le partage des données. L'exemple du scénario industriel qui en découle est celui d'une pièce qui est extraite de la maquette numérique, envoyée à un sous-traitant, modifiée par ce dernier, puis réinsérée dans la maquette initiale. On rencontre alors un problème au niveau du maintien des associations entre les composants. Ce sujet entre dans le cadre d'un projet qui consiste à développer un modèle de gestion des contraintes permettant la transposition de ces dernières de la pièce initiale vers celle modifiée et la propagation des changements au sein de la maquette numérique afin de garantir la cohérence entre les objets. Cette revue bibliographique représente des différents intérêts. Elle nous a permis de situer le thème de recherche par rapport à l'existant. C'est-à-dire situer ce travail de recherche par rapport à l'état de l'art des travaux de recherches achevés ou bien en cours de réalisation.

Dans ce chapitre on abordera quelques concepts de base et on présentera les différentes études déjà menées dans le domaine. Nous commençons tous d'abord par la présentation de la notion de la gestion des changements d'ingénieries, le processus des changements ainsi que la caractérisation de ces changements-là. Ensuite nous présentons les différentes méthodes de la gestion des EC et de leurs propagations existantes dans la littérature. Nous aborderons aussi la notion de la modélisation des dépendances. La deuxième partie de ce chapitre est consacré à l'étude des outils de comparaison des modèles CAO existantes dans la littérature et aussi utilisés dans les systèmes CAO commercialisés.

## 2. Gestion des changements d'ingénierie (ECM)

### 2.1 Product Lifecycle Management (PLM)

#### 2.1.1 Définition et historique

Le **PLM (Product Lifecycle Management)**, aussi en français appelé **GCVP (Gestion du Cycle de Vie du Produit)**, est un outil où sont gérées les données nécessaires permettant par la suite l'évolution du produit de A à Z, jusqu'à sa production. Comme son nom l'indique (gestion du cycle de vie des produits), il vise à gérer et faciliter le partage de l'ensemble des données de définition, de fabrication et de maintenance d'un produit données, tout au long de son cycle de vie, depuis les études préliminaires jusqu'à la fin de vie.

Au niveau du bureau d'études, on parlera plus précisément de **Product Data Management (PDM)** ou **Système de Gestion Données Techniques (SGDT)**, dont le champ d'application est plus restreint et se concentre uniquement sur les données techniques (modèles CAO, etc.). Le PLM est à la fois un l'outil qui permet aux sociétés de protéger leur capital intellectuel et un outil de collaboration obligatoire face à la mondialisation des marchés et à la concurrence importante.

Initialement, les premiers systèmes sont apparus suite à l'introduction de la conception assistée par ordinateur (CAO) au début des années 1980 [40]. En effet, le besoin de gérer les données issues du Bureau d'Études (BE) a rendu obsolète la classique "armoire à plans" renfermant les différentes versions des documents d'études réalisés. Les outils CAO génèrent des données plus complexes. La structure du produit se décompose sous la forme d'un ensemble de fichiers reliés entre eux avec divers liens et relations. Le besoin de gestion de ces données complexes a permis l'émergence de démarche d'intégration aux bases de données en fédérant ces données techniques autour d'un système centralisé, d'où l'apparition des premiers systèmes PLM. Le développement des systèmes PLM a été progressif et il est même encore en évolution. Selon Yildiz et al. [40], on peut distinguer 3 grandes phases (Fig.2. 1) :

1. Dans les années 1980, la nécessité de gérer les données techniques du bureau d'études a conduit à l'émergence des premiers SGDT (Système de Gestion de Données Techniques) ou PDM (Product Data Management). Toutefois, l'échange de données en utilisant les premières versions des systèmes SGDT est strictement réservé aux acteurs du bureau d'études de l'entreprise, alors que leur exploitation est limitée à la production.

2. Dans les années 1990, grâce à l'évolution des SGDT (en particulier à la venue et à l'évolution d'Internet), les différents services de l'entreprise (conception, achats, ingénierie, fabrication, logistique, marketing, vente, après-vente, etc.) sont capables, avec ce type d'outils, de collaborer. La productivité de l'entreprise a ainsi augmenté.

3. Les années 2000 ont marqué un véritable tournant dans le développement du SGDT. Ainsi les données techniques sont échangées actuellement entre l'entreprise et ses divers collaborateurs (fournisseur, donneur d'ordre, partenaire, sous-traitant...). Ce partage de données internes et externes fait du SGDT le PLM que nous partageons aujourd'hui.

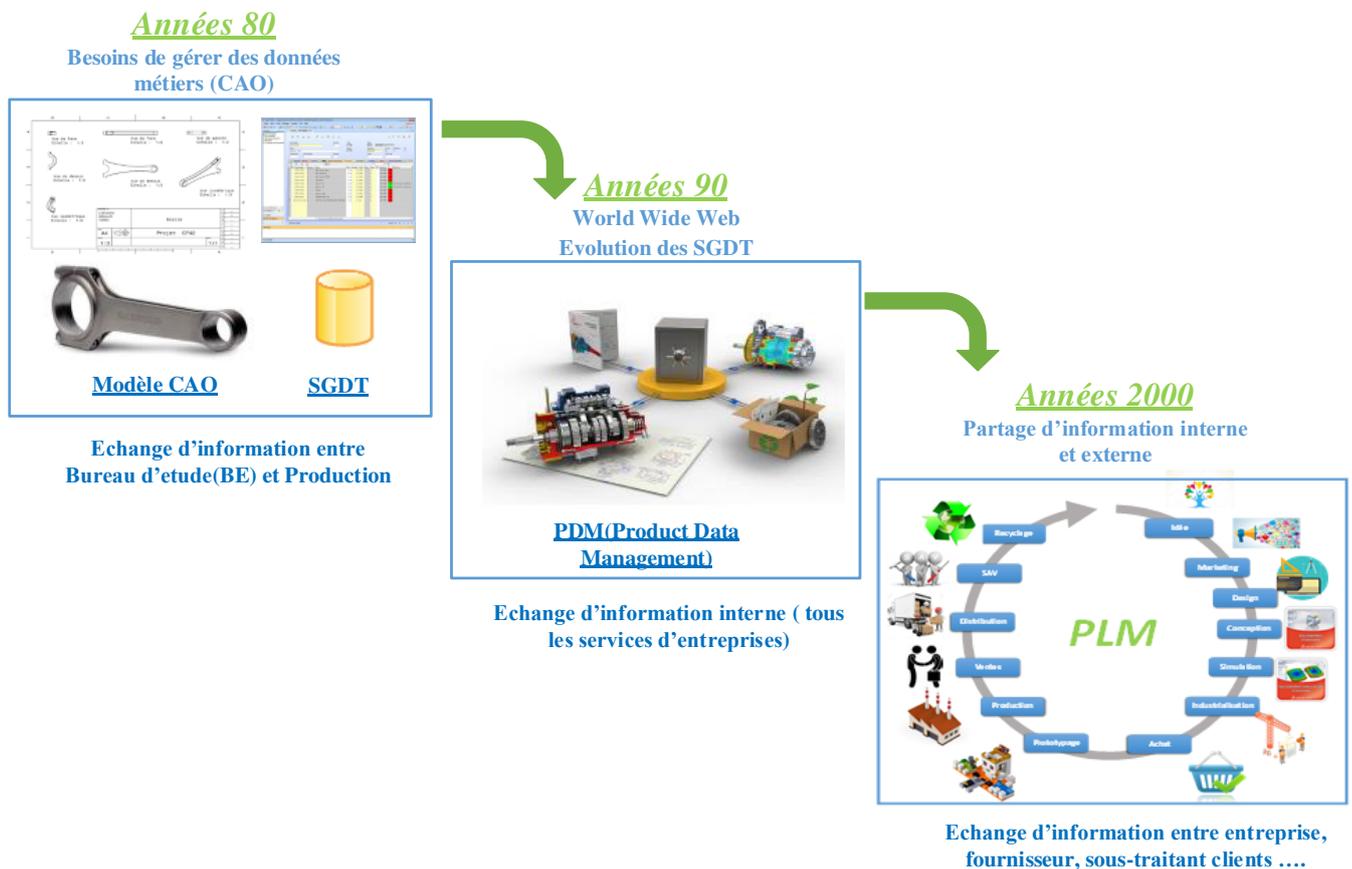


Fig.2. 1. Les 3 grandes phases de l'origine des systèmes PLM.

### 2.1.2 Limitations

Les systèmes PLM gèrent les associations entre fichiers, à partir d'informations externes aux fichiers CAO de nature essentiellement administratives, telles que la date de la dernière modification, l'auteur de la modification, l'état de maturité du fichier (TEC, Approbation en attente, Validé, etc.). En fait, les systèmes PLM gèrent les fichiers comme des boîtes noires. Dans ce contexte, ils ne peuvent prendre en compte que l'impact d'un changement sur les entités directement concernées, et donc sur les fichiers affectés par la modification. Une modification dans un fichier conduit le système PLM, par "effet domino", à déduire que ce changement peut avoir une incidence sur tous les autres fichiers. Par exemple, une modification d'une Feature d'un composant modifie le composant, qui à son tour modifie le composant qui l'utilise ou bien

en relation avec celui-ci (fils, parents, sous-assemblage...), etc. Mais le système PLM montre des limitations importantes pour les gérer. En effet, les associations gérées par le système PLM sont moins nombreuses que celles gérées par les systèmes CAO. Ce manque d'associativité dans les systèmes PLM a été mis en évidence par Tremblay et al.[41]. Derrière le document CAO, il existe de très nombreuses données et informations qui sont aussi liées. Les systèmes de CAO gèrent les associations de plus bas niveau, c'est-à-dire entre des entités géométriques ou des paramètres numériques élémentaires. Cependant, en raison de leur très faible contenu sémantique, leur potentiel d'exploitation est très limité en ce qui concerne la propagation du changement à l'intérieur des systèmes CAO.

## **2.2 La conception collaborative**

### **2.2.1 Définition**

Le Processus de Conception collaborative ou Intégré peut être défini comme une approche de conception interdisciplinaire mettant l'accent sur la collaboration. Tous les acteurs impliqués dans le projet sont réunis lors des étapes de conception des plans et des spécifications afin de développer des solutions optimales pour chacune des disciplines [42]. Brissaud et al.[43] affirment aussi que la conception intégrée est une méthodologie basée sur un environnement d'ingénierie simultanée. Chaque concepteur fait partie intégrante d'une équipe qui est composée de plusieurs compétences différentes.

### **2.2.2 Les changements d'ingénierie dans un environnement de conception collaborative**

La principale difficulté associée à un processus de conception collaborative est la compréhension des données du produit échangées pendant la conception[44]. Par ailleurs, la conception collaborative et le management des changements sont fortement liés. D'une part, plusieurs auteurs proposent des modèles de gestion des données pour soutenir la conception collaborative. D'autre part, le fait de travailler dans un milieu collaboratif de conception facilite énormément la tâche de gestion des données et des changements d'ingénierie.

Le travail de Kocar et al.[45] entre dans ce cadre, il propose une nouvelle approche pour traiter les changements dans un environnement de conception collaborative virtuelle. Cet environnement, appelé Active Distributed Virtual Change Environment (ADVCE) offre une solution de gestion des changements techniques en fusionnant les données graphiques et paramétriques dans une plate-forme virtuelle, ce qui améliore la compréhension des utilisateurs

et réduit ainsi le temps nécessaire à la modification. ADVICE fournit une aide intelligente aux utilisateurs pour prévoir et bien gérer les modifications techniques à déclencher en raison d'un changement spécifique. Aussi, Reddi et al.[46] ont développé des modèles dynamiques de système afin d'étudier les interrelations complexes entre les différents membres d'une chaîne de conception et de production collaborative pour obtenir un processus de gestion des modifications techniques efficace et efficient.

Dans le même cadre Demoly et al. [47] ont proposé un système intitulé PASODE (Proactive ASsembly-Oriented DEsign) en se basant sur les travaux de Bradley et al.[48] et Brown et al.[49]. C'est une approche de la conception simultanée qui diffère de l'approche traditionnelle. Cette approche permet essentiellement de prendre en compte simultanément les deux ensembles d'exigences de conception et de planification, ce qui permet d'éliminer les défauts de conception dus à l'absence de prise en compte simultanée des exigences relatives au cycle de vie (Ils intègrent l'ingénierie des processus d'assemblage dans le processus de conception des produits).

De même pour faciliter la tâche de partage de données dans le cadre de l'ingénierie simultanée des produits tout en concevant la cohérence, certains adoptent des stratégies de modélisation à base de squelettes. Le travail de Demoly et al.[50] rentre dans ce contexte. L'objectif de l'approche proposée est de calculer et de définir un modèle de squelette géométrique basé sur les informations relationnelles du produit et la séquence d'assemblage définie au préalable. L'approche proposée est appelée SKELeton geometry-based Assembly Context Definition (SKL-ACD). Le travail de He et al. [51] aussi présente une méthode de modélisation de la conception paramétrée basée sur le modèle de squelette qui peut être utilisée pour résoudre le problème de la conception intégrée.

### **2.3 Le management des changements d'ingénierie**

La gestion des changements d'ingénierie (GCI) est le domaine scientifique dans lequel les chercheurs prédisent les conséquences de la propagation du changement pour réduire le nombre de changements, les détecter de manière précoce lorsqu'ils surviennent, les traiter et les mettre en œuvre efficacement ou les prévoir pour les anticiper si possible à un stade ultérieur.

Les différentes définitions existant dans la littérature sont présentées en détail dans le premier chapitre tandis que la définition la plus générale (que nous adoptons par la suite) est celle proposée par Jarratt et al.[52] " Car c'est la définition la plus générale et qui englobe tous les

types de modification possibles. Un changement d'ingénierie est une modification apportée à des pièces, des dessins ou des logiciels qui ont déjà été lancés au cours du processus de conception du produit. Le changement peut être de n'importe quelle taille ou de n'importe quel type ; le changement peut impliquer n'importe quel nombre de personnes et prendre n'importe quelle longueur de temps".

L'idée générale de différentes définitions existantes réside dans le fait qu'un changement d'ingénierie est une modification d'un paramètre d'un produit lors de sa phase de conception ou de ré-conception. Un changement d'ingénierie a son propre « cycle de vie » ou processus.

### **2.3.1 Le processus des changements d'ingénierie**

Le processus ECM peut être défini comme étant le processus permettant de réaliser des modifications sur un produit d'une manière planifiée et systématique[53]. Dans la littérature, on trouve souvent des termes qui décrivent le processus des changements d'ingénierie et qui précèdent la réalisation du changement. Le processus de changement se compose principalement de quatre phases : la demande, l'instruction, l'exécution et l'application[8], [54].

Celui qui veut réaliser une modification doit tout d'abord formaliser une demande de modification ou bien ECR (Engineering Change Request). Une fois que cette demande est acceptée on peut dire que la demande est instruite et un ordre de modification ou ECO (Engineering Change Order) est émis. Par la suite, la modification demandée va être exécutée et appliquée. D'autres auteurs ont proposé des processus plus détaillés et d'autres qui sont moins détaillés. Jarrat et al.[20] ont proposé un processus générique de gestion des demandes de changement comme le montre la figure (Fig.2. 2) qui peut être décomposé en six grandes étapes : (1) Le processus commence bien évidemment par une demande de modification (ECR). (2) Il faut ensuite identifier les solutions possibles à la demande de modification. (3) L'impact ou le risque de la mise en œuvre de chaque solution doit alors être évalué. (4) Une fois qu'une solution particulière a été choisie, elle doit être approuvée. (5) La mise en œuvre de la modification technique peut être immédiate ou progressive. (6) Enfin, après un certain temps, le changement doit être réexaminé pour voir s'il a permis d'atteindre les objectifs initialement prévus et quels enseignements peuvent être tirés pour le processus de changement futur.

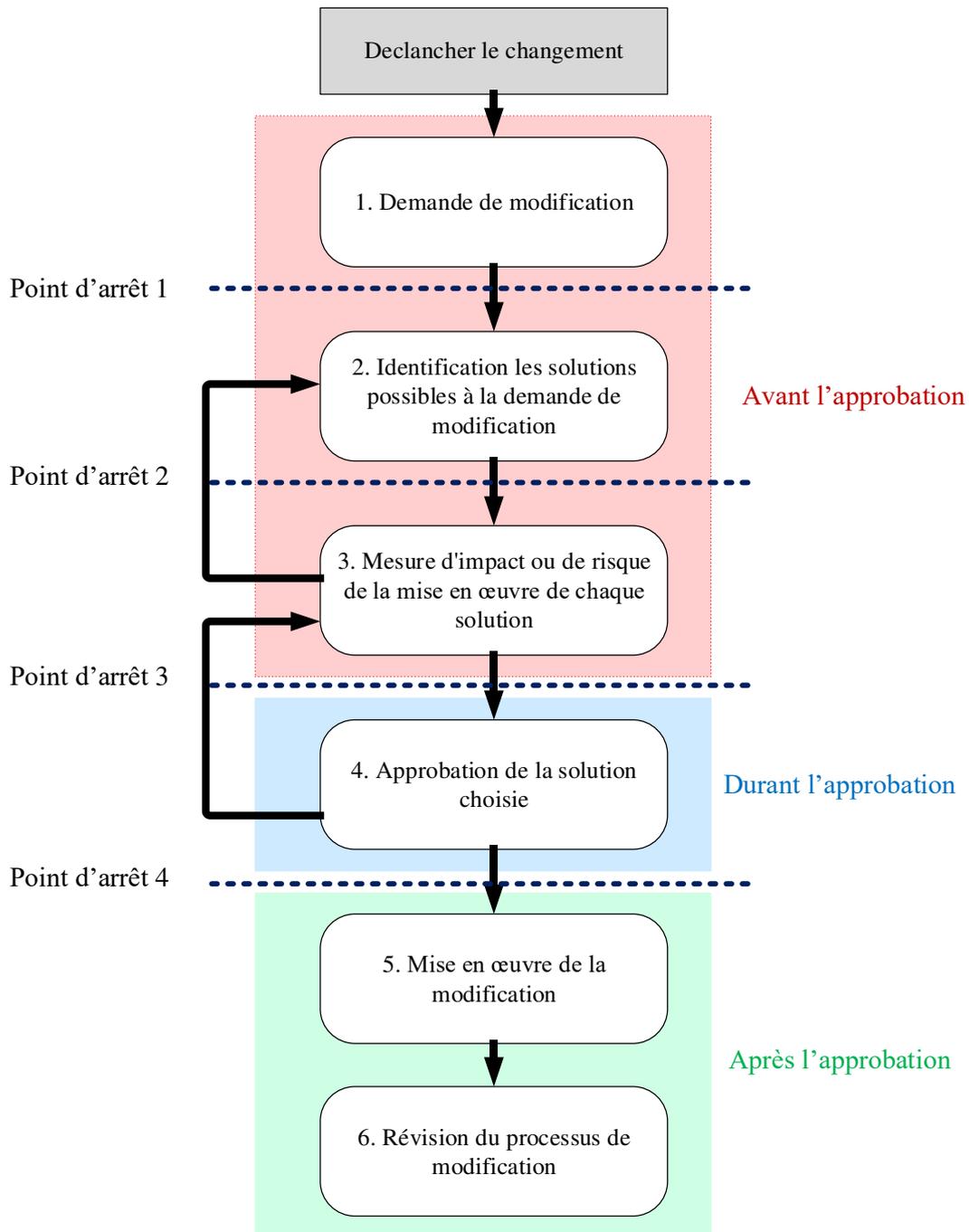


Fig.2. 2.Le processus de gestion des modifications proposées par Jarratt et al.[20].

Le processus de changement utilisé par Rowell et al.(2009) [55] est semblable, par son style, au processus de changement proposé par Jarratt et al.[20]. Il comprend un certain nombre d'étapes, qui peuvent être décomposées principalement en cinq grandes étapes. La motivation d'un changement est le point de départ du processus. Il s'agit d'un facteur stimulant qui motive suffisamment un acteur à enquêter sur une entité ou un système donné. Si le facteur stimulant

s'avère important, et si aucune solution n'est trouvée, elle passe à l'étape de demande de modification (ECR). Durant cette phase (ECR), une analyse d'impact complète de la modification en question sera faite. Cela inclut : la catégorie de changement ; la détermination des services affectés et des attributs affectés ; préciser le motif du changement ; définir la solution proposée et identifier les coûts et l'impact sur les délais. S'il s'avère que le changement est indispensable, il doit être approuvé par les gestionnaires de projet concernés afin de valider tous les changements. Enfin, une fois que l'ECR a été approuvé, il est envoyé aux intervenants concernés pour que les modifications soient effectuées. On parle à ce stade de notification d'un changement d'ingénierie ou ECN. Il s'agit d'un processus typique de gestion des changements de type réactif, car l'impact d'un changement potentiel est évalué une fois qu'un problème connu a été rencontré.

Dans ce même cadre de processus de gestion de modification Ouertani et al. [56], après avoir fait une étude bibliographique approfondie, ont distingué trois phases principales afin de réaliser un EC : Demande de modification ou ordre de modification (ECR). Les demandes de modification sont recueillies avec toutes les informations nécessaires pour leur traitement. L'étape suivante sera le traitement de la modification (ECT) ou une étude de faisabilité est effectuée afin d'évaluer la rentabilité et la faisabilité technique de la modification demandée. L'analyse des impacts de l'EC permet de détecter les conséquences sur le produit et la société. Ensuite, une décision est prise quant à la continuation du processus (annulation ou mise en œuvre du changement). La dernière étape est la réalisation de la solution. La solution est implémentée dans la réalité, les documents sont mis à jour et les acteurs concernés sont informés.

### **2.3.2 Caractérisation des changements**

L'évaluation de l'impact de la propagation du changement n'est possible que si la caractérisation des changements, quantitativement et qualitativement est faite. La caractérisation qualitative consiste à donner un aperçu des variations possibles d'un paramètre sans qu'il soit nécessaire d'en déterminer les valeurs au préalable. On doit poser la question suivante : comment une modification d'un paramètre affecte les autres paramètres ? Autrement dit, quels sont les paramètres affectés par cette modification ? Tandis que la caractérisation quantitative vise à déterminer les nouvelles valeurs des paramètres affectés.

La caractérisation qualitative a été présentée par Kusiak et al.[57] en se basant sur la logique floue et en déterminant le " sens " du changement (croissant, décroissant, nul ou inconnu) d'une variable qui est affecté par une autre. La même idée a été utilisée par Chua et al.[58] , Cohen et al [59] qui ont ajouté des valeurs qualitatives ("L-Low", "M-Medium" et "H-High") à partir d'une expertise empirique.

En ce qui concerne les méthodes utilisant la dépendance quantitative, elles visent à déterminer les nouvelles valeurs des variables engendrées par une modification déjà réalisée ou bien qu'on désire réaliser. Cheng et al.[60] et Kusiak et al.[61] utilisent la dépendance quantitative pour calculer le taux de changement. Clarkson et al. [62] et Hamraz et al.[63] ont utilisé des variables aléatoires, appelée " likelihood ", pour modéliser la probabilité d'occurrence des changements. Ils ont également ajouté des valeurs déterministes des impacts pour incorporer un changement dans le nœud cible [63], [64], [65].

### **2.3.3 La gestion des changements d'ingénierie**

L'ECM est un processus complexe. La complexité de la mise en œuvre d'un EC peut être du fait que cela implique généralement une intervention de plus d'un service fonctionnel ou même d'une organisation entière. Pour faire face à cette complexité, les entreprises doivent adopter une procédure bien définie pour gérer un EC, de la proposition de l'EC ou la demande de modification à sa mise en œuvre finale. Afin d'obtenir une gestion optimale des EC, il faut établir un équilibre entre l'efficacité et l'efficacités de la procédure de gestion des modifications [66].

La gestion des changements d'ingénierie est définie comme le fait d'identifier et d'évaluer l'impact des changements planifiés sur la définition d'une pièce et des différents modèles qui en découlent, tels que les plans de fabrications, les programmes CN ou les modèles d'analyse et de simulation [67]. D'après Fricke et al. [10], l'ECM s'occupe de la prévention, de la détection précoce, de la sélection efficace, de la mise en œuvre efficace et de la capacité continue d'apprendre des changements.

Plusieurs méthodes et approches ont été élaborées dans la littérature sur la gestion des changements d'ingénierie. Hamraz et al. [30] ont classé ces méthodes en trois catégories : Méthodes traditionnelles (ou méthodes qui se concentrent principalement sur une seule couche de produit), Méthodes multi-compagnies et Méthodes multicouches.

### 2.3.3.1 Méthodes traditionnelles

La technique C-FAR (Change Favorable Representation) proposée par Cohen et al.[59] définit les Artefacts comme des entités (des composants) qui sont représentées par des vecteurs et les décrit par leurs caractéristiques (diamètre, longueur, matériau...) représentées par des éléments des vecteurs. Des matrices C-FAR capturent les dépendances entre les attributs de deux entités comprenant les valeurs low, medium, et High (sur la base de l'évaluation d'experts du domaine). Par la suite, les matrices et les vecteurs le long des trajets de propagation des changements sélectionnés sont multipliés pour calculer les valeurs de liaison entre deux attributs. Cette technique a été utilisée par Cohen et al. Pour examiner plusieurs études de cas industriels (p. ex. un pare-chocs de voiture, un circuit imprimé et un moulage par injection). Cependant, la quantité des efforts requis (lors de la multiplication) dégrade la technique (des multiplications matricielles n'est pas toujours évidente) et ne la rend appropriée que pour des produits relativement simples ou peu complexes. Pour éviter un nombre ingérable de multiplications matricielles et vectorielles, il est nécessaire de prédéfinir des chemins de propagation des changements pertinents. Ainsi, la technique ne peut qu'évaluer la propagation des changements le long de ces trajectoires données. Cela ne peut pas être utilisé pour une analyse exhaustive en raison de l'immense quantité de multiplications matricielles et vectorielles requises.

L'outil Redesign IT proposé par Ollinger et al. (2004) [68] modélise un produit comme un graphe de dépendance directe construit en se basant sur les concepts suivants :

- Des grandeurs physiques pertinentes se référant à la fois aux propriétés physiques des composants du produit et aux descriptions des opérations du produit.
- Les contraintes sur les quantités décrivant les exigences de conception sur les quantités.
- Les relations causales entre les quantités décrivant comment une modification d'une quantité influençant les autres quantités.

Le modèle est utilisé pour générer des plans de modification possibles à partir desquels on peut effectuer des analyses de propagation causale. Comme le notent Ollinger et al.[68], les plans générés sont plutôt abstraits ; ils indiquent les quantités et la direction dans laquelle ces quantités doivent être modifiées pour atteindre un objectif de performance précis, mais ne précisent pas les valeurs numériques exactes des quantités. Cependant, l'outil aide le concepteur

à comprendre les conséquences possibles d'un remaniement en indiquant les éléments clés du produit qui seront touchés par un changement.

L'outil CPM (Cambridge Advanced Modeller) proposée par Clarkson et al. (2004) [62] est basé sur une méthode de prédiction numérique et probabiliste du changement fondée sur la théorie des réseaux complexes pondérés qui utilise un modèle de dépendances entre les paires de composants pour calculer et visualiser le risque global de propagation du changement (qui serait imposés aux autres composants lorsqu'une composante est modifiée). Le résultat du CPM est une matrice de la structure de conception (DSM : Design Structure Matrix), où les éléments de colonne indiquent les composants qui initient les changements, les éléments de ligne indiquent les composants qui reçoivent les changements, et les cellules entre deux composants donnés incluent les différentes valeurs du potentiel de risque. L'approche est structurée en quatre étapes, comme le montre la figure (Fig.2. 3), en utilisant l'exemple d'un sèche-cheveux pour la démonstration. La construction de ce réseau implique : (1) La décomposition du produit en sous-systèmes des composants, (2) Le développement des relations de dépendance entre les différents sous-systèmes (dans une matrice DSM binaire, où une croix "×" indique l'existence d'un lien de dépendance), (3) Deux autres DSMs, appelées matrices directes de probabilité "likelihood" et d'impact sont utilisées. Leurs valeurs sont données par des experts en se basant sur leur expérience et leurs connaissances du produit. (4) La matrice de probabilité directe représente la fréquence relative de la propagation d'un changement entre deux composantes. La valeur de ce risque direct est calculée ensuite en multipliant la valeur d'impact avec celle de probabilité.

Cette technique a été appliquée à plusieurs cas d'étude industriels avec des résultats intéressants, notamment : un hélicoptère (Clarkson et al.(2001) [69]), une vanne ferroviaire (Jarret et al. (2002) [70]), un moteur diesel (Jarret et al.(2004) [52]) et un injecteur (Keller (2007) [64]). Plus récemment, Ariyo et al.(2008) [71] ont amélioré l'approche en élaborant une méthode hiérarchique pour prédire les risques à de multiples niveaux de décomposition (c.-à-d. composants, systèmes et produits). Koh et al.(2012) [34] ont combiné cette méthode avec la maison de la qualité (House of Quality) pour évaluer différentes possibilités de modification en fonction des exigences des produits. Hamraz et al.(2012) [72] ont mis au point un algorithme basé sur une matrice qui permet d'exécuter et de calculer les résultats du modèle avec des tableurs et ont suggéré une technique permettant de tenir compte de multiples changements à la fois.

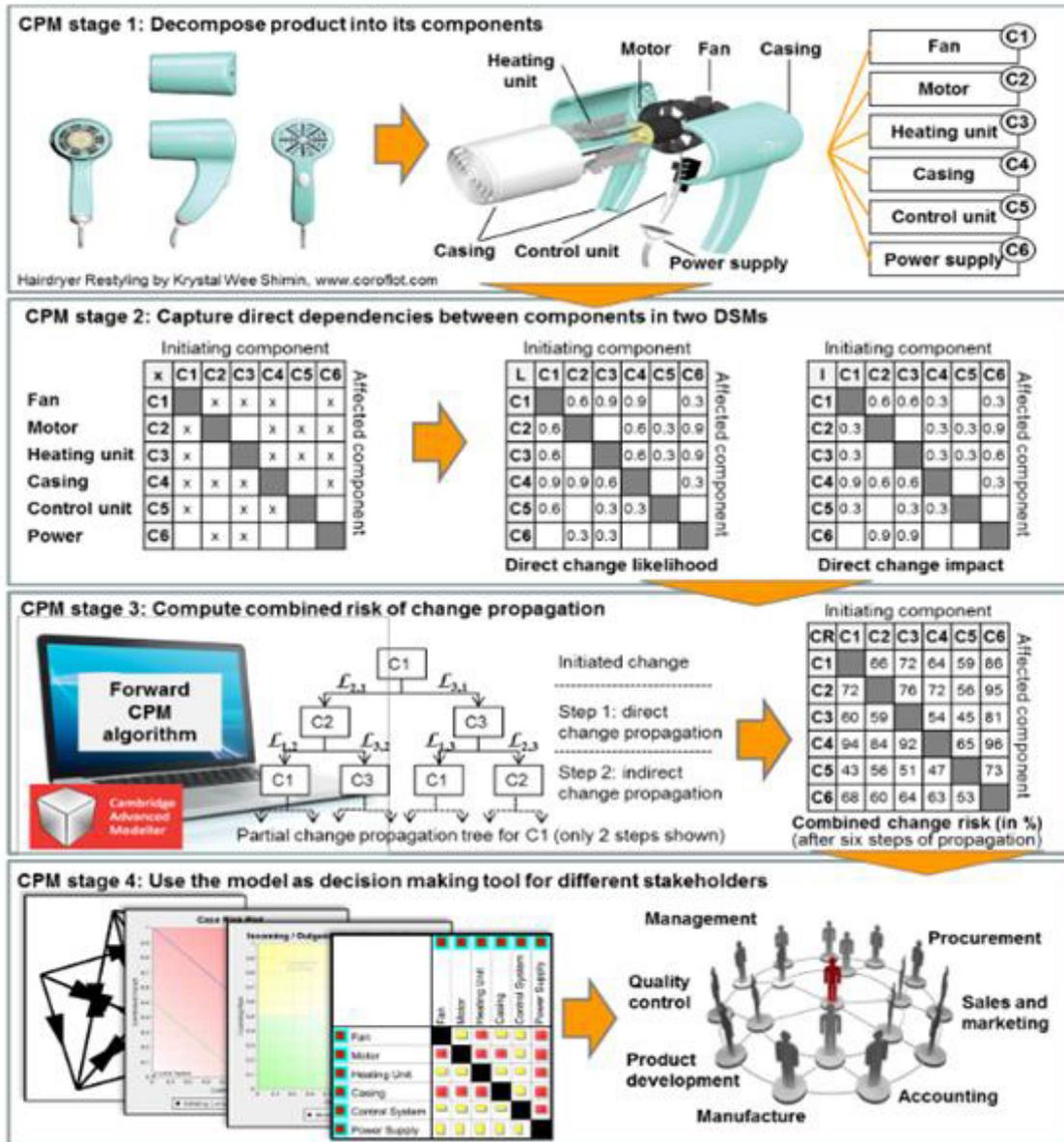


Fig.2. 3. L'approche du CPM proposé par Clarkson et al. [44].

Plusieurs autres méthodes utilisent le DSM basé sur l'approche du CPM et intègrent d'autres types d'informations pour que la modélisation de la propagation du changement soit plus déterministe. L'approche de Rutka et al. (2006) [65] est un bon exemple. Leur analyse de propagation des changements remplace la probabilité directe et les valeurs d'impact par des relations qui dépendent du type et du niveau de changement. L'idée est de décrire la propagation du changement cas par cas, en utilisant des spécifications de changement. Les dépendances entre les éléments initiateurs et cibles sont décrites dans un DSM et sont spécifiées plus en détail en définissant le type et le niveau de changement pour chaque élément. Cette prise en compte des types et des niveaux de changement affine l'information sur les dépendances entre les

composantes et permet une analyse plus précise et plus systématique que celle du CPM. Toutefois, comme les dépendances numériques probabilistes sont remplacées par des dépendances déterministes non numériques, cette méthode nécessite, d'une part, plus d'informations et, d'autre part, ne prend pas en charge l'analyse numérique de propagation des changements.

La méthode proposée par Reddi et al.(2009) [66] tient également compte de différents types de changements (Fig.2. 4). Leur modèle capte les dépendances jugées sur des niveaux discrets (faible, moyen et élevé) entre les attributs des composantes pour différents types de changements (matériau, forme et géométrie). La propagation du changement est effectuée en fonction du composant déclencheur et du type de changement. Les changements indirects sont pris en compte lors de commandes multiples. Par conséquent, l'outil génère une liste des composantes affectées ainsi que les types de changements qu'elles sont censées subir. Cependant, le niveau de détail du modèle empêche son application à des systèmes complexes.

Une autre approche qui est très semblable à celle du CPM a été poussée par Cheng et al. (2012) [60]. Ils ont proposé une analyse de l'impact du changement basée sur la théorie des réseaux pondérés et trois indices de changeabilité qui en découlent. Un produit est modélisé comme un réseau pondéré de pièces, sous-ensembles ou sous-systèmes. Alors que dans le CPM, les relations de dépendance entre ces éléments sont saisies par les valeurs de probabilité de changement et d'impact, cette approche utilise des degrés de couplage. Comme dans le premier cas, ces valeurs sont tirées d'entrevues avec des experts du domaine et saisies dans le cadre d'un DSM par composante. Contrairement à l'algorithme du CPM qui prend en compte toutes les trajectoires de changement possibles en pondérant leurs impacts avec leurs probabilités respectives puis les résumer, les indices proposés ne considèrent que les trajectoires ayant l'impact maximal entre deux composants. Cependant, comme l'a montré une étude de cas réelle d'un ventilateur Roots (Cheng et al. (2011)[60]), ces indices pourraient être utiles pour évaluer l'impact du changement.

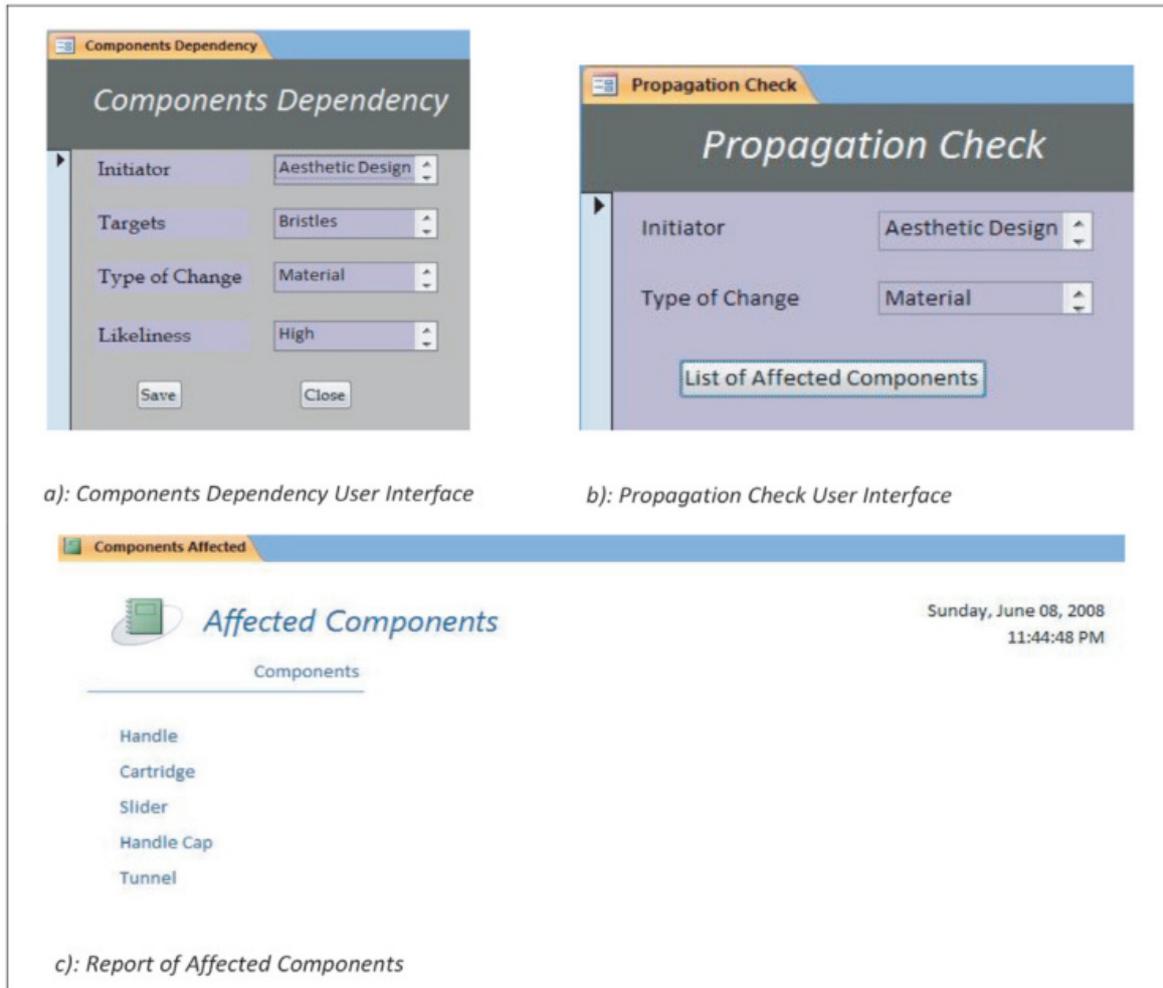


Fig.2. 4. Implémentation de l'outil proposé par Reddi et al. (2009) [66].

Il est important de noter les différences entre les trois approches traditionnelles de modélisation de la propagation du changement. La CPM se concentre sur les composants et s'appuie sur les relations structurelles entre eux comme indices de la propagation du changement. Il s'agit d'une méthode probabiliste qui montre le risque imposé aux autres composants si un composant change. Redesign IT ignore les composants et se concentre sur les grandeurs physiques (p. ex. la température de l'arbre) qui décrivent les comportements des systèmes. Il soutient le raisonnement causal au sujet de la propagation des changements entre ces grandeurs physiques. C-FAR examine les attributs des éléments clés du produit (p. ex. le type et le volume d'un liquide) et la façon dont ils sont liés aux attributs des autres éléments. Il prend en charge la propagation des changements au niveau des attributs. Cependant, les trois méthodes se concentrent sur la propagation du changement principalement à l'intérieur d'une couche de produit et ne tiennent pas compte des autres couches d'information de conception.

### 2.3.3.2 Méthodes multi-compagnies

Chen et al. (2002) [73] ont proposé une approche distribuée d'ECM pour l'ingénierie simultanée afin de gérer les processus, les systèmes et les informations des ECs au sein d'une entreprise individuelle et entre plusieurs entreprises regroupées. La méthodologie comprend un modèle de cycle de vie, un cadre de gestion hiérarchique et distribué et un modèle de référence pour l'ECM. Le modèle correspondant est développé à l'aide un langage de modélisation unifié. Le système de gestion se compose de modules de gestion de projets et de systèmes, de gestion de données et de service de modification technique. Ces modules travaillent ensemble de manière intégrée pour faciliter la communication, la coordination, le contrôle et l'intégration d'un système de gestion des modifications techniques. Toutefois, le système ne tient pas compte de la propagation des changements.

Rouibah et al. (2003) [53] ont présenté une approche basée sur des paramètres pour soutenir les processus d'ingénierie simultanés multi-compagnies (l'effort de conception du produit est simultané et fait intervenir plusieurs compagnies) et l'ECM. Leur approche vise à fournir rapidement et en tout début de processus, aux parties concernées, un aperçu de l'impact des changements proposés. La saisie et la gestion des paramètres se concentrent sur les paramètres du système et de l'interface, qui sont susceptibles d'intéresser les différentes parties prenantes à l'effort de conception. Les paramètres sont appelés " les variables d'ingénierie élémentaires " qui définissent le système et ses interfaces telles que les dimensions, les forces, les mouvements et toutes les autres variables de conception élémentaires. L'analyse de la propagation des changements est effectuée au niveau des paramètres en tenant compte du degré de dépendance entre les différents paramètres. Le véritable défi de cette approche est d'identifier les paramètres significatifs qui doivent être modélisés.

Kocar et al. (2010) [74] ont proposé «ADVISE (Active Distributed Virtual Change Environment ), un environnement virtuel pour l'ECM intégrant des environnements de conception collaborative virtuelle et des techniques d'exploration de modèles séquentiels pour faciliter l'ECM. ADVISE apporte une vision claire de la modélisation de l'ECM et fournit une solution compacte pour combiner des données paramétriques et graphiques dans un objet virtuel. Grâce à cet outil, une modification technique peut être perçue facilement, rapidement et avec précision, même par les membres non techniques de la commission de modification technique. Cependant, la contribution la plus importante de ADVISE est la saisie des données

impliquées dans la gestion des modifications, le traitement des données et le retour à la gestion des modifications elle-même pour offrir des "conseils" sur les paramètres de décision les plus importants dans la gestion des modifications techniques. Il est important de souligner que le processus de propagation du changement au sein d'ADVICE utilise des données historiques comme entrées et applique une technique d'exploration séquentielle pour faire émerger des modèles et probabilités potentiels. Par conséquent, l'outil ne peut être utile dans le cas de changements inhabituels pour lesquels il n'existe pas d'expérience antérieure similaire. De plus, le système a besoin d'une énorme quantité de données historiques collectées et stockées dans le bon format dans la base de données de l'ECM. Il ne peut pas récupérer les informations textuelles collectées lors des revues de conception.

Les méthodes multi-compagnie présentées dans la présente section sont des méthodes qui rentrent dans le cadre de l'ingénierie simultanée et qui prennent en considération l'aspect collaboratif. Mais ces différentes méthodes de gestion des modifications sont des méthodes qui nécessitent la détermination des variables ou des paramètres d'ingénieries significatifs qui doivent être modélisés. Aussi ces dernières ne tiennent pas compte de la propagation des changements (effet boule de neige).

### **2.3.3.3 Méthodes multicouches**

Chen et al. ont proposé une méthode fondée sur des modèles pour la ré-planification de la conception afin de contrôler efficacement et de réduire la propagation des changements de conception ([75], [76], [77], [78]). Leur approche se déroule en trois phases. Dans la première phase, un problème général de ré-conception est formulé sous la forme d'un modèle basé sur des contraintes, composé de paramètres et de fonctions, où les paramètres décrivent les composants physiques et/ou les propriétés du comportement et les fonctions définissent leurs interrelations. Ces relations sont capturées dans la matrice binaire dite matrice de dépendance de la conception. Au cours de la deuxième phase, à la demande d'une nouvelle configuration, des solutions alternatives sont générées pour former un champ de sélection de solutions. Enfin, au cours de la troisième phase, on cherche et on sélectionne la solution optimale pour la refonte du modèle qui comporte le moins d'impact possible sur le projet. Plus récemment, Li et al. (2010) [79] ont présenté une extension de leur méthode en pré-généralisant et en stockant de manière dynamique les modèles de ré-design avant que la demande de ré-design soit faite pour accélérer la 2ème phase.

Ma et al. (2008)[80] ont proposé une modélisation des relations d'ingénierie associative dans le cadre d'un schéma de modélisation de caractéristiques normalisé et d'un algorithme de propagation du changement. Ce schéma modélise toutes les informations du produit sous la forme d'un réseau de dépendances de Features, où les Features peuvent se référer à des fonctions, des comportements, des spécifications de produit ou des détails de planification de processus. La propagation des changements est analysée entre les entités en fonction des dépendances de leurs variables. Lorsqu'une modification de variable est proposée, l'algorithme recherche dans le réseau si les entités connectées peuvent accepter cette modification et envoie un avis si l'une d'entre elles ne peut pas y répondre. Deux études de cas ont servi à illustrer la méthode proposée. Cependant, le réseau de dépendances nécessite des informations détaillées et semble être très compliqué à développer. De plus, Ma et al. n'ont pas suffisamment abordé la façon dont les Features et les variables peuvent être identifiées et, ce qui est plus critique, comment leurs interrelations peuvent être définies.

Pasqual et al.[81] ont proposé un modèle de réseau de propagation du changement multi-domaine comprenant les domaines du produit, du changement (processus), et du social. Ainsi, le domaine de produit est un réseau de composants, le domaine de changement un réseau de demandes de changement, et le domaine social est un réseau de personnes. Pour l'analyse des changements au sein de ce réseau, ils ont proposé un référentiel d'outils et de métriques existants.

Albers et al.[82] implémentent un modèle de contact et de distribution dans l'environnement du logiciel Cambridge Advanced Modeler (CAM). Leur outil permet de modéliser les relations fonctionnelles entre la fonction et la forme et de révéler les liens entre les exigences fonctionnelles et les composantes physiques sur plusieurs niveaux de l'abstraction. Leur modèle peut soutenir l'analyse de prédiction des changements en combinaison avec la CPM[83].

Rahmani et al.[84], [85] ont proposé une représentation graphique d'une interface qui a été implémentée dans un logiciel basé sur Java. Leur outil permet de relier les données du produit de plusieurs domaines d'ingénierie et de classer et représenter les interfaces dans un format structuré. Il améliore le partage de l'information et la coordination entre les équipes de conception collaboratrices et peut être utilisé pour soutenir la gestion des changements inter-domaines et interdisciplinaires.

Les méthodes présentées dans cette section sont des méthodes dites méthodes multicouches car elles prennent en considération les différentes couches (provenant des différents domaines d'ingénierie) du produit qui sont généralement présentées sous la forme d'un réseau de dépendances. Le défi de ces méthodes c'est de déterminer les relations de dépendance reliant les couches en question.

## **2.4 La propagation des changements**

Un nombre important de relations entre les différents composants d'un produit conduit à des interactions complexes pendant le processus de développement. Les composants des produits sont reliés entre eux par différents paramètres tels que l'espace, la géométrie, le matériau, la fonction et le comportement.

En raison de ces interactions complexes, une seule modification apportée à une pièce peut avoir des conséquences sur les autres pièces et entraîner des modifications supplémentaires. Ce phénomène de boule de neige peut prendre différentes formes et provoquer une cascade de modification. Ce processus est appelé propagation des modifications. De nombreux modèles de prédiction du changement ont été développés dans différents travaux scientifiques, se concentrant principalement sur la prédiction des changements, leurs chemins de propagation et sur le développement d'outils de gestion de ces changements et leurs propagations. Dans les parties qui suivent les méthodes de propagation sont classées en trois grandes catégories : Méthodes matricielles, méthodes nécessitant une base de données et méthodes basées sur d'autres concepts sous-jacents.

### **2.4.1 Méthodes matricielles**

La première étape consiste à décomposer le produit en ses sous-systèmes et à mapper leurs dépendances dans une matrice telle que la Design Structure Matrix (DSM). La DSM est un outil de gestion de complexité qui se focalise sur les éléments d'un système complexe et leurs interrelations mutuelles. Clarkson et al.[62] ont développé un modèle de prédiction du changement (CPM) qui utilise le DSM pour établir les dépendances entre les composants d'un produit et utilise ensuite des techniques de gestion des risques pour prévoir le risque de propagation d'un EC entre les composants de celui-ci. Pour calculer le risque de propagation d'un changement, la valeur de probabilité et la valeur d'impact doivent d'abord être générées et ensuite multipliées entre elles. Pour chaque changement initié, le risque de propagation prévu

peut être calculé et ensuite tracé sur des diagrammes de dispersion des risques pour une comparaison facile.

	A	B	C	D	E	F	G
Element A	<b>A</b>	1				1	
Element B		<b>B</b>		1			
Element C	1		<b>C</b>				1
Element D				<b>D</b>	1		
Element E		1			<b>E</b>	1	
Element F			1			<b>F</b>	
Element G	1				1		<b>G</b>

Fig.2. 5. Design structure Matrix.

Koh et al.[34] ont récemment développé une méthode de modélisation du changement (CMM) qui est basée sur la CPM et sur la House of Quality (HoQ) de Hauser et al.[86]. À l'aide de la MMT, il est possible d'obtenir des chemins de propagation des changements qui sont éventuellement déclenchés par les différentes options de changement et qui sont ensuite évalués en termes de leurs influences sur les attributs du produit afin que le designer puisse choisir la variante optimale pour le changement.

Tang et al.[87] ont élaboré un système de gestion d'EC basé sur la DSM (ECM) qui peut non seulement déterminer la présence ou non d'une relation entre deux éléments, mais qui comporte également un champ d'information supplémentaire parallèlement au DSM où des détails concernant les dépendances, comme le type (ex. Matériau ou géométrie) et leur degré de dépendance peuvent être indiqués. En plus d'un DSM pour le domaine produit, deux représentations DSM supplémentaires sont construites, l'une pour le processus et l'autre pour le domaine organisationnel. De cette façon, on obtiendra une vision plus globale et plus globale de la propagation de l'EC.

Flanagan et al.[88] ont introduit une méthode d'analyse fonctionnelle de la propagation du changement (FACP) qui est basée sur le DSM et qui doit aider les concepteurs à trouver d'éventuelles trajectoires de propagation du changement, les évaluer et puis leur permettre de choisir la meilleure solution. Alors que certaines méthodes, comme le CPM, ne prennent en compte que les dépendances entre composants, le FACP intègre également les relations

fonctionnelles dans son analyse (Fig.2. 6). De cette façon, le changement n'est pas seulement considéré en termes de fonction ou de forme, mais est plutôt considéré comme étant une combinaison des deux puisque " deux composants (formes) ne peuvent s'influencer mutuellement que s'il existe un lien fonctionnel entre eux "[88]. Mais, l'approche proposée ignore les liens indirects qui peuvent exister.

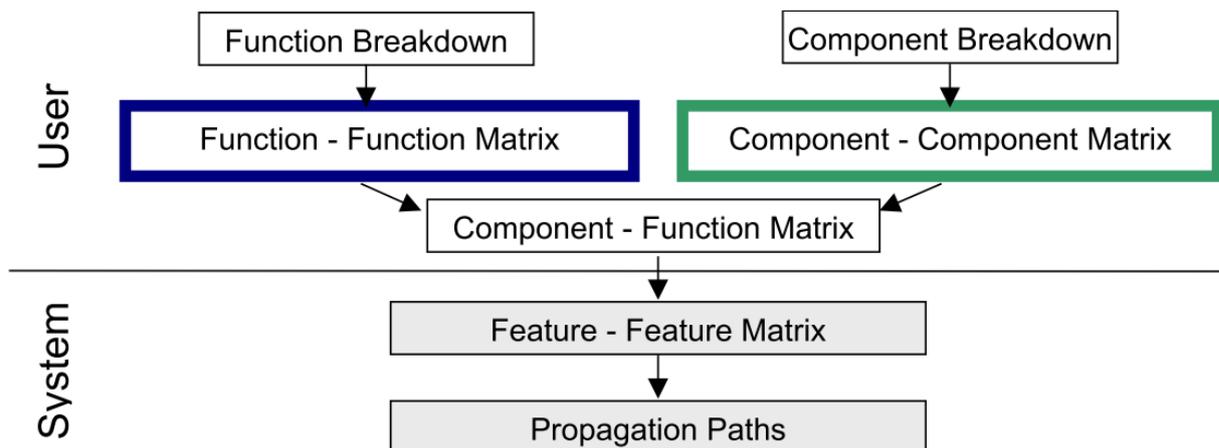


Fig.2. 6. Méthode proposée par Flanagan et al. [88].

En utilisant les faces topologiques comme unités de traitement de base et en se concentrant sur les relations de contraintes qui existent entre les faces topologiques, l'étude de Leilei et al.[89] a permis de justifier la propagation des changements et de faire progresser la gestion de la modification dans la conception des outils de montage de façon proactive. Quatre contributions de cette étude peuvent être soulignées. Premièrement, pour une gestion efficace d'EC, les lacunes de la DSM construite au niveau des composants ou des pièces ont été soulignées et un modèle empirique basé sur les contraintes topologiques a été présenté au niveau des faces. Deuxièmement, le CV (Change Vector) représentant les changements dans les faces topologiques et la RM présentant les relations entre les faces topologiques ont été introduites. Troisièmement, des algorithmes ont été proposés pour atteindre l'objectif de détection de la propagation des changements de pagination des aéronefs à l'outillage d'assemblage. Quatrièmement, l'indice CI (Change Influence) est proposé pour évaluer l'influence des changements.

Les méthodes matricielles présentées sont tous des méthodes basées sur la matrice DSM. Les dépendances décrites par cette dernière diffèrent d'une méthode à une autre (quantitatives,

qualitatives...). La limitation importante de ces méthodes est le nombre de composants du système. Plus le système est complexe plus le calcul nécessite des efforts plus importants et le risque d'erreur augmente.

#### **2.4.2 Méthodes nécessitant une base de données**

Une autre approche pour traiter les ECP est l'utilisation d'une base de données. La plupart du temps, les auteurs suggèrent une approche basée sur des bases de données pour réduire la dépendance à l'égard des connaissances d'experts, pour rendre la méthode aussi faisable aux designers débutants/non techniquement expérimentés ou pour permettre à de nombreuses personnes de participer. Kocar et al.[74] ont conçu l'Active Distributed Virtual Change Environment (ADVICE) qui peut être utilisé dans le cadre de la gestion des changements d'ingénierie (ECM) et qui améliorera le processus ECM en fournissant aux designers les infotextes et des graphes, afin de pouvoir également servir aux membres non techniciens de l'Engineering Change Board (ECB). ADVICE a pour objectif d'apporter un soutien à ses utilisateurs en donnant une priorité aux ECR et en prévoyant une propagation possible. En modélisant le système ECM proposé dans un environnement de conception collaborative virtuelle, une représentation 3D simulée en temps réel et partagée d'EC' peut être fournie [74].

Grantham et al. (2006) [90] ont mis au point la méthode REM (Risk in Early Design Méthode) qui permet d'évaluer les risques avant que la conception de la forme du produit soit décidée, donc au moment de son élaboration. Le REM est une extension de la méthode de conception des fonctions de défaillance (FF DM) qui relie les fonctions du produit aux défaillances qui ont déjà eu lieu. En stockant les défaillances antérieures dans une base de données et en les utilisant pour évaluer le risque qu'une défaillance similaire apparaisse dans la conception actuelle, la dépendance vis-à-vis des experts qui identifient les risques en fonction de leur propre expérience sera réduite et donc les biais seront éliminés.

Ma et al.[80] ont développé une méthode qui modélise les relations d'ingénierie associative dans un schéma de modélisation de caractéristiques uniformes, d'où l'appellation FUMAS (Unified Feature Modelling Scheme) dans ce qui suit. Les auteurs affirment que les données obtenues aux étapes antérieures de la conception sont nécessaires aux étapes ultérieures, alors qu'en même temps, les étapes ultérieures influencent les décisions qui ont été prises au cours des étapes antérieures, ces données sont alors enregistrées dans une base de données. Ma et al. Proposent donc un moyen par lequel le contrôle de la cohérence de l'information entre les

différentes applications aux différentes phases du cycle de vie des produits est assuré et l'efficacité des ECP peut être optimisée à tous les niveaux.

Les méthodes d'ECM qui nécessitent des bases de données sont des méthodes qui ont pour objectif de minimiser l'intervention des experts. En effet les bases de données utilisées sont généralement construites à l'aide des données historiques. Ce qui limite ces méthodes lorsque des modifications inhabituelles auront lieu.

### **2.4.3 Méthodes basées sur d'autres concepts sous-jacents**

Les méthodes suivantes ne nécessitent pas de bases de données peuplées et sont basées sur d'autres approches que l'utilisation de matrices telles que le DSM pour décomposer les conceptions de produits en fonction de leurs dépendances. Cohen et al.[59] ont mis au point la méthodologie de la représentation favorable au changement (C-FAR) qui vise à faciliter la représentation, la propagation et l'évaluation qualitative du changement. Ainsi, les informations sont extraites du modèle de données STEP (Standard for the Exchange of Product) afin de faciliter la traçabilité des changements. La méthode C-FAR est basée sur EXPRESS qui définit ces artefacts principaux comme des objets ou des entités qui, à leur tour, sont décrits par leurs attributs. C-FAR considère donc les dépendances d'un produit en fonction de son niveau d'attribut.

ReDesignIT (RDIT) est un outil informatique, développé par Ollinger et al.[68], qui est destiné à être utilisé au cours des premières phases du redesign et qui génère et évalue différents plans de re-design. Le programme classe les différentes demandes de modification en fonction de leur efficacité et détermine les effets indésirables, dont on peut contrer. Les auteurs admettent que les plans de modifications proposés sont plutôt abstraits, car ils ne montrent que les différentes variables et dans quelle direction elles doivent être modifiées, et ne peuvent pas préciser les valeurs numériques pour ces variables.

Yang et al.(2011)[91] ont récemment élaboré une méthode de recherche des chemins de l'ECP en tenant compte des liens entre les paramètres afin d'aider les concepteurs à trouver des solutions optimales de changement, c'est pourquoi cette méthode sera appelée la méthode PLN (PLN-base Méthode). Tout d'abord, le modèle PLN doit être créé en analysant les liens entre les différents paramètres du produit. Les auteurs distinguent ainsi deux types de liens : les liens de contrainte, qui peuvent être influencés par les concepteurs, et les liens fondamentaux, qui ne

peuvent être influencés. Un algorithme d'itération permet de chercher et d'évaluer les chemins de propagation ECP afin de trouver le chemin optimal.

Différentes études ont montré que le changement peut se propager aux paramètres directement et/ou indirectement liés des éléments du système. Ainsi, les changements apportés peuvent toucher d'autres services de l'organisation qui ne sont pas directement concernés par ces changements. Par exemple, le changement de la taille de l'artefact peut entraîner un changement des dimensions de la boîte d'emballage, le changement du matériau du produit peut affecter l'outil de coupe pendant la fabrication. Ainsi, le phénomène de propagation du changement est non seulement dû à l'interconnexion physique des composants du système, mais aussi aux paramètres reliant les différents services de la compagnie.

## **2.5 Modélisation des dépendances**

Afin que le concepteur puisse évaluer la propagation du changement, il est nécessaire d'identifier les dépendances entre les différents composants du produit, ce qui est une étape fondamentale pour caractériser l'impact d'un changement. Dans le cadre de la revue de la littérature sur les changements d'ingénierie (EC), de nombreux auteurs ont proposé des outils et des méthodes pour caractériser les dépendances, évaluer l'impact du changement et le propager [30][59][61][64][92][93]. Dans de tels modèles, chaque dépendance existante entre les éléments d'un produit est définie qualitativement, quantitativement, ou bien par des équations. La matrice de structure de conception ("Design Structure Matrix" ou DSM) est une matrice qui aide à identifier les composants d'un produit d'une part et les relations paramétriques ou de dépendance qui existe entre eux d'autres parts. C'est une matrice carrée de taille  $(n \times n)$  avec  $n$  est le nombre de composants. Chaque cellule (sur la ligne  $i$  et la colonne  $j$ ) indique la présence ou l'absence d'une interaction ou dépendance entre les éléments  $i$  et  $j$  dans un sens spécifique. La représentation binaire de la DSM peut être étendue de plusieurs façons, en incluant plusieurs types d'attributs pour caractériser les interactions. Ces étiquettes peuvent contenir, par exemple, le nombre d'interactions, l'importance, l'impact ou la force des dépendances.

## **2.6 Conclusions par rapport à l'état de l'art de l'ECM**

Centaines méthodes de gestion des modifications et propagation des modifications présentées dans les sections précédentes ont montré des résultats satisfaisants. Mais il n'empêche que les

méthodes se basant sur des approches matricielles présentent toutes des limitations importantes lorsque le nombre de composants du système est important. Les efforts requis (lors de la manipulation de ces matrices) dégradent les techniques proposées et ne les rend appropriées que pour des produits relativement simples ou peu complexes (n'ayant pas un nombre de pièce important). Aussi, la plupart de ces approches nécessitent encore l'intervention des experts pour identifier certains paramètres significatifs comme la probabilité ou bien l'impact d'une modification. En stockant les actions antérieures dans une base de données et en les utilisant pour évaluer le risque qu'une modification similaire apparaisse dans la conception actuelle, la dépendance vis-à-vis des experts qui identifient les risques en fonction de leur propre expérience sera réduite et donc les biais seront éliminés. Cependant, le système a besoin d'une énorme quantité de données historiques collectées et stockées dans le bon format dans la base de données l'ECM afin d'être utilisé ultérieurement. La majorité des outils présentés nécessitent donc des experts ou des concepteurs expérimentés capables de faire la décomposition initiale du modèle de produit et la cartographie des dépendances. Le reste, au contraire, nécessitent une base de données bien alimentée.

Certaines méthodes ne sont destinées qu'aux premières étapes du cycle de vie du développement du produit, tandis que d'autres peuvent être appliquées tout au long du processus de développement. Toutefois, certains documents ne précisent pas à quel moment du développement d'un produit leur méthode peut être appliquée, ni à quel niveau de complexité du produit ils conviennent

Dans ces travaux de thèse on propose un outil de gestion des modifications d'ingénierie qui se situe à l'intérieur du système CAO. D'une part l'outil proposé peut être exploité tout au long du processus de développement du produit. D'autre part le problème du choix des paramètres d'ingénierie les plus significatifs sera résolu, car l'approche proposée se base sur les paramètres intrinsèques du modèle CAO même (les associations qui existent entre les différents composants du système). Les dépendances entre les composants seront automatiquement déduites. D'où la dépendance vis-à-vis des experts est éliminée et le risque d'erreur est diminué.

### **3. Les Outils de Comparaison des modèles CAO**

#### **3.1 L'utilité de la comparaison des modèles CAO**

Les ingénieurs utilisent souvent la version initiale ou antérieure du modèle pour en créer une nouvelle version modifiée. En conséquence, un même modèle ou une même pièce existe sous plusieurs révisions (ou versions). Comme les modèles CAO (les modèles de conception assistée par ordinateur) permettent la modification rapide des modèles établis et de familles entières de produits, des révisions successives de ces produits peuvent être générées. L'évolution des modèles CAO est si rapide que l'on peut facilement perdre de vue les dissimilarités différenciant les différentes révisions. Il est donc primordial pour les utilisateurs de modèles CAO de pouvoir spécifier quelles sont les différences entre les différentes révisions afin de savoir s'il est approprié d'utiliser une révision particulière du modèle [1]. Cependant la comparaison CAO est à la fois l'un des domaines les plus importants et les moins maîtrisés de la conception et de l'ingénierie aujourd'hui. Le logiciel de comparaison évalue automatiquement le degré exact de similitude géométrique entre deux modèles, fournit une méthode permettant d'authentifier que les deux modèles sont identiques à tous les niveaux fonctionnels.

Selon Brière-Côté [94] les deux aspects clés qui caractérisent les problèmes de comparaison des modèles CAO 3D sont (Fig.2. 7) :

- Cardinalité : un modèle de référence peut être comparé à un modèle cible unique (1 :1 ou par paire) ou à de nombreux modèles (1 : n), généralement à partir de grands ensembles ;
- Niveau de détail - la quantité d'information attendue de la comparaison, qui est variable en fonction de la finalité souhaitée, allant d'un simple diagnostic "Oui-Non" ou "Échec-Réussite" à des informations détaillées sur les différences entre les modèles comparés.

Les cardinalités plus élevées exigent une plus grande efficacité informatique. De même, des niveaux de détail plus élevés exigent, des algorithmes de calcul des différences plus détaillés et, par conséquent, des schémas de représentation des différences plus complexes. Brière-Côté [94] a donc défini une relation entre le niveau de détail et la cardinalité (Tab.2. 1), ce qui lui a permis d'identifier six fonctions de base différentes pour comparer les modèles CAO 3D. Chaque fonction de base se rapporte à une question élémentaire à laquelle la comparaison des modèles CAO 3D est censée répondre et donne un aperçu du type de résultat attendu.

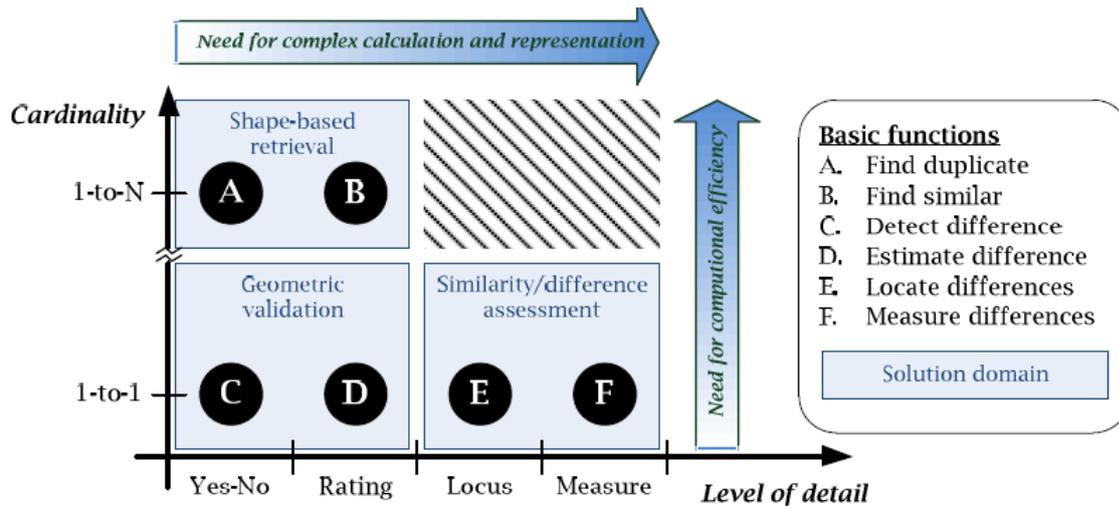


Fig.2. 7. La relation entre le niveau de détail exigé et la cardinalité et les Fonctions de base qui en résultent [73].

Pour distinguer les différences individuelles entre deux modèles, en fournissant leur contexte respectif en relation avec les formes modélisées ou, en outre, une description détaillée de leurs caractéristiques (Features), nous faisons référence à l'identification des différences qui existent entre les modèles. Nous visons à mettre en valeur les changements, extraire automatiquement la géométrie des pièces afin de les comparer : dimension des différents éléments (face, arêtes, trous, ...) permettant de reconnaître les faces identiques ...On veut aussi mettre en correspondance les éléments « ajoutés/supprimés » avec ceux du modèle initial et enfin représenté les résultats de comparaison visuellement.

Le problème de la détermination des différences entre les modèles CAO 3D qui est intrinsèquement complexe peut être divisé en trois phases :

- Une procédure, une méthode ou un algorithme permettant de comparer deux modèles CAO 3D distincts,
- La représentation, le résultat du calcul doit être représenté sous une forme qui peut être manipulée davantage ;
- La visualisation, les différences entre les modèles doivent souvent être présentées en fonction du besoin.

Tab.2. 1. Les fonctions de base de la comparaison des modèle 3D proposé par [74].

	<i>Fonction de base</i>	<i>Question de base</i>	<i>Résultat escompté</i>
<b>A.</b>	Trouver des modèles dupliqués	Quels sont les modèles équivalents ?	Ensembles finis d'objets
<b>B.</b>	Trouver des modèles similaires	Quels sont les modèles similaires ?	Distributions classées, sur une base d'échelle
<b>C.</b>	Détecter la différence	Les modèles sont-ils différents ?	Résultats binaires (Oui/Non, Pass/Fail, etc.)
<b>D.</b>	Estimation de différence	Dans quelle mesure les modèles sont-ils différents ?	Mesures qualitatives, globales et à grande échelle
<b>E.</b>	Localiser les différences	Où sont les différences ?	Rapports graphiques, localités, régions
<b>F.</b>	Élaborer les différences	Quelles sont les différences ?	Classifications, mesures locales, descriptions

### 3.2 La comparaison des modèle 3D dans la littérature

Un outil permettant de mesurer la similarité entre un certain nombre de modèles peut éviter le re-design des produits standards et existants et des composants similaires, ce qui permet aux concepteurs d'accroître l'efficacité du processus de conception. Déterminer les similitudes entre les divers composants est une étape clé pour réduire la phase de conception d'un nouveau cycle de produits. Les fichiers CAO peuvent être utilisés pour mesurer la similarité d'un ensemble de composants comparables. Les scientifiques ont proposé un certain nombre d'approches pour mesurer la similitude entre un certain nombre d'objets 3D [3 - 13] .

L'évaluation de la similitude des pièces mécaniques 3D a récemment fait l'objet d'une attention particulière. Hong et al.[95] ont classé les recherches antérieures sur les techniques de comparaison de la similarité de forme en deux catégories. Ils ont supposé que la première catégorie ne compare que la similitude géométrique et que la seconde compare certaines caractéristiques (Features) spécifiques prédéfinies. Le premier type de comparaison se base généralement sur une seule caractéristique géométrique comme le fait Ohbuchi et al.[97] qui ont adopté cette technique et ont utilisés le moment d'inertie pour juger la similitude de deux pièces différentes. Cependant, cette technique présente des nombreux inconvénients si les

pièces sont très complexes et cela prend beaucoup de temps et les résultats ne sont pas parfaits. Le second type, quant à lui, ne peut pas traiter tous les formats et a besoin de plus d'informations parce qu'il devrait extraire les Features typiques comme les trous et les filets des pièces et les comparer[98]. Bien que cette technique nécessite des informations plus précises, elle nécessite moins de calculs et donne des résultats plus raisonnables.

Il existe aussi des méthodes de détection de similitude basées sur la "signature". La signature de forme remplace la géométrie 3D comme modèle aisément compréhensible et transférable de géométrie 3D pour la comparaison numérique des modèles [99]. Il existe différentes approches pour convertir une pièce en ses signatures, telles que les méthodes statistiques, les méthodes basées sur des graphiques, la technologie de groupe (GT), etc., telles que présentées à la figure (Fig.2. 8).

Elinson et al.[98]ont décrit une approche permettant de mesurer la similarité entre les modèles CAO en générant *des graphes de structure* à l'aide des fonctions d'usinage des modèles, puis il compare les graphes des différents modèles afin de déterminer la similitude entre eux. Bespalov et al.[100] présentent une technique de mise en correspondance des formes par la décomposition spectrale des modèles 3D. Pour ce faire, on décompose les modèles à l'aide d'un algorithme de décomposition en leurs structures arborescentes en racine respectives et on calcule ensuite les distances entre ces arbres. Lu et al.[101] proposent une approche basée sur les fonctions géométriques partielles et sur l'histogramme de la courbe-squelette. Ici, un courbe squelette est extrait d'un modèle 3D, puis un histogramme de distribution est généré à partir des segments du squelette assemblé en fonction des informations topologiques. Cet histogramme est utilisé pour la mesure de similarité

La distribution de forme est une autre méthode utilisée pour mesurer la similarité des objets 3D [102], [103]. L'idée principale est de calculer une distribution statistique de la distance entre deux points générés aléatoirement à la surface de l'objet et d'estimer la dissimilitude de forme à partir de cette distribution. Il s'agit d'une approche basée sur la géométrie. Mais il a aussi ses imperfections inhérentes. Les pièces qui ne sont pas très différentes en forme mais ayant des fonctions de conception (Features) qui sont distinctes ne peuvent pas être bien différenciées [104]. Dans le même contexte de détection de similarité Siddique et al.[105] développent une application informatique en Visual C++ afin de déterminer les points communs dimensionnels et positionnels entre deux pièces mécaniques données en dégagant les dimensions issues des

fichiers IGES des modèles représentés. Cependant, l'information sur les Features ne peut pas être extraite de ce type de fichiers. Cette limitation restreint son utilisation.

Chu et al.[104] considèrent que les méthodes les plus efficaces sont celles qui génèrent une "signature de forme" à partir d'un modèle CAO et comparent ensuite différentes signatures selon une fonction de mesure prédéfinie (ou fonction de similarité). Aussi, ils ont classé les méthodes pour accéder à la similarité des formes 3D principalement en six catégories selon les informations CAO qui composent la signature créée, en se basant sur : les Features, fonction spatiale, histogramme, image en coupe, graphe topologique et statistiques des formes. Encore une fois, chacune de ces méthodes a ses propres restrictions dans la discrimination des formes 3D. La principale limite de toutes ces méthodes (basées sur la création d'une signature du modèle 3D) est le fait qu'il n'existe pas de signature de forme unique. D'ailleurs, ces études de mesures de similarité supposent la transformation de la forme initiale du modèle pour changer les objets dans un format comparable, ce qui peut augmenter le risque de perdre d'importantes informations géométriques liées à la conception du produit. Dans le contexte de la conception, l'entité géométrique (la forme) est importante pour mesurer la similarité entre un certain nombre d'objets 3D.

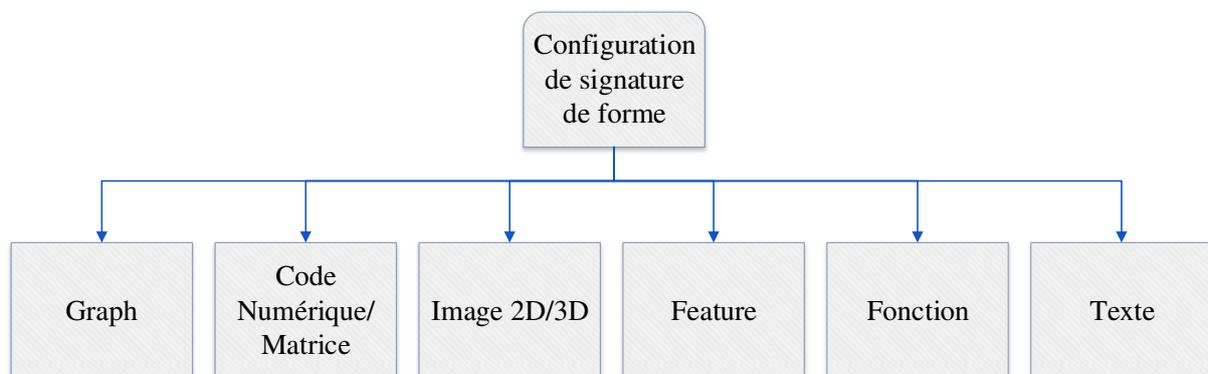


Fig.2. 8. Les différents types de signatures existantes.

La plupart de ces travaux n'utilisent donc pas le modèle CAO (modèle 3D) en lui-même mais font des transformations (STEP Files, Signature, Graphs...) afin de pouvoir enfin comparer les pièces. Cependant, le passage du modèle 3D aux différentes formulations comparables utilisées plus tard pour la détection de similarité peut entraîner une perte significative d'informations. La transformation ruine les caractéristiques géométriques des modèles et donc l'identification de caractéristiques identiques n'est plus possible. D'un autre côté, ces méthodes ignorent les vraies

caractéristiques significatives du modèle. Cependant, une approche pour la mesure automatisée de la similarité d'un ensemble de modèles CAO 3D qui extrait automatiquement les informations géométriques du modèle CAO lui-même a été proposée par Sagar et al.[96]. Ce modèle génère des fichiers texte avec les informations triées pour étayer le processus de comparaison et à examiner la similarité des modèles. Une macro a été écrite en Visual Basic incorporant toutes les API nécessaires pour extraire les informations au format texte des modèles SolidWorks.

Le principal défi de la mesure de la similitude entre un certain nombre de modèles 3D de représenter les formes pour en évaluer l'efficacité algorithmique, sans modifier l'information géométrique et en permettant une indexation facile. Malheureusement, les approches proposées par les différents chercheurs, comme indiqué dans la section 2, n'accordent pas d'importance à la forme géométrique des modèles, mais plutôt à la représentation graphique et non pas à la représentation CAD qui a orienté les recherches sur la correspondance globale, ignorant ainsi les formes de la géométrie partielle. Au cours des processus d'appariement global des formes, les transformations ignorent souvent les attributs des modèles CAO, qui contiennent des informations importantes sur la conception. Les modèles CAO doivent être créés à l'aide de certaines fonctionnalités avec dimensions spécifiques. Les caractéristiques elles-mêmes, leurs dimensions et les positions relatives ont un rôle important à jouer pour représenter le caractère unique du modèle. Pour mesurer la similitude entre les modèles 3D, les informations géométriques (caractéristiques, dimensions, position, etc.) doivent être prises en compte afin qu'elles ne soient pas détériorées au cours du processus de transformation

### **3.3 La comparaison avec les systèmes de CAO 3D**

La fonctionnalité de Comparaison est généralement disponible dans les systèmes de CAO 3D parmi de nombreuses autres fonctionnalités d'analyse de modèles numérique.

Selon leur documentation respective, l'objectif principal des systèmes de CAO 3D, en fournissant des capacités de comparaison de modèles par paires, est de permettre aux utilisateurs de comparer les versions successives d'un modèle et, en fin de compte, de gérer le changement dans les modèles CAO 3D. Notamment, les systèmes de CAO 3D sont les seuls logiciels qui permettent de comparer les représentations procédurales des modèles, c'est-à-dire la séquence des opérations de modélisation et leurs paramètres correspondants (aussi appelés arbres de caractéristiques, arbres de spécifications, historiques de construction ou de

modélisation, etc. La comparaison des modèles importés est également possible, mais dépend fortement des capacités d'importation du système de CAO 3D, qui sont susceptibles d'induire un certain niveau de dégradation des données au cours du processus de conversion (par exemple, perte des Features des modèles, modifications topologiques, etc.)

### 3.3.1 La comparaison sous SolidWorks

Si on dispose de 2 documents sur SOLIDWORKS qui semblent proches, il est parfois difficile de détecter quelles modifications ont été apportées d'un modèle à l'autre. SOLIDWORKS dispose d'un outil permettant de mettre en valeur les différences entre ces deux modèles. C'est la fonction « COMPARER » qui est disponible, par défaut, dans l'onglet « Évaluer » et qui permet de comparer des modèles 3D. Après avoir sélectionné les 2 fichiers à comparer, l'outil liste les différences de propriétés, sur les fonctions et la géométrie, détectées entre les deux pièces.

En effet, la comparaison de géométrie permet de mettre en valeur les faces ainsi que les volumes différents entre les 2 pièces. Ce type de comparaison permet d'identifier rapidement les similitudes et les différences géométriques. C'est un outil qui fonctionne bien avec les fichiers importés (géométrie morte) puisqu'il n'examine que la géométrie, et non les Feature d'une pièce. C'est un moyen simple de comparer les différentes modifications apportées à un fichier et de comprendre les changements de conception. Comparer la géométrie peut être réalisée de deux manières : Comparaison de volume et Comparaison de faces. Chacun de d'entre eux montrera graphiquement ou visuellement les volumes / faces communs, uniques (qui existent dans une pièce et non pas dans l'autre) et modifiés des fichiers.

Tandis que, la comparaison des fonctions ou « Features » permet de lister toutes les Features du Feature Manager et de marquer celles qui ont été ajoutées ou enlevées. Il est également possible de détecter les modifications au niveau des paramètres qui ont été apportés à chaque fonction. En conclusion, on peut ainsi trouver plus rapidement les ajouts, suppressions et modifications de matières apportés entre 2 documents (comparaison des volumes). De même pour l'option Comparaison de faces, dans laquelle on peut choisir de mettre en évidence les faces non modifiées, les faces uniques ou les faces modifiées.

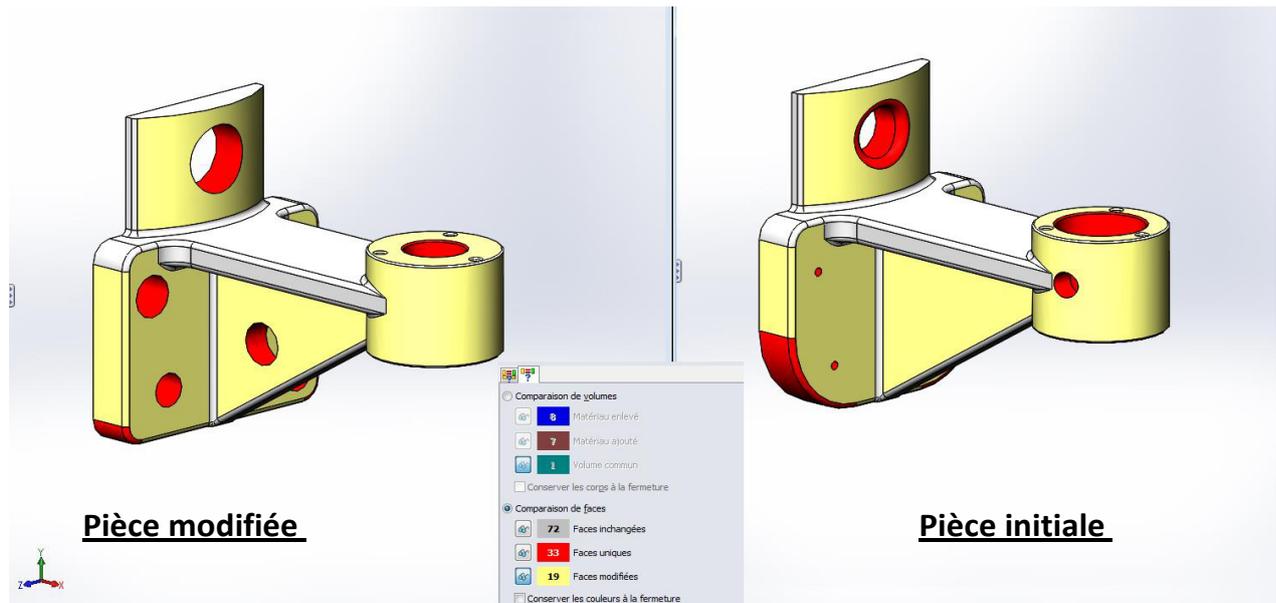


Fig.2. 9. Visualisation des différences entre les 2 modèles en utilisant l'outil « Comparer » du SolidWorks.

On peut visualiser le résultat de comparaison sous différentes formes (Fig.2. 9). D'abord, sous la forme visuelle où les zones ajoutées ou enlevées sont mises en évidence grâce à des couleurs différents. Avec l'exemple choisi on peut mieux comprendre la visualisation des modifications, en gris (Couleur initial de la pièce) la partie de la pièce qui n'a pas été modifiée, en jaune celle qui a été ôtée/remplacée et en rouge, celle ajoutée.

### 3.3.2 La comparaison sous Catia V5

Si on veut comparer deux versions successives d'une pièce avant/ après modification l'Atelier Compare Parts (atelier Conception mécanique >Healing Assistant) permet de réaliser cette comparaison.



Ce module, intitulé « Vérification géométrique », a pour fonction de comparer deux versions d'une Part et d'en extraire les différences. On peut visualiser le résultat sous différentes formes (Fig.2. 10). Tout d'abord, sous la forme visuelle où les zones ajoutées ou enlevées et inchangées sont mises en évidence grâce à des couleurs. Pour l'exemple choisi, on obtient l'aperçu ci-

dessous avec en bleu la partie de la pièce qui n'a pas été modifiée, en jaune celle qui a été ôtée/remplacée et en rouge, celle ajoutée.

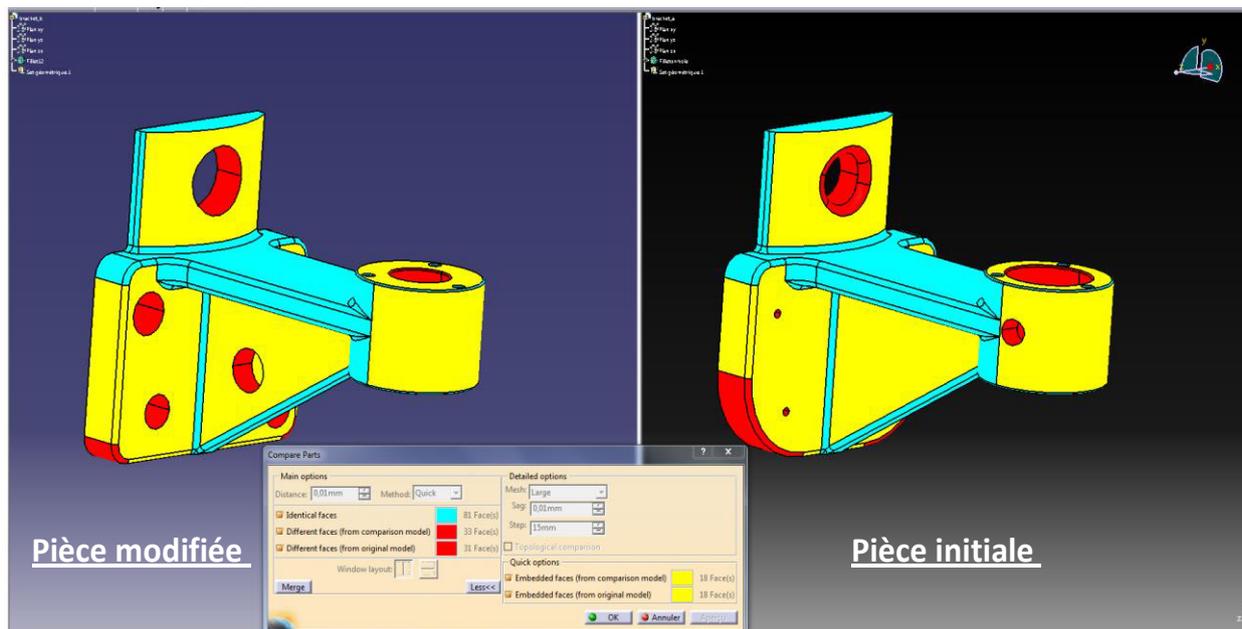


Fig.2. 10. Visualisation des différences entre les 2 modèles en utilisant l'outil « Compare parts » sous Catia.

Il est ensuite possible d'obtenir un résultat plus détaillé sous la forme d'une pièce contenant différents sets géométriques comprenant les géométries des faces résultantes. L'outil permet de choisir le modèle sur lequel on souhaite afficher les résultats. L'affichage fusionne tous les sets géométriques (géométries inchangée, ajoutée et enlevée), il faut alors montrer/cacher les sets selon le résultat que l'on souhaite sélectionner.

### 3.3.3 Conclusion par rapport aux outils proposés dans CATIA et SolidWorks

Concernant le module de comparaison sous Catia, les résultats ont surtout un intérêt pour la visualisation car les courbes (résultat « physique » de la comparaison) ne sont pas associées aux faces dont elles sont extraites (courbes « mortes »). De plus, il est impossible d'obtenir les informations liées à la géométrie des faces modifiées alors que la représentation des résultats du calcul est l'un des ingrédients nécessaires pour une comparaison de modèles 3D réussite.

Ce module de Catia utilisé indépendamment ne permet pas d'atteindre l'objectif recherché et c'est la même chose pour le module de comparaison sous SolidWorks : Il est impossible d'obtenir les informations géométriques des faces modifiées (présentation des résultats de la

comparaison). D'autre part, il ne permet pas la distinction entre les faces qui ont été supprimés et celle qui sont récemment créés. Ces deux types de face sont pour lui, les deux, tout simplement des faces qui existent dans un modèle et pas dans l'autre. Pourtant il est très important de les différencier car perdre des faces implique la perte des nombreuses informations, comme liens existant avec celle-ci, ce qui peut même endommager la cohérence du modèle entier. Aussi, il est impossible de faire une comparaison géométrique et de Feature à la fois. Cependant, la combinaison de ces deux outils pourrait être une solution très efficace (*comparaison des éléments géométriques et des features*). D'où vient l'idée de l'implémenter à l'aide des API du SolidWorks et des fonctions prédéfinies qui n'est pas une tâche facile.

#### 4. Analyse des associations et des objets associés

Les termes « association », « relation », « contrainte » et « lien » ont été déjà définis dans le chapitre précédant. La différence entre eux a été bien mise en valeur. En général, les différents sens accordés à ces termes sont :

- Le terme association veut dire groupement, réunion en un tout de choses diverses. « Associer », c'est le fait de réunir des choses en vue de les rendre solidaires, de les mettre en accord ; joindre en soi différents éléments ou caractères.
- Relation : Caractère, état de deux ou plusieurs choses entre lesquelles existe un rapport.
- Contrainte : Obligation créée par les règles en usage dans un milieu, par les lois propres à un domaine, par une nécessité.
- Lien : Ce qui établit entre des choses abstraites un rapport, en particulier logique ou de dépendance.

##### 4.1 Utilisation des termes dans la littérature

Dans le cadre de la gestion et propagation des modifications au sein d'un assemblage les dépendances ont été utilisées comme un outil qui expriment si un changement de la valeur d'un élément ( de l'assemblage) peut générer des changements sur les valeurs des autres paramètres ou des autres éléments [3]. En effet, une dépendance entre deux paramètres d'un produit est définie comme : « l'effet direct du changement de la valeur d'un paramètre sur un autre » [3][106]. Les matrices DSM /DMM sont des outils qui aident à capturer et visualiser les dépendances existantes dans un modèle produits ([72], [85], [107]).

Les graphes de dépendance ont été aussi utilisés dans le domaine de gestion des changements d'ingénierie. Dans le domaine de la gestion du changement, la théorie des graphes est utilisée pour la modélisation et la visualisation des différentes dépendances entre les éléments d'un système. Ce type de graphe est souvent appelé "Graphique des dépendances" mais le terme "Réseau de dépendances" est aussi utilisé couramment. Les matrices, DSM ou DSM/DMM sont assimilables à un graphe de dépendance dans la mesure où elles contiennent les mêmes données (valeurs de dépendance). En effet, une matrice adjacente d'un graphique (avec laquelle un graphique est construit) coïncide avec la définition d'un DSM. Ainsi nous trouvons l'utilisation de ces deux types de structures de données (matrices et graphes de dépendance) dans les méthodes ECM.

Tandis qu'en CAO, les associations sont définies comme des contraintes géométriques et paramétriques établies entre les entités géométriques (faces, arêtes, sommets). Ma et al.[108] distingue deux types d'associations : géométriques et non géométriques. Cependant, Tremblay [109], arrange les associations en 3 types : relation , liens et contraintes. Afin de caractériser ces différents types d'associations, quatre propriétés ont été déterminées : la Cardinalité qui définit le nombre d'entités utilisés, la Direction, La temporalité qui est une caractéristique qui montre qu'une association est persistante, transitoire ou semi-persistante, la dernière propriété c'est l'agrégation et la décomposition : l'agrégation est le regroupement d'un ensemble d'associations afin de faciliter sa manipulation à un niveau d'abstraction supérieur. La décomposition est l'opération inverse. Dans les travaux de Louhichi et al.[110] les associations sont des contraintes d'assemblage établies au sein de la DM (Assemblage CAO). Selon Chen et al.[111] une association décrit un groupe de liens structurés de manière sémantique. Dans le contexte d'un assemblage, plusieurs auteurs ([30], [19], [25]) utilisent le terme relation de contact (ou relation) pour décrire une dépendance entre deux composants d'un assemblage.

Dans la suite de notre travail nous visons à créer notre propre modèle de dépendance. Les associations utilisées seront les contraintes d'assemblages. Sous SolidWorks les contraintes se regroupent en trois catégories : Les contraintes standards, les contraintes avancées et les contraintes mécaniques. Les contraintes standard sont les plus utilisées. Notre modèle proposé se base principalement sur ce type de contrainte. Les contraintes standard sont les suivantes : Coïncidente, Parallèle, Perpendiculaire, Tangente, Coaxiale, Blocage, Distance et Angle.

## 5. Conclusion

Dans ce deuxième chapitre, une étude bibliographique approfondie a été faite. Cette étude a été principalement divisée en trois grandes parties : la première partie a été consacrée aux différents outils de gestion des modifications. On a commencé par le Système de Gestion de Données Techniques jusqu'au système PLM actuels. Les différents travaux de recherche qui ont été menés dans le domaine de management des changements et le domaine de propagation des modifications ont été étudiés. La seconde partie a été dédiée à la comparaison des modèles CAO, l'utilité et les outils disponibles. Enfin, la troisième partie est consacrée à l'analyse des associations et des objets associés et l'utilisation des différents termes dans la littérature.

Pour conclure, on peut constater que toutes les méthodes de propagation de changements présentées commencent nécessairement par la caractérisation des dépendances existantes entre les éléments d'un système avant même de parler du mécanisme de propagation des changements. Les éléments des systèmes présentés sont généralement liés par des relations déjà connues grâce à un modèle théorique ou obtenues via des interventions des experts par exemple. Cependant, l'obtention de ce modèle de dépendance est mise en difficulté si le modèle théorique n'est pas disponible ou s'il est incomplet. Par conséquent, nous avons pensé à la construction du modèle de dépendance via d'autres méthodes tout en éliminant l'intervention des experts, sans la nécessité d'une base de données déjà alimentée à l'avance et sans avoir recours à l'historique des modifications. L'approche proposée (dans ce mémoire) doit être aussi applicable à n'importe quel stade du cycle de vie du produit.

**Chapitre 3 : Algorithmes proposés  
pour la propagation des modifications  
au sein du modèle CAO**

# Chapitre 3 : Algorithmes proposés pour la propagation des modifications au sein du modèle CAO

## 1. Introduction

Aujourd'hui, les défis sont d'abord les délais, le coût et la qualité. Mettre un produit sur le marché le plus rapidement possible devient un objectif majeur. Le produit doit être : économique et de bonne qualité. De plus, le cycle de vie d'un produit est de plus en plus court. L'innovation constante, pour moderniser le produit, devient donc importante. La définition géométrique d'un produit progresse continuellement, grâce au travail de différents spécialistes qui utilisent, analysent et modifient les informations existantes, mais aussi décident et agissent. Ainsi, le risque de perdre des informations augmente, surtout lorsque des modifications sont apportées. D'où l'idée de créer un Modèle pour la gestion de ces changements, et surtout pour faciliter la tâche de l'utilisateur. Dans ce chapitre, on va présenter l'algorithme général sur lequel l'approche proposée est fondée et qui se divise en quatre grands sous-algorithmes. Ces sous-algorithmes sont présentés en détails. Dans une première étape les données du modèle CAO sont extraites et enregistrées dans une base de données. Dans une deuxième étape, ces données vont être exploitées afin de décrire les associations et les dépendances entre les différentes entités géométriques et caractériser les éventuels changements. Ensuite, une fois les modifications sont détectées, la réconciliation des associations existantes initialement est exigée, afin de pouvoir mesurer le risque de propagation de ces modifications.

## 2. Mise en situation

Un assemblage CAO est composé de plusieurs composants et/ou sous-ensembles combinés dans un environnement unique. Différentes relations, qui déterminent le comportement de chacun des composants dans le sous-ensemble, sont créées. Ces relations peuvent varier de simples relations basées sur les contraintes qui déterminent la position d'un composant dans l'assemblage à des relations plus évoluées telles que la capacité d'adaptation. Ainsi, chaque composant de l'assemblage est lié et contraint avec d'autres composants et par conséquent ces liens ont une importance significative pour maintenir la cohérence des données CAO. Cependant, la

conception ne se limite pas à la phase de développement. La définition du produit peut subir plusieurs changements afin d'améliorer et d'affiner la conception tout au long de son cycle de vie. Les modifications qui sont définies comme des changements de forme ou des modifications qui affectent l'ajustement, la fonction, le matériau ou les paramètres de taille sont désignées sous le terme de changements d'ingénieries. Ainsi, les contraintes ou les contraintes existant entre les composants du modèle CAO peuvent être facilement affectées ou endommagées. Afin de gérer ces modifications nous nous basons par suite de nos travaux, sur les contraintes d'assemblage. Ce modèle de gestion des assemblages CAO permet de réconcilier un assemblage CAO après modification des pièces et d'aider l'utilisateur à propager les modifications apportées.

À première vue, il semble facile d'organiser et de suivre les études de conception, en utilisant les modèles d'ingénierie, les dessins techniques et les assemblages. Mais au fil du temps, des révisions, des versions, des ordres et des avis de modification s'imposent. Le système dérapera lorsque vous souhaitez réutiliser et/ou partager rapidement des données. Ajoutant à cela le nombre accru des intervenants, et l'obligation de maximiser le temps de réalisation des différentes tâches. Et de ce fait, il n'est pas rare que les développeurs de produit ne finissent pas à recréer tout simplement les données. Pire encore : des données essentielles peuvent ainsi être perdues ou écrasées. Prenant l'exemple du scénario industriel qui en découle lorsqu'une pièce est extraite de la maquette numérique (Assemblage CAO), envoyée à un sous-traitant, modifiée par ce dernier, puis réinsérée dans la maquette initiale. On se trouve donc devant la nécessité de réinsérer cette pièce dans l'assemblage initiale et de propager les modifications qui ont été faites tout en gardant la cohérence des données qui existent initialement dans le modèle. Bien évidemment, tous cela doit se faire dans un temps record pour ne pas retarder les autres processus de cycle de vie du produit.

### **3. Approche proposée**

Dans le cadre du travail en groupe le partage de données est devenu omniprésent et indispensable. On s'intéresse par la suite par le scénario suivant (Fig.3. 1) On dispose d'un modèle CAO initial d'entrée duquel on va extraire une pièce pour la modification. Une fois la modification a été réalisé la pièce est réinséré de nouveau dans l'assemblage. Du coup le designer doit intervenir pour gérer les changements réalisés et les propager par la suite vers le reste de l'assemblage. On obtient donc finalement un assemblage CAO modifié tout en gardant la cohérence existante initialement entre les différents composants. L'objectif principal sur lequel nous allons se

concentrer par la suite c'est la propagation des modifications. Ainsi cette étape est devisée en deux sous grandes étapes (Fig.3. 1) :

- L'analyse d'impact d'une modification ;
- La gestion de la modification.

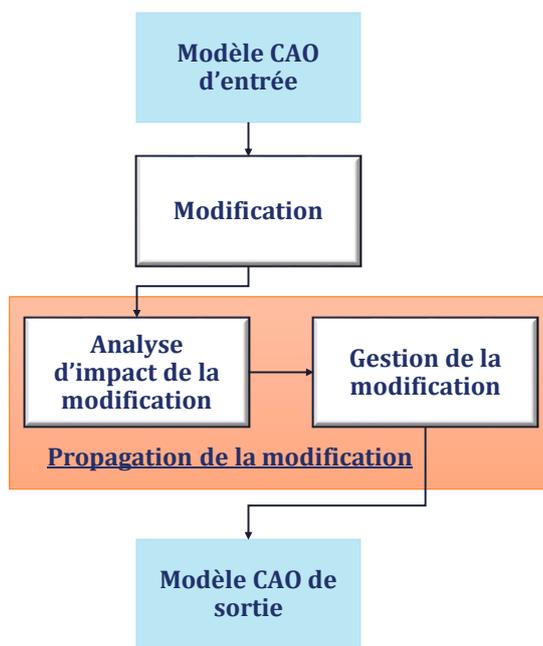


Fig.3. 1. Le processus général.

Le modèle CAO contient des informations importantes qui doivent être extraites à l'aide des API (Application Programming Interface). Ces données-là doivent être exploitées ultérieurement pour générer les différents sous-ensembles existants à partir d'un modèle d'assemblage CAO, ainsi pour déterminer les associations existantes entre les différentes « Features ». On peut distinguer deux phases dont la première phase est liée à la partie extraction des données topologiques et géométriques, tandis que la seconde est associée aux contraintes d'assemblage entre les pièces (extraction des données des contraintes). L'étape de l'extraction des données vise à explorer et faciliter l'utilisation des données topologiques et géométriques de la pièce.

Donc notre modèle est basé principalement sur ces données extraites et en particulier sur les données d'assemblages. Ces données vont être par la suite analysés et exploités pour faciliter la gestion des modifications. L'algorithme général du modèle proposé se compose principalement de quatre grandes étapes (Fig.3. 2) :

- L'extraction des données du modèle CAO, topologiques et géométriques ainsi que les contraintes d'assemblage. Ces données seront par la suite utiles pour la détermination des différentes associations existantes entre les composants de l'assemblage.
- La comparaison des deux versions de la pièce initiale notée (iPart) et modifiée notée (mPart). Grâce à cette comparaison on peut détecter et localiser les modifications mise en œuvre et on peut même quantifier leur importance.
- La réconciliation des associations existantes initialement ainsi que la détermination des entités qui ont été affectées par la modification réalisée. De plus les entités susceptibles d'être affectées par la propagation des modifications doivent être résolues.
- La propagation des modifications est donc l'étape finale par laquelle on doit procéder afin d'obtenir notre assemblage CAO final modifié (mis à jour), sans perte d'information ou de cohérence.

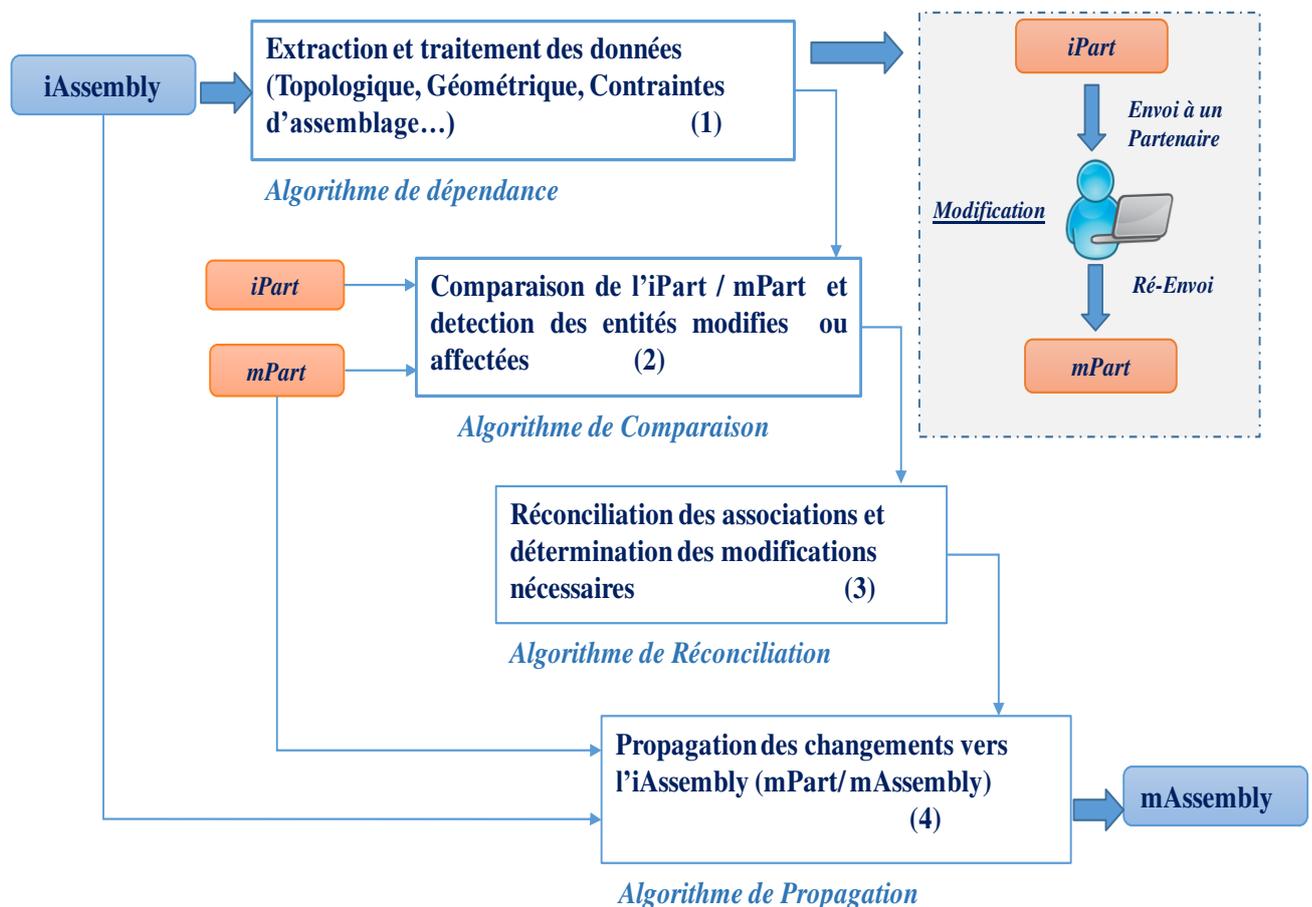


Fig.3. 2. L'algorithme général du modèle proposé.

### 3.1. Algorithme de dépendance

Notre but est ici d'identifier les dépendances existantes entre les différents éléments (composants) du modèle CAO en premier lieu et puis de formuler une caractérisation de ces dépendances.

#### 3.1.1. Extraction des données du modèle CAO

Cette première phase est la phase la plus importante qui va ensuite influencer toutes les autres phases de l'algorithme général. En effet, extraire les données du modèle est une étape fondamentale qui doit être traitée soigneusement, car ce sont ces données qui seront plus tard traitées. Toutes les autres phases sont essentiellement basées sur les résultats obtenus durant cette phase. Le générateur séquentiel d'assemblage collecte toutes les données qui sont créées dans le système CAO à partir du « Feature Manager Design Tree » ou bien l'arbre de création du modèle. Une fois l'assemblage est chargé, l'algorithme commence par énumérer les différents composants existants et extraire les contraintes d'assemblage qui les relient (Fig. 3. 3). Comme on a déjà mentionné, le déroulement de l'extraction des données se compose de deux étapes. La première étape est liée à la pièce (extraction des données topologiques et géométriques), tandis que la seconde est associée aux contraintes d'assemblage (extraction des données de contrainte). Les données reliées à la pièce sont : les sommets, les arêtes, les lignes, les faces, les surfaces et le volume. Les données extraites doivent être stockées dans la base de données générée et associée à l'assemblage en question. Pour chaque face, il est évident d'extraire les lignes associées, les arêtes, ainsi que la liste des sommets apparaissant sur cette face. En utilisant les fonctions prédéfinies depuis les API les données géométriques telles que le périmètre, la surface, la superficie ou le volume de la pièce sont ainsi facilement déduites. Bien évidemment il faut distinguer les données topologiques de celles géométriques.

Tout en sachant qu'un assemblage est un ensemble de pièces liées entre elles par des contraintes d'assemblage ou des liaisons, il est indispensable de les décrypter. Ces contraintes ont plusieurs propriétés, telles que le type, les paramètres et les pièces impliquées. Ces données sont extraites et sauvegardées dans la base de données de l'assemblage en vue d'être utilisées ultérieurement. Mathew [112] a développé un algorithme d'extraction de contrainte d'assemblage (Assembly Mate Extraction Algorithm). Les liens entre les pièces de l'assemblage sont extraits à l'aide de cet algorithme et stockés dans la base de données. Au-delà du type de contrainte et des types d'entités constituant la contrainte, le vecteur de la contrainte est également déterminé.

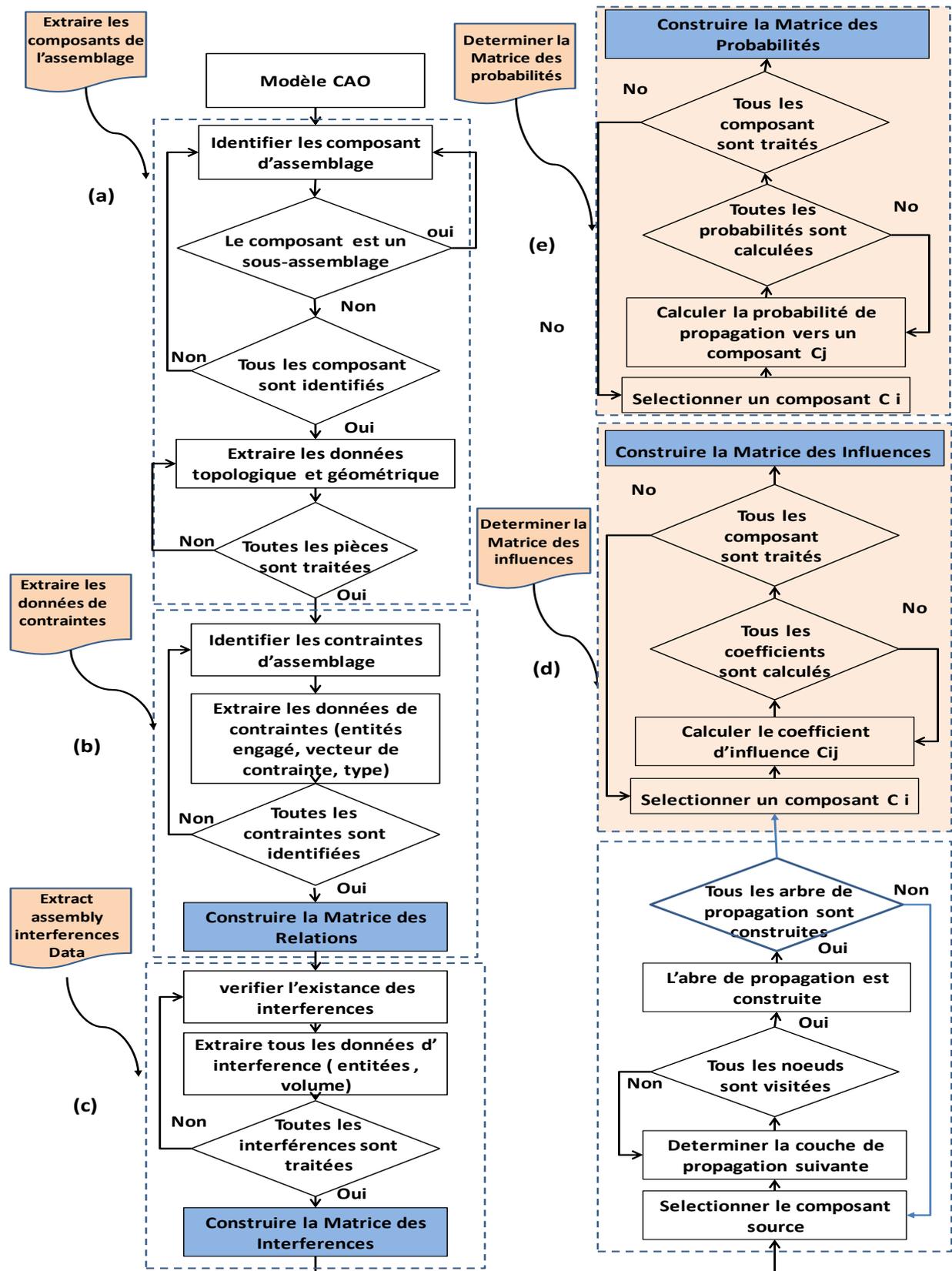


Fig. 3. 3. L'Algorithme de Dépendance.

Dans ce travail, on utilise principalement les mêmes fonctions API pour développer notre propre algorithme d'extraction de données de contrainte, qui explore les contraintes d'assemblages présents dans l'arbre de création du gestionnaire de fonctionnalités ou « Feature Manager ». Ensuite, il vérifie la nature de chaque contrainte et collecte les données de chacune d'entre elles : le nom, le type et les paramètres. La reconnaissance du type de contrainte est très importante. En fait, cela donne une idée sur la nature du contact entre les différents composants accouplés. Ces informations sont utiles pour la génération des sous-ensembles existants et la détermination des paramètres d'influence. En fait, deux composants sont assemblés par l'intermédiaire des entités topologique (Face, arêtes...). Par conséquent, la collecte des entités engagés dans les contraintes ou « Mate entities » est très utile pour obtenir la nature du contact géométrique, les paramètres géométriques et la position de l'entité, ainsi que le composant correspondant ou « Parent component ».

Une fois ces données sont extraites et sauvegardés dans la base de données qui a été créé, on doit passer à l'interprétation de ces données-là. De plus elles doivent être présentées sous une forme visuelle et simple à manipuler (Fig.3. 4).

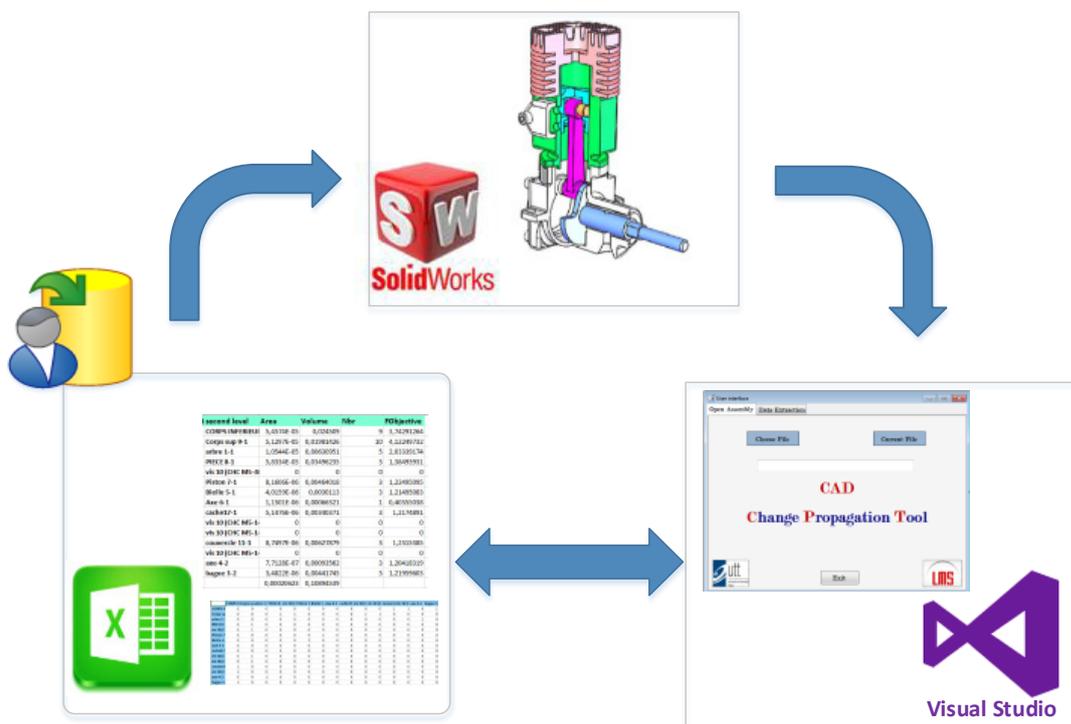


Fig.3. 4. Les différents types d'interfaces.

Les relations entre les éléments fonctionnels d'un système peuvent être divisées selon Yu et al.[113] en deux catégories : les liens fonctionnels et les interférences de la topographie. Les liens fonctionnels représentent la connexion directe de deux objets. L'autre type de relation, appelée "interférence", existe lorsqu'un objet affecte un autre élément de l'assemblage en raison de leur disposition spatiale, même s'il n'y a pas de connexion directe entre ces objets. De ce fait, on a procédé à créer des matrices dites de relations et d'interférence qui nous permettent de bien décrire les relations existantes au sein d'un modèle CAO. Ces dernières matrices vont être détaillées dans la section suivante.

### 3.1.1.1. La Matrice des Relations

La Matrice des relations notée [RM] (Relationship Matrix), est une matrice carrée symétrique, de taille N, où N est le nombre total des composants de l'assemblage. Chaque élément de la matrice RM peut avoir trois attributs possibles en fonction des données extraites précédemment :

$$RM = \begin{matrix} & C_1 & C_2 & \dots & \dots & \dots & \dots & C_n \\ \begin{matrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ \vdots \\ C_n \end{matrix} & \left[ \begin{array}{cccccc} R_{1,1} & R_{1,2} & \dots & \dots & \dots & R_{1,n} \\ R_{2,1} & R_{2,2} & \dots & \dots & \dots & R_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & R_{n-1,1} & \vdots & \vdots & R_{n-1,n-1} & R_{n-1,n} \\ R_{n,1} & \vdots & \vdots & \vdots & R_{n,n-1} & R_{n,n} \end{array} \right] & \text{avec: } \begin{cases} R_{i,j} = 0 & ;\text{si } i = j \\ R_{i,j} = Nb & ;\text{si } i \neq j \end{cases}
 \end{matrix}$$

Les relations considérées dans cette partie sont les contraintes d'assemblages reliant les différents composants. Ces contraintes sont extraites de l'arbre de création du modèle CAO.  $Nb$  est égale au nombre de contraintes qui connectent le composant  $C_i$  avec un autre composant du même assemblage  $C_j$ . On a choisi de travailler avec les contraintes car ces dernières décrivent concrètement et précisément le type de dépendance existante entre deux entités. En fait une contrainte est indécomposable, elle représente le niveau de connaissance le plus bas. Le niveau d'abstraction d'une contrainte est inférieur à celui d'un lien, étant donné que le lien est une agrégation organisée de contraintes.

### 3.1.1.2. La Matrice des Interférences

En effet, deux composant, appartenant au même assemblage, peuvent avoir un nombre de contraintes qui les relie différent de zéro, mais sans avoir un vrai contact entre eux (par exemple contrainte de : coaxialité, parallélisme...). Par conséquent, le Nb (nombre de relations) ne donne pas une idée claire sur le contact géométrique entre les composants. Cependant, la détection des interférences examine l'existence des interactions entre les différents composants d'un assemblage et détermine les différentes situations d'interférence éventuelle, telles que le contact surfacique et/ ou le chevauchement. Pour un composant donné l'outil de détection d'interférence capte tous les composants interférant avec celui-ci. La détection des interférences est faite en utilisant la fonction prédéfinie *IModeler : CheckInterference* des API du SolidWorks. Ainsi, notre matrice d'interférence est construite. Mais il faut noter qu'on a négligé le contact surfacique. Par conséquent deux composant sont considérés comme interférents si et seulement si leur volume d'interférence est différent de zéro. La matrice obtenue est aussi une matrice carrée de taille N avec N c'est le nombre total des composants d'un assemblage sans prendre en considération les éléments de fixation (les vis, les écrous, les boulons...).

$$IM = \begin{matrix} & C_1 & C_2 & \dots & \dots & \dots & \dots & C_n \\ \begin{matrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ \vdots \\ C_n \end{matrix} & \begin{bmatrix} I_{1,1} & I_{1,2} & \dots & \dots & \dots & I_{1,n} \\ I_{2,1} & I_{2,2} & \dots & \dots & \dots & I_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & I_{n-1,1} & \vdots & \vdots & I_{n-1,n-1} & I_{n-1,n} \\ I_{n,1} & \vdots & \vdots & \vdots & I_{n,n-1} & I_{n,n} \end{bmatrix} & \text{avec: } \begin{cases} I_{i,j} = 1 & \text{;s'il y a interférence entre } C_i \text{ et } C_j \\ I_{i,j} = 0 & \text{;s'il y a pas d'interférence entre } C_i \text{ et } C_j \\ I_{i,j} = 0 & \text{;si } i = j \end{cases} \end{matrix}$$

### 3.1.1.3. Le graphe des liaisons

Le graph des liaisons est un graphique qui décrit le modèle CAO ainsi que les différentes relations existantes entre ces composants avec une représentation simplifiée. Où les nœuds représentent les composants et les arcs établissent les relations entre eux. L'inconvénient majeur de ce type de représentation est la taille du graphique qui devient très important lorsque le nombre des pièces augmente (même inconvénient pour les matrices).

La représentation d'un graphe de liaison devient plus facile en présence d'une matrice d'adjacence. La matrice d'adjacence consiste à utiliser une matrice carrée ayant pour taille le

nombre de sommets du graphe. Sur les lignes et les colonnes on trouve le nom des sommets. Dans notre cas la matrice d'adjacence du graphe de liaison désiré c'est la matrice des relations construite au début. Notre matrice et une matrice symétrique, par conséquent le graphe des liaisons généré est forcément un graphe non orienté. Cependant, en présence de différent nombre de relation, le graphe sera forcément un graphe valué ou pondéré (sur les arcs on trouve une valeur qui définit le poids de la liaison) (Fig.3. 5).

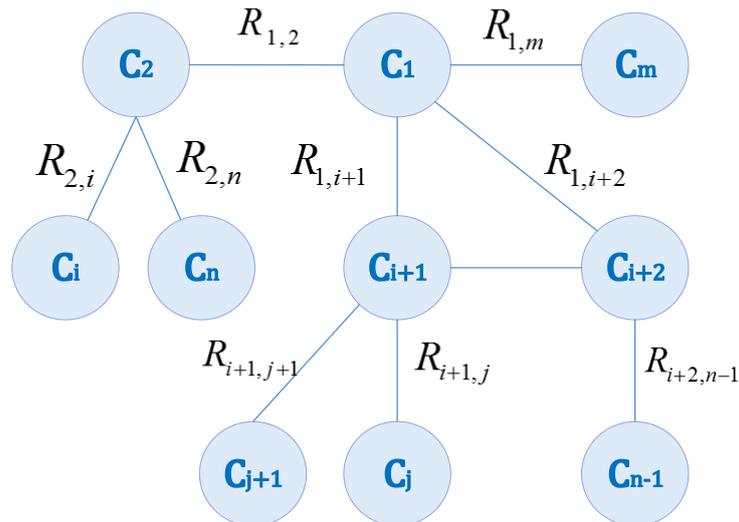


Fig.3. 5 . Exemple d'un graph des liaisons.

### 3.1.2. Arbre de Propagation

#### 3.1.2.1. Algorithme de parcours de graphe

Un arbre de propagation n'est qu'un graphe orienté non connexe et acyclique qui groupe tous les chemins de propagation possible partant d'un nœud initiateur de propagation jusqu'à atteindre l'un des nœuds d'arrêt de propagation. Contrairement aux notions d'arbre et d'arborescence en théorie de graphe (graphe de liaison), cet arbre de propagation est un graphe orienté dans lequel on peut avoir l'apparition d'un nœud plusieurs fois. En effet, différents chemins peuvent mener à un même nœud. Il est aussi important de définir une couche de propagation, elle est définie par l'ensemble de nœuds qui partagent la même distance au nœud instigateur.

Notre arbre de propagation est construit à partir du graphe de liaison définie précédemment. En choisissant le composant (le nœud) initiateur du changement nous devons déterminer tous les chemins de propagations possible ainsi que les nœuds qui vont participer à la propagation. En

théorie de graphe, il existe deux algorithmes pour parcourir un graphe ; **L'algorithme de parcours en profondeur** (ou DFS, pour *Depth-First Search*) et **l'algorithme de parcours en largeur** (ou BFS pour *Breadth First Search*). L'algorithme de parcours en profondeur est un algorithme de parcours de graphe qui se décrit naturellement de manière récursive. Son principe correspond à « la méthode intuitive qu'on utilise pour trouver la sortie d'un labyrinthe sans tourner en rond ».

La recherche d'un chemin « en profondeur » depuis un sommet S fonctionne comme suit : Il poursuit alors le chemin dans le graphe jusqu'à atteindre un sommet déjà visité. On revient alors sur le dernier sommet où on pouvait suivre un autre chemin puis explore un autre chemin. L'exploration s'arrête quand tous les sommets depuis S ont été visités. Le nom d'algorithme en profondeur est dû au fait que, contrairement à l'algorithme de parcours en largeur, il explore en fait « à fond » les chemins un par un : pour chaque sommet, il marque le sommet actuel, et il prend le premier sommet voisin jusqu'à ce qu'un sommet n'ait plus de voisins (ou que tous ses voisins soient marqués), et revient alors au sommet père.

L'algorithme de parcours en largeur (ou Breadth First Search (BFS)) diffère de l'algorithme de parcours en profondeur par le fait qu'il consiste, à partir d'un sommet donné, de visiter tous les sommets successeurs. On répète l'opération tant qu'il existe des sommets non visités. Ce type de parcours est d'une manière imagée, une sorte de propagation. En effet, on explore tous les sommets qui sont directement accessibles puis ceux qui sont accessibles en passant par un sommet intermédiaire, puis deux et ainsi de suite. Le DFS permet de trouver un chemin entre 2 sommets, tandis que le BFS permet de trouver le plus court chemin entre 2 sommets. Ce qui fait que le temps d'exécution du BFS est moins important que celui du DFS. Vu que le BFS s'est avéré plus performant que le DFS et vu de sa définition s'adapte bien avec le phénomène de propagation, nous adoptons par la suite de notre travail le principe de la recherche en largeur pour construire notre arbre de propagation à partir du graphe de liaison.

### **3.1.2.2. Création de l'arbre de propagation**

Notre graphe de liaison se compose principalement de N nœuds (le nombre de composant du système), donc pour construire notre arbre de propagation désiré on commence par préciser le nœud initiateur, ou source de modification. La détection des nœuds les plus proches (directement liées à celle-ci) représente la seconde étape. On a formé ainsi la première couche de propagation (la couche verte figure (Fig.3. 6)). Maintenant chacune des nœuds de la première couche va être

considéré comme une nouvelle source de modification ou initiateur, les nœuds qui lui sont directement liés seront détectés et ils forment ainsi la deuxième couche (la couche orange figure (Fig.3. 6)), et ainsi de suite jusqu'au compléter le balayage de tous les nœuds du graphe. Mais les nœuds déjà visité ne peuvent être considéré comme initiateur qu'une seule fois.

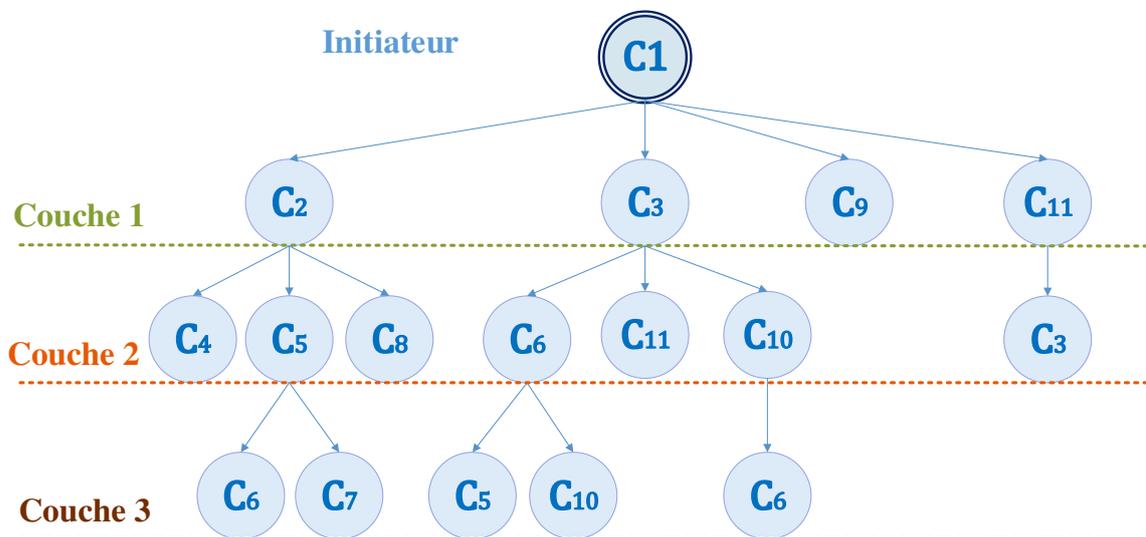


Fig.3. 6. Construction d'un arbre de propagation.

Pour atteindre un nœud donné à partir d'un nœud source (source de la modification) différents chemins sont trouvés. En partant du composant C1 on peut atteindre le composant C6 par trois voies différentes :  $C1 \rightarrow C2 \rightarrow C5 \rightarrow C6$  ;  $C1 \rightarrow C3 \rightarrow C10 \rightarrow C6$  ou bien :  $C1 \rightarrow C3 \rightarrow C6$ .

### 3.1.3. La Matrice des influences

#### 3.1.3.1. Caractérisation d'un changement

D'après Masmoudi et al.[3], « L'évaluation d'impact de la propagation de changement ne devient possible qu'après la caractérisation des changements, quantitativement et qualitativement.» Caractériser qualitativement un changement permet de donner un aperçu des variations possibles sans chercher à déterminer leurs valeurs. Tandis que la caractérisation quantitative vise à déterminer les nouvelles valeurs des paramètres affectés par la mise en œuvre d'un changement.

La caractérisation qualitative a été élaborée par Kusiak et al. [57] à travers la détermination du sens du changement d'une variable affectée. Quatre valeurs sont possibles : croissante, décroissante, nulle ou inconnue. Le même principe a été utilisé par plusieurs autres chercheurs

comme Chua et al. [58] et Cohen et al. [59]. Ces derniers ont quantifié l'effet d'un changement d'une variable sur les autres comme : faible, moyen ou élevé.

La caractérisation qualitative ne permet pas donc d'évaluer correctement un changement. En effet il est impossible de mesurer l'influence concrète d'un changement d'un paramètre donné sur les autres, d'où l'importance de la caractérisation quantitative. En fait, selon Kusiak et al. [57], elle permet de calculer le taux de changement. Certains auteurs [31][34][107][114][115] ont utilisé les termes *likelihood* et *impact* pour décrire la probabilité d'occurrence des changements et intégrer un changement dans le nœud cible. Cependant, ces valeurs sont données par des experts (concepteurs) en se basant sur leur expérience et leurs connaissances du produit. Par conséquent, nous avons pensé à caractériser quantitativement les changements qui touchent l'un des composants d'un assemblage CAO sans utiliser ni expertise ni donnée externe du modèle. Nous visons par la suite à caractériser quantitativement un changement en se basant uniquement aux données du modèle CAO extraites précédemment et des matrices créées dans la section préliminaire.

L'objectif principal est de déterminer un coefficient que nous avons appelé « coefficient d'influence » existant entre les composants d'un assemblage. Ainsi, l'extraction des données associées, la détermination du sous-ensemble et la création des arbres de propagation ne sont que des étapes préliminaires. Maintenant, après avoir construit la matrice d'interférence, la matrice de relations et la détermination des sous-ensembles existants, nous sommes capables de créer la matrice d'influence qui nous donne une idée claire sur le degré d'influence de chaque composant sur les autres composants de l'assemblage. Un composant est dit influençant un autre composant si une modification qui le concerne, a nécessairement un effet ou une conséquence sur le second et vice-versa. Une seule modification est facile à gérer, mais cette modification déclenchera une série de modifications consécutives, qui est par définition le phénomène de propagation du changement. Ainsi, afin d'en faciliter la gestion d'un changement et l'évaluation d'impact de sa propagation, vient l'idée de déterminer le degré d'influence d'une modification affectant un composant (bien déterminé) sur le reste des composants de l'assemblage. C'est la caractérisation quantitative du changement.

### **3.1.3.2. Coefficient d'influence au sein d'un assemblage**

Il est remarquable qu'une petite modification apportée par le concepteur, ait une grande influence sur un nombre important de pièces d'assemblage, que ces pièces proviennent du même sous-

ensemble ou non. De plus même si les deux composants n'ont pas de lien direct, la modification de l'un peut se propager vers l'autre (par l'intermédiaire des autres composants). À ce niveau-là, on distingue principalement deux types d'influence. Le premier type d'influence ne considère que l'influence d'un composant donné sur d'autres composants ayant un nombre de liens directe différent de zéro. Ce type d'influence est appelé influence directe. Cependant, si un composant a une influence sur d'autres modules de l'assemblage, non directement liées, on parle d'influence indirecte. Nous avons alors distingué deux phénomènes différents, il est donc compréhensible que la détermination du coefficient ou paramètre d'influence diffère d'un type à un autre. Néanmoins, dans la majorité des cas, un composant d'un ensemble qui subit une modification directe causée par l'un des composants qui lui sont directement liés, d'une part, subit également simultanément des modifications indirectes dues aux autres composants qui le connectent à la source de la modification, d'autre part. On parle dans ce cas d'influence mixte. La figure (Fig.3. 7) présente différentes séquences d'assemblage représenté par un graphe de liaison. Cette dernière montre bien les différents types d'influence qui peut exister au sein d'un assemblage. Pour le cas où le graphe se compose uniquement par deux sommets adjacents (deux sommets qui sont reliés par une arête), on parle dans cas ((Fig.3. 7(a)) de phénomène d'influence directe. C'est à dire que l'on peut passer de l'un à l'autre directement sans passer par d'autres sommets ou d'autre composant. Cependant, Lorsqu'il s'agit d'une chaîne (une suite de sommets adjacents) on est dans le cas d'influence indirecte car on doit forcément passer pour les autres nœuds intermédiaires (Fig.3. 7(b)).

Alors que lorsqu'il s'agit d'un cycle ou bien une chaîne fermée le phénomène d'influence mixte apparait. Étant donné qu'on peut trouver différents chemins qui nous permettent d'atteindre un seul composant (comme le montre l'arbre de propagation Fig.3. 6). Le troisième cas de la figure (Fig.3. 7 (c)) explique le fait que la pièce appelée  $P_3$  peut avoir simultanément une modification causée directement par la pièce  $P_1$  et une autre causé par la modification de  $P_2$  lui-même affecté par le changement qui a touché  $P_1$ . C'est à dire une modification qui affecte  $P_1$  peut avoir deux chemins de propagation différents afin d'atteindre  $P_3$ . Soit directement de  $P_1$  vers  $P_3$ , de  $P_1$  vers  $P_2$  et puis  $P_3$ .

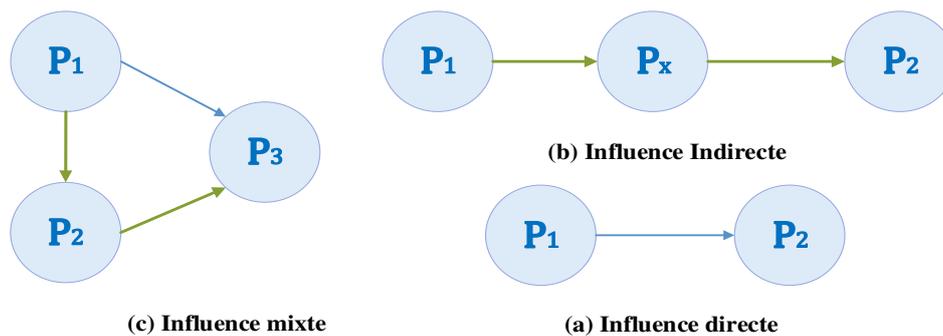


Fig.3. 7. Les différents types d'influence qui peuvent exister au sein d'un assemblage.

Nous avons défini ces différents types d'influence par analogie avec le modèle de Kusiak et al. [57] qui a distingué deux types d'inférences qualitatives et quantitatives : l'inférence parallèle et sérielle. Ces deux types d'inférences (parallèles et sérielles) sont fondés sur la théorie de l'algèbre qualitative définie par Kleer et al. [116].

### 3.1.3.2.1 Coefficient d'influence direct

Toutes les méthodes de propagation de changements présentées précédemment dans le chapitre 2 commencent par la caractérisation de changement, et la détermination de dépendance entre les éléments d'un système et finissent par analyser le phénomène de propagation des changements.

Nous nous concentrons à ce stade sur la détection des Coefficients d'Influence (CI) existant entre les pièces ayant des liens directs. Les modèles de dépendance sont exprimés grâce à un ensemble d'équations, de relations qualitatives ou quantitatives ou même de règles d'influence (Exemple : si la valeur d'un paramètre  $x$  égale à une valeur ou appartient à un intervalle alors un deuxième paramètre vaut une certaine valeur). Pour construire notre propre modèle de dépendance, on s'est principalement basé sur : les surfaces de contact, les volumes d'interférence et le nombre de relations entre deux composants donnés. De ce fait, trois facteurs différents sont utilisés afin de décrire la dépendance existante entre les deux pièces. Par conséquent pour obtenir une équation décrivant la dépendance totale trois coefficients de pondération sont intégrés.

Kusiak et al. [57] ont défini la dépendance quantitative entre deux variables  $a$  et  $b$  comme le produit de la variation du premier divisée par sa valeur initiale et la valeur initiale du deuxième divisée par sa variation comme le montre l'équation (3. 1) :

$$\psi_{ab} = \left(\frac{\Delta b}{b}\right)\left(\frac{a}{\Delta a}\right) \quad (3.1)$$

Notre modèle de dépendance se base sur la dépendance surfacique, la dépendance volumique et la dépendance relationnelle. En prend par la suite chaque facteur indépendamment des autres et en appliquent le principe de dépendance quantitative de Kusiak. Pour déterminer la dépendance existante entre deux variables « a » et « b » nous devons perturber la valeur initiale de « a » et mesurer ainsi la nouvelle valeur de « b ».

**Hypothèse 1 :** On suppose qu'une modification du composant  $C_i$  affectera toutes les zones de contact de ce dernier avec le composant  $C$

$$\psi_{ij}^S = \left(\frac{\Delta S_{ij}}{St_i}\right)\left(\frac{St_j}{\Delta S_{ji}}\right) \quad (3.2)$$

$\psi_{ij}^S$  C'est la dépendance surfacique, avec  $\Delta S_{ij} = St_i - Sc_{i,j}$  C'est la surface du composant  $C_i$  qui peut être influencée par le composant  $C_j$ . Elle est en fait, caractérisée par la surface engagée dans une interférence de coïncidence.  $St_{ij}$  Est la surface totale du composant  $C_i$

**Hypothèse 2 :** On suppose qu'une modification du composant  $C_i$  affectera toutes les zones d'interférence entre ce dernier et le composant  $C_j$ .

$$\psi_{ij}^V = \left(\frac{\Delta V_{ij}}{Vt_i}\right)\left(\frac{Vt_j}{\Delta V_{ij}}\right) \quad (3.3)$$

$\psi_{ij}^V$  C'est la dépendance volumique, avec  $\Delta V_{ij} = Vt_i - VI_{ij}$  ;  $VI_{ij}$  est le volume d'interférence entre composant  $C_i$  et le composant  $C_j$  et  $Vt_j$  est le volume total de la pièce  $j$ .

**Hypothèse 3 :** On suppose qu'une modification du composant  $C_i$  affectera toutes les relations qu'il a avec le composant  $C_j$ .

$$\psi_{ij}^{Nb} = \left(\frac{\Delta Nb_{ij}}{Nb_i}\right)\left(\frac{Nb_j}{\Delta Nb_{ji}}\right) \quad (3.4)$$

$\psi_{ij}^{Nb}$  C'est la dépendance relationnel, avec  $\Delta Nb_{ij} = Nb_i - Nb_{ij}$  ,  $Nb_i$  est le nombre des relations du composant  $C_i$  avec tous les composants et  $Nb_{ij}$  c'est le nombre des relations entre ce dernier et le composant  $C_j$ .

On obtient finalement  $CI_{ij}^D = \psi_{ij}$  le coefficient d'influence direct entre les deux composants  $C_i$  et  $C_j$  est donné par l'équation (3.7).  $\psi_{ij}$  C'est le terme qui exprime la dépendance général.

$$\psi_{ij} = \alpha \times \psi_{ij}^S + \beta \times \psi_{ij}^V + \gamma \times \psi_{ij}^{NB} \quad (3.5)$$

### 3.1.3.2.2 Coefficient d'influence indirecte

Un assemblage ou un sous-assemblage est un groupe de composants ou une collection de pièces assemblées en une seule unité, destiné à être utilisé dans la fabrication d'un ensemble plus grand. Il existe donc des nombreux liens entre les différents composants. Certains d'entre eux ont des liens directs et d'autres sont reliés à l'aide de composants intermédiaires. Ce facteur doit donc être pris en considération lors de la détermination Coefficient d'influence. Par conséquent, lors du calcul des coefficients d'influence entre deux composantes données  $C_i$  et  $C_j$ , il faut développer notre modèle de dépendance et prendre en considération la notion de l'influence cumulative ou cumulée (qui est bien en accord avec la notion de l'inférence sérielle).

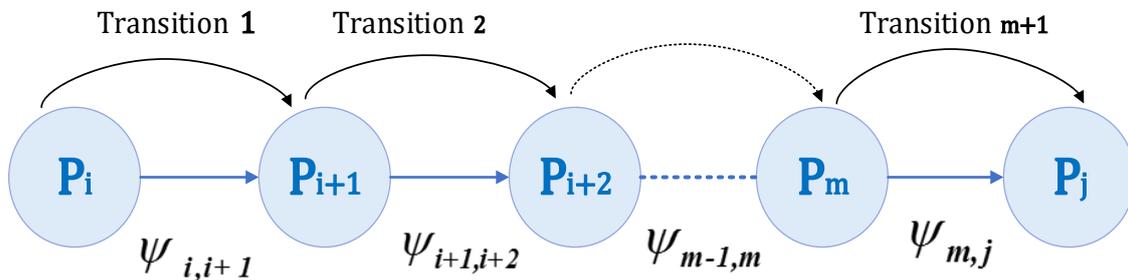


Fig.3. 8. Phénomène d'influence entre une Pièce i et Pièce j indirectement connectées.

$$\psi_{ij} = \psi_{i,i+1} \times \psi_{i+1,i+2} \times \dots \times \psi_{m-1,m} \times \psi_{m,j} \quad (3.6)$$

Deux composants  $C_i$  et  $C_j$  sont reliés par l'intermédiaire de  $m$  autres composants, signifie que l'influence de  $C_i$  atteint  $C_j$  après  $(m+1)$  transissions et à travers les  $m$  composants comme le montre la figure (Fig.3. 8). Les valeurs des coefficients d'influences indirectes sont déterminées en se basant sur les travaux de Kusiak et al. [57] et Yun et al. [117]. Conséquemment, l'effet cumulé se traduit par la multiplication de tous les coefficients d'influences directes cumulés, l'équation (3. 6) montre bien ce résultat. Le coefficient d'influence indirecte appelé  $CI_{ij}^{ID} = \psi_{ij}$ .

### 3.1.3.2.3 Coefficient d'influence mixte

La séquence d'assemblage montré par le graphe de la figure (Fig.3. 9) présente tous les cas d'influence possible qui peuvent être présent entre les composants d'un assemblage. D'abord les pièces  $P_i$  et  $P_m$  sont directement connectés, seul un coefficient d'influence directe  $\psi_{i,m}$  existe entre eux. Tant dis que  $P_{i+2}$  et  $P_i$  sont connecté par l'intermédiaire de  $P_{i+1}$ . On parle ici d'un coefficient d'influence indirecte qui est égale au produit de  $\psi_{i,i+1}$  et  $\psi_{i+1,i+2}$  (effet cumulative ou influence sérielle). Concernant le phénomène d'influence mixte, apparait lorsqu'il existe plusieurs chemins de propagation différent qui permet de passer du composant qui a déclenché la modification vers un autre composant. La figure (Fig.3. 9) montre une chaine fermée composée de  $P_i$ ,  $P_{i+1}$  et  $P_j$ . D'une Part, on peut passer de  $P_i$  à  $P_j$  directement (chemin vert dans la figure). D'autre part on peut passer par  $P_{i+1}$  afin d'atteindre  $P_j$  (chemin rouge dans la figure). C'est donc le cas de l'influence parallèle, pour calculer le coefficient d'influence entre les deux composants il suffit de sommer les coefficients des deux chemins.

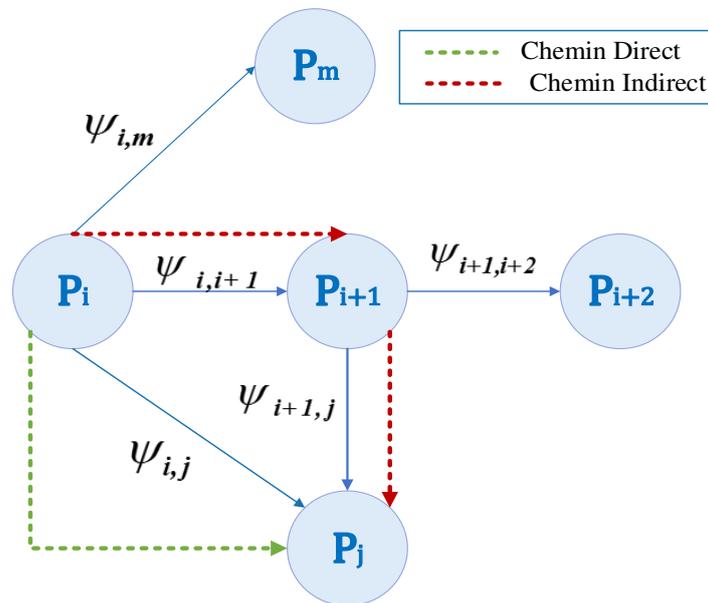


Fig.3. 9. Phénomène d'influence mixte au sein d'un assemblage.

$$\psi_{ij}^* = (\psi_{i,i+1} \times \psi_{i+1,j}) + \psi_{ij} \quad (3.7)$$

Le premier terme de l'équation (3. 7) correspond bien au coefficient d'influence indirecte entre  $P_i$  et  $P_i$  nommé  $CI_{ij}^{ID}$ . Tandis que le deuxième terme représente le coefficient d'influence directe  $CI_{ij}^D$ . La somme des deux et appelée donc coefficients d'influence mixte noté  $CI_{ij}^M$  avec  $CI_{ij}^M = \psi_{ij}^*$ .

### 3.1.3.3. Matrice des Influences

Ainsi, dans le but de définir les différents coefficients d'influence existant au sein d'un assemblage, une matrice dite Matrice d'influence est créée. C'est une matrice carrée de taille  $N$  (le nombre total des composants) mais non symétrique. En effet, le coefficient d'influence d'un composant  $C_i$  sur un deuxième composant  $C_j$  est différent du coefficient de celui-ci sur  $C_i$ . La matrice créée est appelé **IFM** (Influence Matrix).

$$IFM = \begin{bmatrix} CI_{11} & CI_{12} & \dots & CI_{1,n} \\ CI_{21} & CI_{22} & \dots & \dots \\ \vdots & \vdots & \dots & \dots \\ \vdots & \vdots & \ddots & CI_{n-1,n} \\ CI_{n1} & \dots & CI_{n,n-1} & CI_{n,n} \end{bmatrix} \quad (3. 8)$$

Avec  $CI_{ij} \neq CI_{ji}$ , L'interférence en général est définie comme la rencontre, l'assemblage ou la conjonction de deux phénomènes distincts et différents ou encore comme le phénomène résultant de la superposition de deux chaînes différentes. Ainsi, pour deux composants  $C_i$  et  $C_j$  la surface de la pièce  $C_j$  qui peut être influencé par toute modification affectant la pièce  $C_i$  est différent de la surface de la pièce  $C_i$  qui peut être influencée par toute variation affectant la pièce  $C_j$ . Ce phénomène n'est pas un phénomène réciproque.

Les arbres de propagations déterminées lors de la section précédente vont nous faciliter la tâche de la construction de cette matrice. En effet en utilisant l'arbre de propagation partant du nœud  $C_i$ , on peut facilement déterminer les coefficients d'influence existants entre  $C_i$  et tout le reste de l'assemblage. Pour un composant donné  $C_j$  il suffit de trouver tous les chemins possibles qui le relie avec  $C_j$ ; Calculer le coefficient d'influence total tout au long de chaque chemin en

multipliant les coefficients d'influence directe successifs ; puis finalement sommer ces différents coefficients obtenus.

On obtient finalement la Matrice d'influence qui permet de quantifier l'effet de chaque composant de l'assemblage sur le reste des composants. Ainsi, au cours de la phase ECO, si nous avons reçu un ordre de modification qui concerne une pièce donnée, il est possible de mesurer l'influence de cette proposition sur l'ensemble du modèle. Grâce à ces coefficients, nous pouvons déterminer le composant qui sera le plus touché. Si une légère modification déclenche une série de modifications importantes, il vaut mieux l'annuler ou en proposer une autre, qui a moins d'effets.

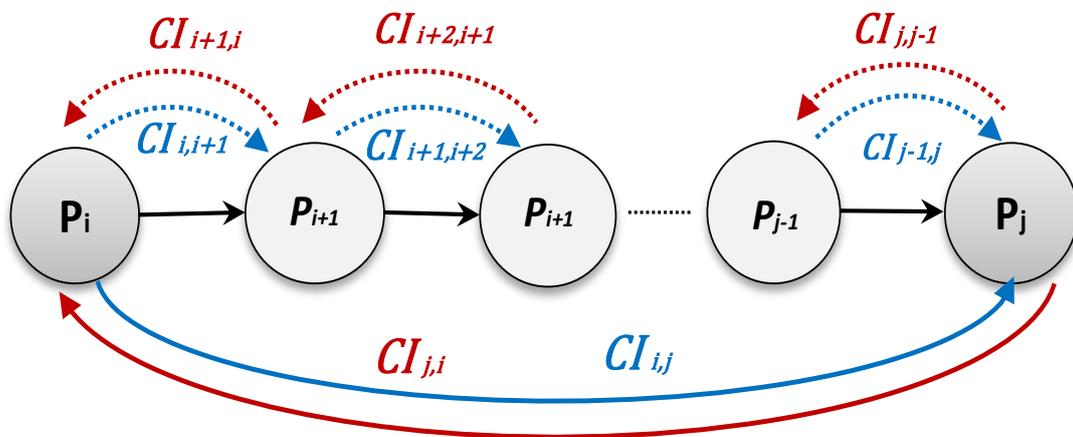


Fig.3. 10. La non-réciprocité du phénomène d'influence entre une Pièce i et Pièce j.

En guise de conclusion, la détermination des coefficients d'influence directe, indirecte et mixte est essentiellement fondée sur les surfaces influencées, les volumes d'interférence et le Nombre de relation. Les surfaces influencées représentent la surface de contact entre deux pièces données. Ces zones de contact sont déterminées par l'interférence de coïncidence. Tout comme, le volume d'interférence ces deux paramètres sont retenus directement à l'aide de l'add-in développée. Le nombre des liens existants entre deux composants est déterminé à partir de la matrice des relations. Cette dernière matrice nous permet de caractériser quantitativement un changement avant de le faire propager.

### 3.1.4. Probabilité de Propagation d'une modification

#### 3.1.4.1. Probabilité de Propagation directe

C'est bien évident qu'une modification qui touche l'un des composants d'assemblage d'un système CAO va se propager vers les autres composant qui sont reliés entre eux par des contraintes d'assemblage et ayant des liens et des interférences avec celui-là. La propagation dans ce cas est dite propagation directe puisque les composants sont directement liés. Ce phénomène est fortement lié aux relations qui existent entre les différents constituants du système, par conséquent nous allons considérer que la probabilité de propagation de modification entre deux composants est égale à la probabilité que le lien existant entre eux soit endommagé.

Par définition la Probabilité est la fonction permettant de « mesurer » la chance de réalisation d'un évènement de  $P(\Omega)$ . Soit  $(\Omega, A)$  un espace probabilisable. Une probabilité que  $(\Omega, A)$  est une application satisfaisant les 3 axiomes suivants :

$$0 \leq P(A) \leq 1, \quad \forall A \in \mathcal{A}$$

$$P(\Omega) = 1$$

$$P\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} P(A_i), \quad \forall (A_i)_{i \in I} \text{ ensemble dénombrable d'évènements disjoints}$$

Dès lors que P est définie,  $(\Omega, A, P)$  s'appelle un espace probabilisé. Lorsqu'il n'y a pas lieu d'attacher aux différents évènements élémentaires des probabilités différentes, on a pour tout  $\omega_i$ ,  $p_i = p$ . On dit que l'univers est équiprobable. Lorsque l'univers est fini, de cardinal  $|\Omega|$ , on a  $p = 1/|\Omega|$ . On définit alors la probabilité P comme précédemment : soit A un évènement quelconque.

$$P(A) = \frac{|A|}{|\Omega|} \quad \text{Cette probabilité est appelée probabilité uniforme sur } \Omega$$

Dans notre cas on a un système de N composants ayant un nombre bien défini de liens que les relient comme le montre les matrices des Relations et des Interférence. On définit,  $(\Omega, A, P)$  un espace probabilisé avec : La Probabilité P(A) est la fonction permettant de « mesurer » la chance de propagation d'une modification d'un Composant  $C_i$  vers un autre composant  $C_j$ . On considère qu'une modification d'un Composant  $C_i$  donnée s'est propagée vers un composant  $C_j$  si seulement si l'un des liens que les réunissent est touché.

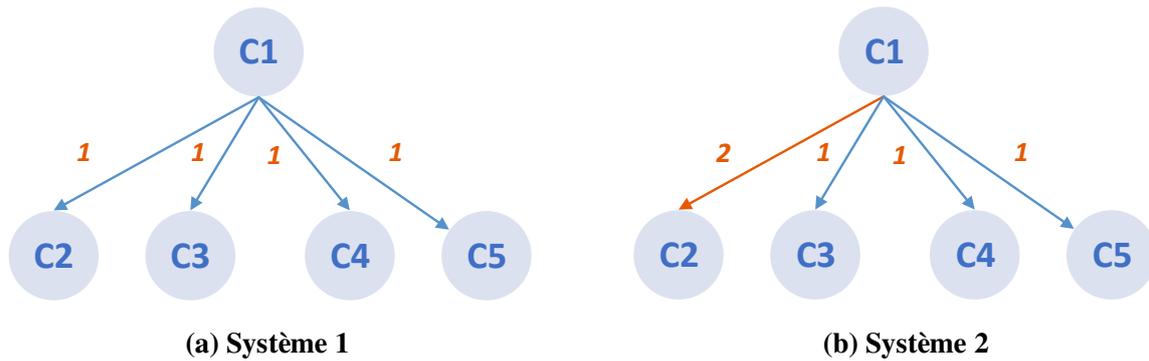


Fig.3. 11. Exemple de système avec différents nombres de liens.

Soit un système  $\Omega$  composé par 5 composants :  $C_1, C_2, C_3, C_4, C_5$ . Le Composant  $C_1$  est relié avec tous les autres composants du système avec une seule relation (Fig.3. 11. (a)) Supposons que  $C_1$  change quelle est la probabilité que les autres composants soient affectés ? En d'autres termes quelle est la probabilité de propagation de la modification vers chacun des composants ?

- $A_i$  : La modification se propage vers le composant  $C_i ; 1 < i \leq 5$
- En utilisant la loi uniforme on obtient :

$$P(A_2) = P(A_3) = P(A_4) = P(A_5) = \frac{1}{4} \quad (3.9)$$

- En appliquant la même loi uniforme et en supposant que  $C_1$  a un lien double avec  $C_2$  (Fig.3. 11. (b)) on obtient :

$$P(A_2) = \frac{2}{5}; P(A_3) = P(A_4) = P(A_5) = \frac{1}{5} \quad (3.10)$$

Donc dans le cas général, Supposons qu'un composant  $C_i$ , change la probabilité de propagation de la modification vers un composant  $C_j$  est donnée par :

$$P(A_j) = \frac{Nbr_{ij}}{\sum_{x=1}^{x=n} Nbr_{ix}} \quad \text{avec: } Nbr_{ij} \text{ C'est le nombre de liens entre le composant } C_i \text{ et } C_j. \quad (3.11)$$

### 3.1.4.2. Propagation indirecte

La modification c'est alors propagé vers les composants directement liés à celui qui a été touché en premier appelé « initiateur ». Les composants qui ont été touché cette propagation forme la première couche (même principe que pour l'arbre de propagation). Cependant, la propagation ne

s'arrête pas là, étant donné que les constituants de la première couche ont eux-mêmes des liens avec d'autres composants. Par conséquent la propagation continue et elle est appelée propagation indirecte car les composants affectés qui forment la deuxième couche ne sont pas directement liés à l'initiateur. Ce phénomène continue jusqu'à atteindre tous les composants du système qui peuvent être affectés.

Afin de calculer la probabilité de propagation de la modification aux différentes couches des composants du système on a recours à la création d'un arbre de probabilités ou arbre pondéré qui permet de décrire ce phénomène. Les données de l'arbre pondéré vont nous servir par la suite pour déterminer les différentes probabilités.

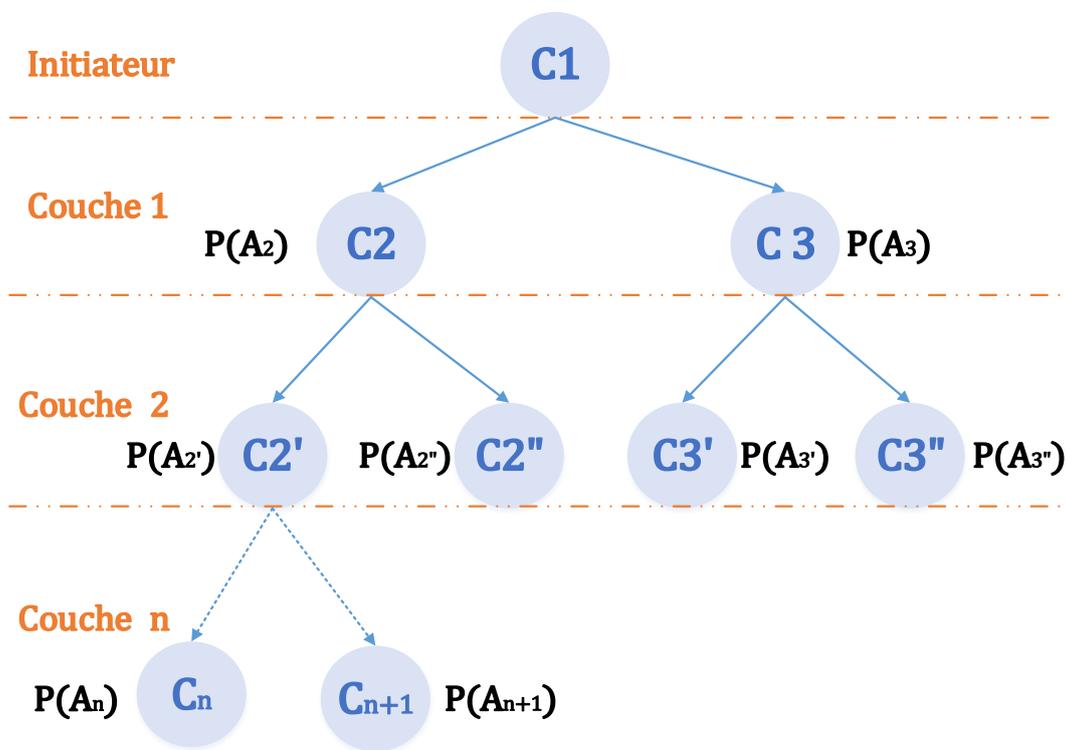


Fig.3. 12. Arbre de probabilités de propagation.

Pour décrire un arbre à plusieurs couches, on va tout simplement rajouter des branches, aux extrémités de celles existant déjà. D'une part l'arbre se lit et se construit en partant de sa racine. Les branches partant de la racine sont dites primaires et mènent à des nœuds. Les branches reliant deux nœuds sont dites secondaires. Un chemin partant de la racine pour relier un nœud est appelé un trajet. D'autre part il faut noter que sur les branches secondaires, on lit toujours une probabilité conditionnelle. Aussi un arbre pondéré vérifie les propriétés suivantes :

- Le poids d'une branche primaire indique la probabilité de l'évènement correspondant.
- Le poids d'une branche secondaire est la probabilité conditionnelle de l'évènement qui se trouve à son extrémité sachant que le trajet menant à son origine a été réalisé.
- La somme des probabilités inscrites sur les branches issues d'un même nœud est toujours égale à 1.
- La probabilité d'un trajet est égale au produit des poids des branches le constituant.

Par conséquent la probabilité que la modification se propage vers le composant  $C_3$ , après avoir passé par le composant  $C_3$  est égale au produit de  $P(A_3)$  et  $P(A_3')$ .

En utilisant la loi de probabilité définie dans l'équation (3. 11) et le principe de l'arbre pondéré, on peut déterminer une matrice de probabilité de propagation. Il suffit de prendre à chaque fois pour la colonne  $i$  le composant  $C_i$  comme initiateur et on calcule la probabilité de propagation de la modification qui le touche vers le reste des composants du système. Et ainsi de suite jusqu'à atteindre les  $n$  composants du système. On obtient ainsi une matrice carrée de taille  $N$  mais cette dernière n'est pas symétrique car la probabilité qu'une modification qui a touché  $C_i$  se propage vers  $C_j$  n'est pas égale à la probabilité qu'une modification qui a touché  $C_j$  se propage vers  $C_i$ . En effet la probabilité calculée ne dépend pas uniquement des liens existants entre les deux composants, mais aussi de l'ensemble des liens de chacun des composants avec le reste du modèle.

$$\begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 \vdots \\
 C_n
 \end{array}
 \begin{array}{c}
 C_{int} \\
 \left[ \begin{array}{c}
 P_{int \rightarrow 1} \\
 P_{int \rightarrow 2} \\
 P_{int \rightarrow 3} \\
 \vdots \\
 P_{int \rightarrow n}
 \end{array} \right]
 \end{array}$$

$C_{int}$  c'est le composant qui a déclenché la propagation ou "l'initiateur"  
 $P_{int \rightarrow i}$  c'est la probabilité qu'une modification qui à toucher le composant  $int$   
 se propage vers le composant  $i$

$$\begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 \vdots \\
 C_n
 \end{array}
 \begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 \vdots \\
 C_n
 \end{array}
 \begin{array}{c}
 C_3 \\
 \dots\dots\dots
 \end{array}
 \begin{array}{c}
 C_n
 \end{array}
 = \mathbf{P}$$

Fig.3. 13. Construction de la matrice de Probabilité.

$$\mathbf{P} = \begin{pmatrix} 0 & P_{2 \rightarrow 1} & \dots & P_{n \rightarrow 1} \\ P_{1 \rightarrow 2} & 0 & \dots & P_{n \rightarrow 2} \\ P_{1 \rightarrow 3} & P_{2 \rightarrow 3} & \dots & P_{n \rightarrow 3} \\ \vdots & \vdots & \ddots & \vdots \\ P_{1 \rightarrow n} & P_{2 \rightarrow n} & \dots & 0 \end{pmatrix} \quad (3.12)$$

### 3.1.5. Risque de Propagation d'une modification

Le risque de propagation du changement est défini par Yun et al.[117] comme un produit de la probabilité de propagation du changement et de l'impact du changement. La probabilité de propagation d'un changement est la mesure selon laquelle un changement de conception d'un composant entraînera directement un changement de conception d'un autre composant par propagation. Alors que, l'impact du changement est la probabilité que le composant affecté soit reconçu si le changement se propage vers celui-ci.

Également pour Clarkson et al. [69] le degré de propagation du changement prévue entre les sous-systèmes est mesuré comme un coût probabiliste, ou risque, où le risque est défini comme le produit de la probabilité que le changement se produise et de l'impact, ou coût, du changement. Par analogie à ces deux dernières définitions on a défini le risque de propagation d'une modification comme le produit de la probabilité de propagation d'un changement proposé avec les coefficients d'influences qui donne estimation de taux de modification d'un composant si le changement se propage vers lui. Ainsi le produit exprime un risque de modification qui mesure la probabilité qu'un composant ait affecté si un changement initial le touche, c'est le risque d'endommagement ou d'affection. Conséquemment, pour trouver la matrice de risque d'un assemblage il suffit de multiplier les deux matrices de probabilité et d'influence (Fig.3. 14).

$$\begin{array}{ccc}
 \begin{array}{c} \text{IFM} \\ \left( \begin{array}{cccc} 1 & CI_{2,1} & \dots & CI_{n,1} \\ CI_{1,2} & 1 & \dots & CI_{n,2} \\ CI_{1,3} & CI_{2,3} & \dots & CI_{n,3} \\ \vdots & \vdots & \ddots & \vdots \\ CI_{1,n} & CI_{2,n} & \dots & 1 \end{array} \right) \end{array} & \times & \begin{array}{c} \text{P} \\ \left( \begin{array}{cccc} 0 & P_{2 \rightarrow 1} & \dots & P_{n \rightarrow 1} \\ P_{1 \rightarrow 2} & 0 & \dots & P_{n \rightarrow 2} \\ P_{1 \rightarrow 3} & P_{2 \rightarrow 3} & \dots & P_{n \rightarrow 3} \\ \vdots & \vdots & \ddots & \vdots \\ P_{1 \rightarrow n} & P_{2 \rightarrow n} & \dots & 0 \end{array} \right) \end{array} \\
 \text{Matrice d'influence} & & \text{Matrice de probabilité}
 \end{array}
 =
 \begin{array}{c} \text{R} \\ \left( \begin{array}{cccc} 0 & R_{2 \rightarrow 1} & \dots & R_{n \rightarrow 1} \\ R_{1 \rightarrow 2} & 0 & \dots & R_{n \rightarrow 2} \\ R_{1 \rightarrow 3} & R_{2 \rightarrow 3} & \dots & R_{n \rightarrow 3} \\ \vdots & \vdots & \ddots & \vdots \\ R_{1 \rightarrow n} & R_{2 \rightarrow n} & \dots & 0 \end{array} \right) \\
 \text{Matrice de risque}
 \end{array}$$

Fig.3. 14. Matrice de Risque de Propagation d'une modification proposée.

### 3.2. Algorithme de Comparaison

#### 3.2.1 Importance de l'outil de comparaison

D'une part, les nouveaux produits sont rarement conçus à partir de zéro. Un des moyens les plus utilisés et les plus simples pour créer des nouvelles pièces dans le domaine mécanique est de modifier des pièces existantes qui lui ressemblent. La réutilisation des ressources de conception existantes permet également d'optimiser la qualité de la conception et d'améliorer son efficacité. Parce que les modèles récents peuvent parfois contenir beaucoup d'expérience et de connaissances. Il serait donc intéressant d'avoir un algorithme efficace de comparaison de similarité qui soit utilisé en tout temps dans le processus de conception. D'autre part, certains travaux connexes, estiment que plus des deux tiers des activités de conception impliquent une modification de la conception, une variation du design ou une conception sur la base de cas.

Il s'avère donc que les modifications de la conception sont devenues un mal nécessaire, En travaillant sur la propagation des changements de conception, nous avons perçu que la comparaison des pièces initiale/modifiée est une étape fondamentale. Cette étape permet l'identification (avec exactitude) des Features et des entités affectées. C'est une étape importante qui doit être nécessairement exécutée avant de passer à la propagation des changements.

L'approche développée est basée sur l'extraction et la structuration des données du modèle CAO à l'aide de l'API SolidWorks, par la suite les Features et les entités affectées par une modification vont être énumérées, et enfin on calculera le taux de modification apporté à la pièce en question. Ainsi notre outil de comparaison des pièces CAO développé par la suite peut être utilisé dans le

cadre de la propagation des modifications et peut aussi être utilisé comme un outil d'aide à la conception.

### 3.2.2 Algorithme générale de la comparaison de deux pièces CAO

Les environnements de conception assistée par ordinateur (CAO) sont principalement basés sur des modèles solides, car ils permettent une meilleure description de la géométrie et de la forme des pièces conçues et produites. Par la suite, leur omniprésence croissante a mis en évidence de nouveaux problèmes, tels que la détection des modifications, la gestion du changement et la propagation du changement. En fait, la détection et la propagation du changement sont bien liées.

L'idée est de créer un modèle permettant au concepteur de comparer deux versions différentes d'un modèle CAO. L'approche proposée se compose principalement de quatre grandes étapes, comme le montre la figure (Fig.3. 15). L'étape initiale consiste à charger les deux modèles que nous voulons comparer. La première étape consiste à extraire les données. Cette étape est fondamentale. Le modèle est principalement basé sur « les Features de conception », existant dans la Feature Manager du modèle CAO. On extrait tous les Features existantes ainsi que tous leurs attributs. Les données extraites doivent être sauvegardées dans une base de données, car elles seront utilisées ultérieurement. Après avoir recueilli les données nécessaires, il est temps de procéder à la comparaison (la deuxième étape de l'algorithme). Trois types de Features sont alors disponibles : Features communes, nouvelles Features et Features supprimées. En fait, ces Features devraient être transformées en éléments plus significatifs. Afin de donner plus de précision et de signification au taux de modification calculé par la suite. Par conséquent, pour chacune de ces Features considérées, on passe à la reconnaissance des faces affectées (la troisième étape). À ce stade, les caractéristiques de chaque face sont extraites et stockées dans la base de données utilisée précédemment. Enfin, on est désormais capable de calculer le taux de modification souhaité (sur la base des données enregistrées).

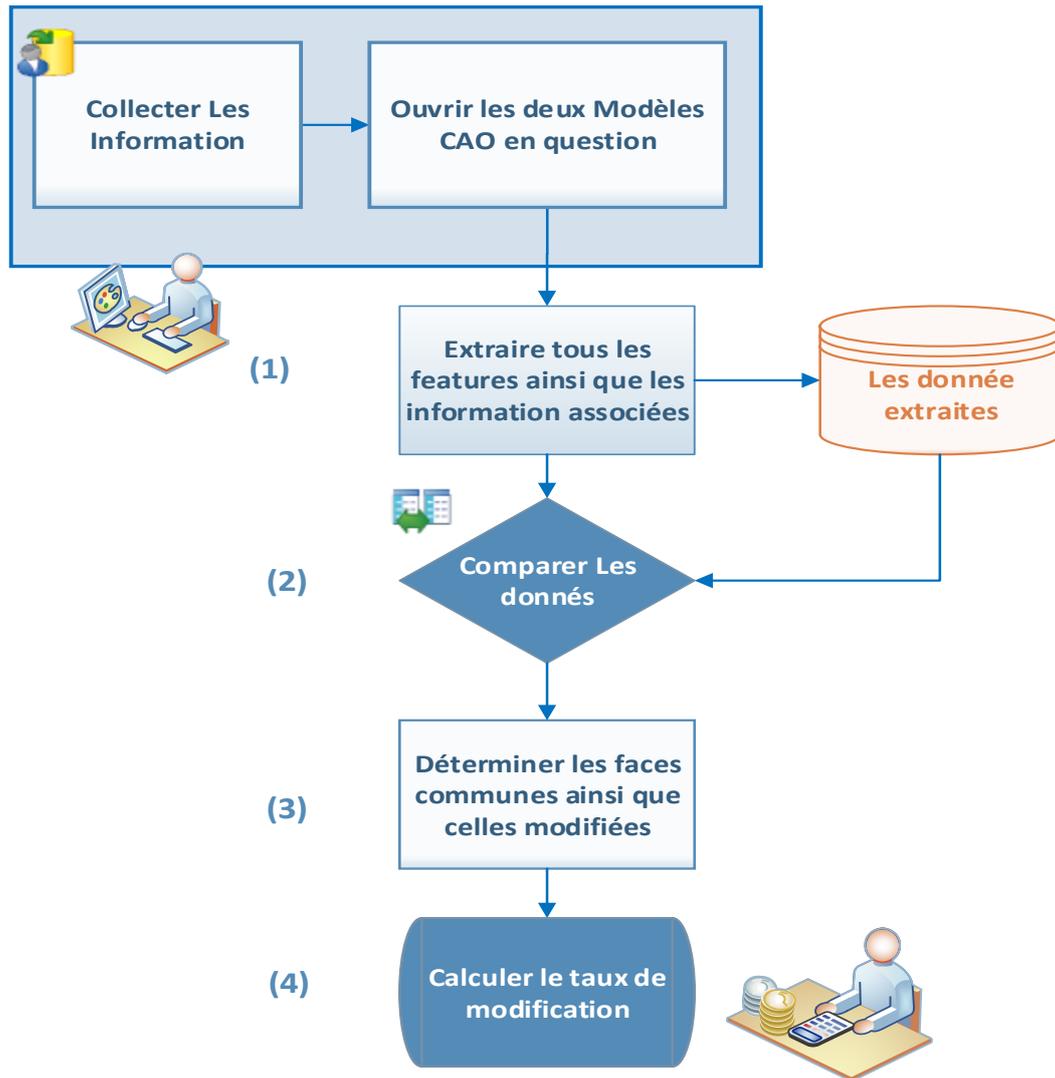


Fig.3. 15. Algorithme général de Comparaison.

### 3.2.2.1 Extraction des Features

L'arbre de création du gestionnaire de fonctionnalités fournit une vue d'ensemble du modèle CAO actif. Par conséquent, si on veut se focaliser sur les détails des esquisses et des Features, il faut y penser. En effet, toutes les données relatives aux Features et aux dimensions supplémentaires pour représenter le modèle CAO sont entièrement définies par le concepteur pendant la phase de développement. Toutes ces informations sont stockées de manière séquentielle dans l'arbre de création.

Notre objectif est d'extraire les données du modèle de l'arbre de création du Feature Manager. SolidWorks API (Application Programming Interface) contient des fonctions, des routines, des protocoles et des outils à relier à l'arbre de création Feature Manager. Le modèle développé exploite les fonctions existantes qui permettent de se connecter à ce Feature Manager et d'extraire les informations souhaitées. Il est bien connu que dans SolidWorks Feature Manager, toutes les informations sont regroupées sous des fonctionnalités ou des dossiers. En vérifiant ces derniers, la totalité des informations sur les Features, les esquisses (sketch) ainsi que les caractéristiques des faces ou des plans sont extraites. Plusieurs étapes doivent être franchies. Ces étapes comprennent la vérification des sous-fonctions ou « sub-Features », la comparaison des faces et l'extraction des dimensions. Pour chaque Feature on doit accéder au type, au nom et aux données de paramètres de l'élément ainsi qu'à l'élément suivant dans l'arbre de création. Ces informations importantes sont stockées et cette action est répétée jusqu'à ce que l'objet qui pointe vers la fonction suivante soit nul. Cette procédure est refaite (deux fois), la première pour extraire la version initiale des informations du modèle CAO tandis que la seconde pour la version modifiée du modèle. Nous obtenons finalement deux listes de Features : la première liste de Features de la première version et la deuxième liste de Features de la deuxième version (modifiée). En outre, leurs caractéristiques, dimensions et types sont sauvegardés.

### **3.2.2.2 Comparaison des données extraites**

Dès que les données recherchées sont obtenues, la comparaison peut être faite pour détecter les entités affectées et les modifications qui ont eu lieu. Dans un premier temps, trois listes de fonctions sont établies : Les Fonctions communes, les Nouvelles Fonctions et les Fonctions Supprimées. Certes, les Nouvelles fonctions sont des fonctions qui n'existent que dans la version modifiée de la pièce, c'est-à-dire qu'elles ont été ajoutées récemment, mais elles n'existent pas initialement (Avant modification). De même, les éléments supprimés sont des éléments qui existent initialement, mais qui ont été complètement effacés et qui n'existent pas dans la Pièce modifiée. Les fonctions communes sont celles qui existent initialement dans le modèle CAO et qui sont conservées même après la modification. Dans un premier temps, la comparaison est basée uniquement sur les types et les noms de fonctions. Ainsi, si une fonction ayant le même nom et le même type existe dans les deux pièces (initiale et modifiée), celle-ci est considérée comme une fonction commune. Toutefois, les dimensions de ces caractéristiques communes ne sont pas nécessairement les mêmes. Par conséquent, l'étape suivante est la comparaison des dimensions.

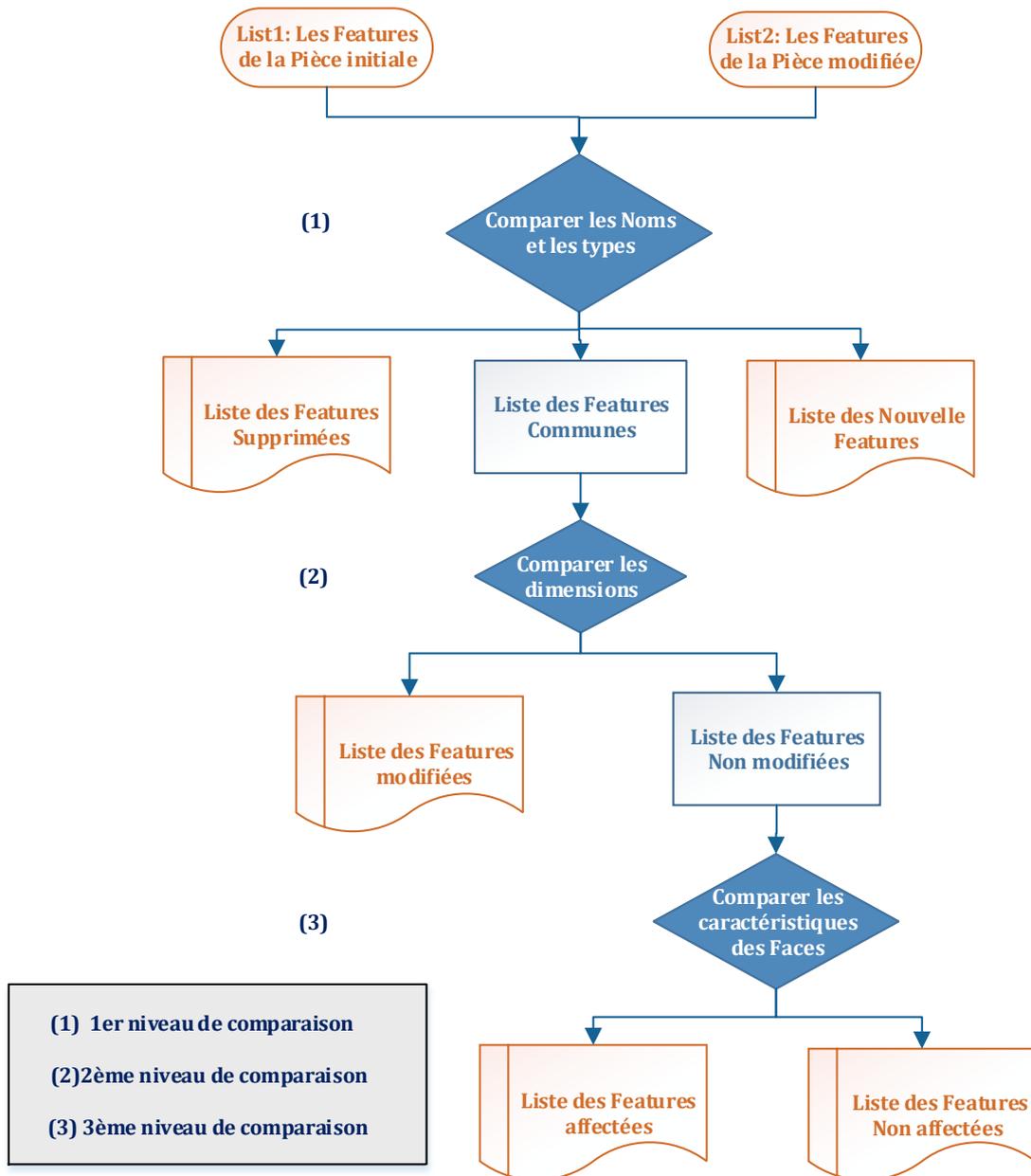


Fig.3. 16. Algorithme de comparaison des Features.

Comme on a initialement sauvegardé toutes les dimensions des fonctions, à ce stade, on peut accéder à la liste des dimensions des fonctions communes et on peut les comparer afin de détecter les caractéristiques révisées. Si nous détectons donc une dimension différente, la fonction correspondante est automatiquement considérée comme directement affectée ou modifiée. À la fin de cette comparaison dimensionnelle, la liste des fonctions communes sera divisée en deux blocs : les fonctions directement modifiées et les fonctions non modifiées. Il y a certaines fonctions qui ne sont pas réellement modifiées, mais leurs entités associées sont affectées. Ainsi,

la comparaison de troisième niveau est une comparaison basée sur les faces, à ce niveau nous allons comparer les faces des fonctions non modifiées une par une en fonction du nombre des arêtes et de la surface de chaque face.

Si nous détectons une modification, la fonction correspondante est ajoutée à la liste des fonctions indirectement affectées, plus de détails sont présentés à la Figure (Fig.3. 16). À la fin de cette étape, on obtient quatre listes :

- (a) Liste des Features supprimées ;
- (b) Liste des nouvelles Features ;
- (c) Liste des Features directement affectées (modifiées) ;
- (d) Liste des Features indirectement affectées ;

Il est à noter que pour chaque fonction, certaines propriétés sont également sauvegardées, comme le nombre de faces liées, la surface de chaque face, le nombre des arêtes, et les dimensions. Ces données seront utiles plus tard pour la détermination du rapport de modification.

### 3.2.2.3 Reconnaissance des faces

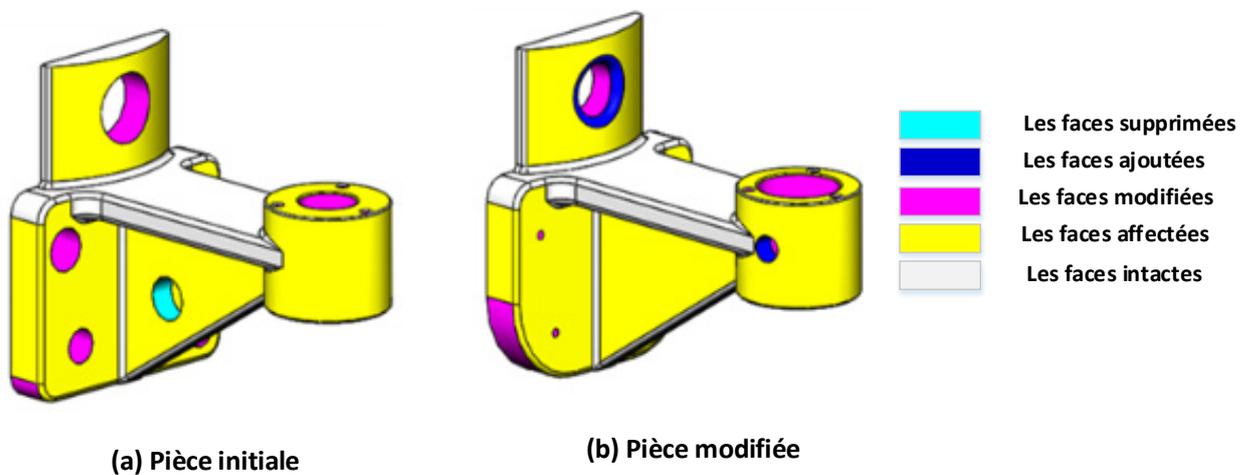


Fig.3. 17. Reconnaissance des faces.

Cette étape est une étape complémentaire pour aider le concepteur à mieux visualiser les entités affectées. Par conséquent, il nous suffit d'utiliser les données précédemment enregistrées. Pour chaque liste de fonctions des listes obtenues lors de l'étape précédente (a, b, c et d). On extrait les faces associées à chaque Feature puis elles sont colorées en fonction du type de fonctions comme indiqué dans la figure suivante (Fig.3. 17).

### 3.3. Algorithme de Réconciliation

#### 3.3.1 Mise en œuvre des associations initiales

Le terme association est un terme générique qui a déjà été défini dans le premier chapitre et qui exprime une dépendance entre des éléments ou des entités. De plus, on l'a déjà mentionné que le terme association désigne le niveau de connaissance le plus élevé, tandis que le mot contraint désigne le niveau de connaissance le plus bas. En conséquence, nous partirons du niveau le plus faible (contraintes d'assemblage) jusqu'au atteindre le niveau le plus haut et établir des associations entre les différentes Features des différents composants de l'assemblage.



Fig.3. 18. Exemple des listes de Features associées.

En fait les contraintes d'assemblages utilisés sont généralement des contraintes standards qui relient deux entités géométriques ou topologiques. À partir de ces contraintes là des associations entre les Features vont être déterminées. Les données de contraintes extraites sont bien évidemment enregistrés dans la base de données. Ces données-là sont la base sur laquelle l'algorithme de détermination des associations de Features est fondé. En effet pour chaque entité engagée dans une contrainte on a pu déterminer la Feature parent ou bien la Feature associée. Ensuite les deux entités de contraintes associées vont donner deux Features associées et ainsi de suite jusqu'à terminer toutes les contraintes. On obtient ainsi une liste de Features associées deux à deux. En conséquence, une méthode de tri est utilisée afin de former une liste de Features associées. Car une Feature peut avoir plus qu'une seule autre Feature associée à elle, ainsi on obtient des listes de Feature associées. Chaque liste contient ainsi les Features reliés entre eux et ayant des interactions et des influences les unes sur les autres. Par conséquent, les modifications apportées à l'une des deux Features qui appartiennent à une même liste ou qui sont "associées" doivent nécessairement se propager à l'autre.

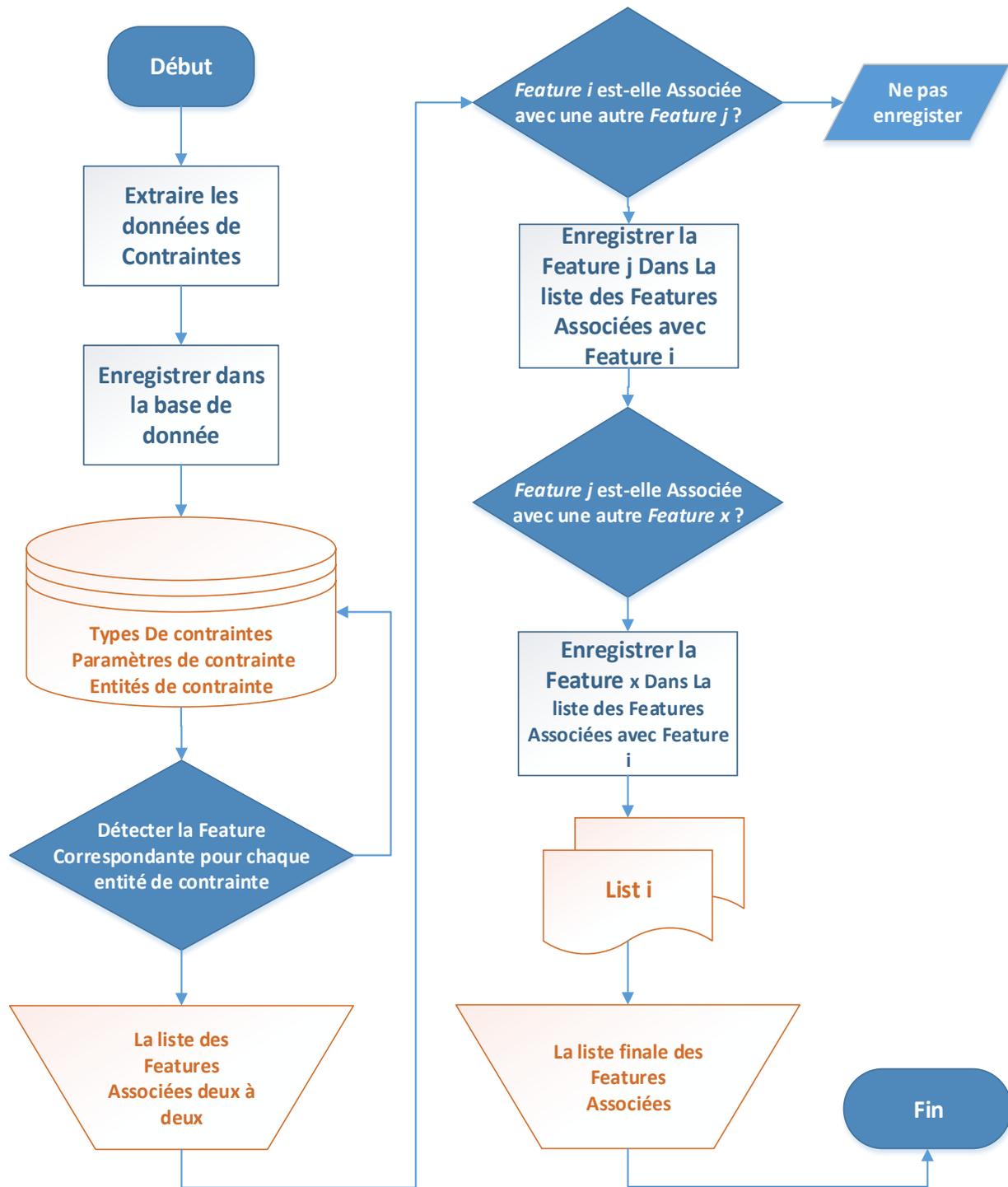


Fig.3. 19. Algorithme de détermination des Associations de Features.

### 3.3.2 Mise en œuvre des correspondances

Comme on a supposé dès le début que dans le cadre de ce travail l'une des pièces de l'assemblage va être extraite, modifiée puis réinsérer de nouveau dans l'assemblage et comme on a vu dans la partie comparaison que certaines Features vont être affectées. En effet il y a des Features qui vont

être affectées ou modifiées tandis qu'il 'y d'autres qui vont être supprimés et d'autres qui sont nouvellement créés. De ce fait, les listes des Features associées récemment créés doivent être aussi modifiées. Pour ce faire des correspondances entre les Features initiales et celles modifiées de la pièce qui a subi la modification doivent être faites. Car les Features récemment créés n'ont pas des Features associées à elles dans les listes de la partie précédente. D'autre part il 'y a d'autres qui ont été supprimées et qui doivent être supprimé ou bien remplacées par leur équivalent.

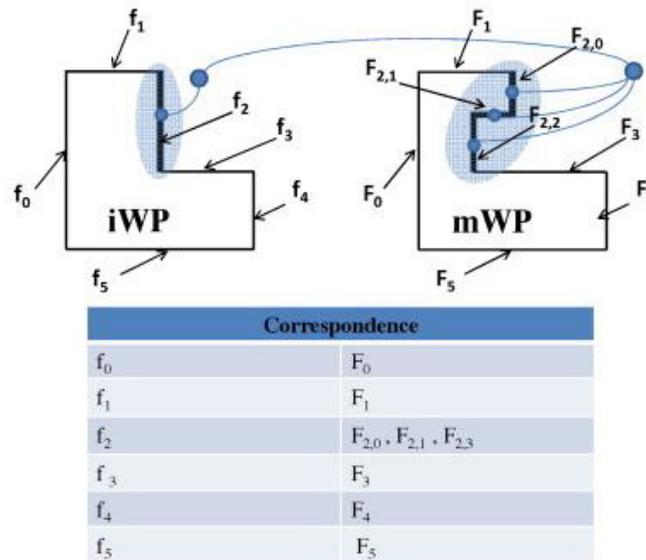


Fig.3. 20. Le principe de correspondances utilisées par Louhichi et al.[2].

De ce fait, pour résoudre ce problème, des correspondances doivent être faites. Le point de départ dans cette partie est le travail de Louhichi et al. [2] qui effectuent des correspondances entre le Work Package initial « iWP » et le Work Package modifié « mWP » en associant, en général, m entités du l'iWP à n entités du mWP. Les entités en question sont des entités topologiques et généralement des faces (Fig.3. 20)

Nous allons appliquer le même principe mais pour des Features. Toutes en reliant avec l'algorithme de comparaison qui a déjà détecté. Les Features créés récemment, les Features supprimées et aussi celles qui ont été modifiées, nous procédons à la détermination des correspondances possibles qui peuvent être mise en œuvre (Fig.3. 21).

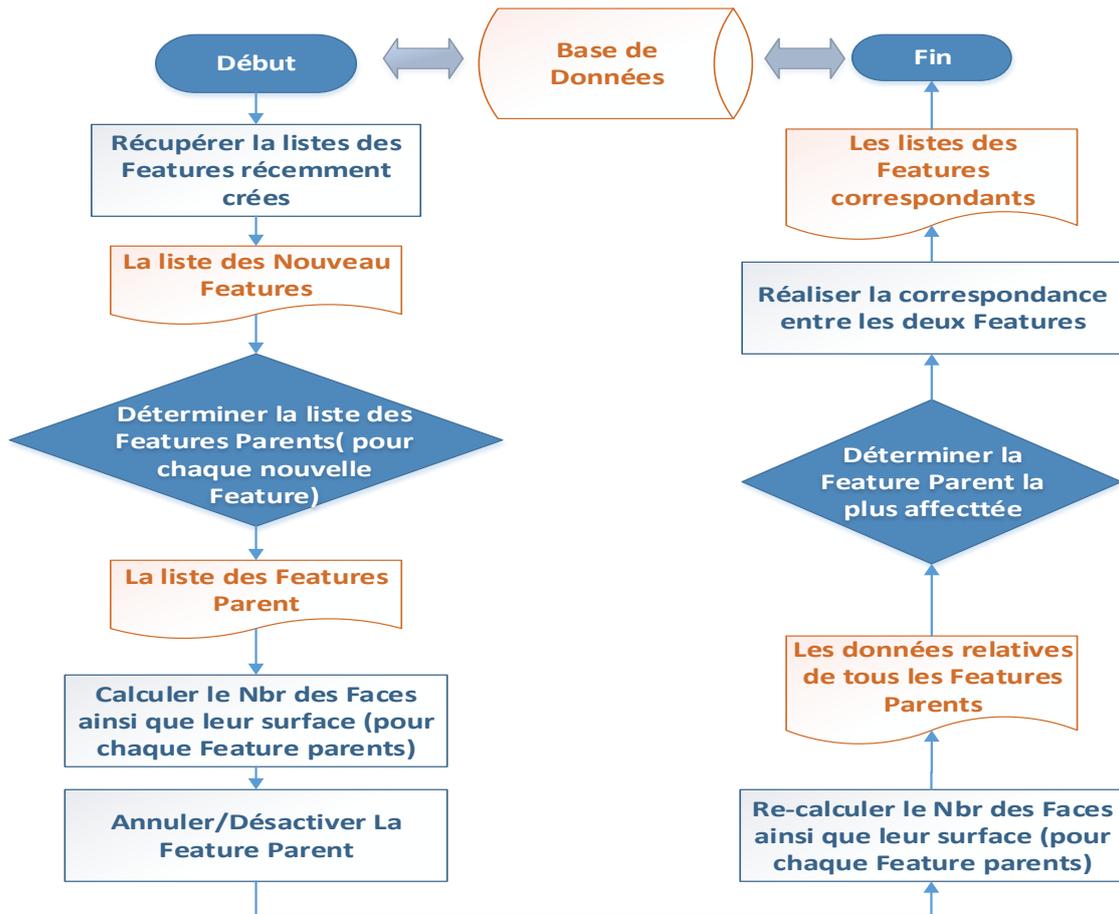


Fig.3. 21. Algorithme de détermination de correspondances entre les Features.

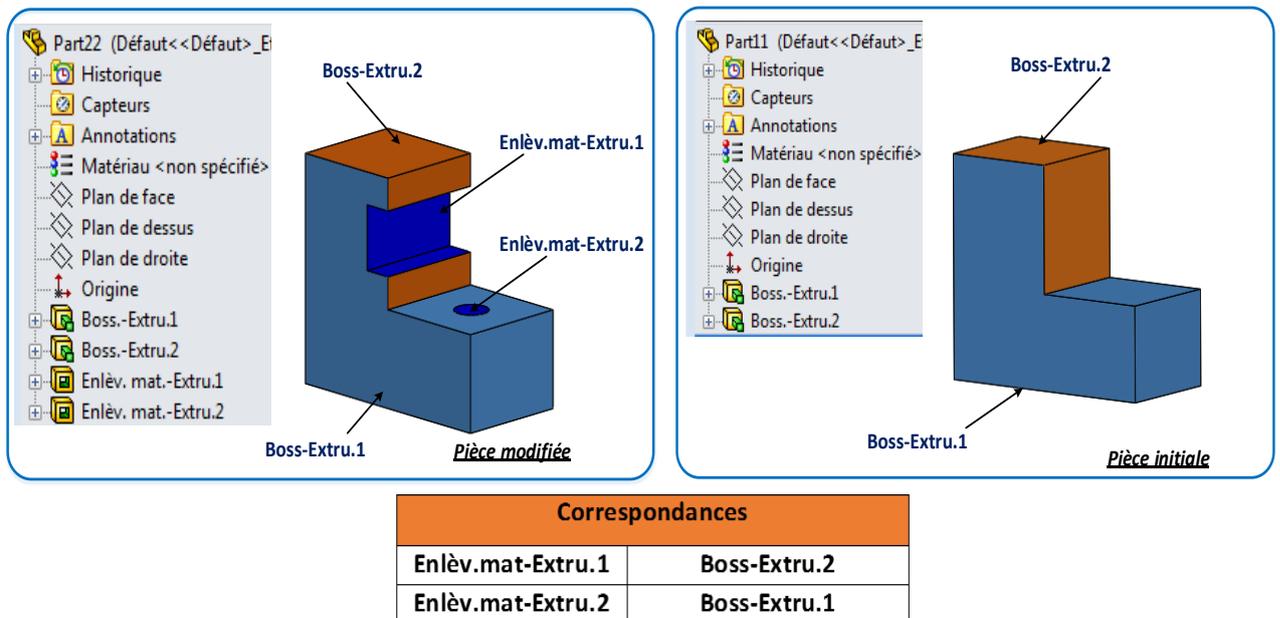


Fig.3. 22.Principe de correspondance entre les Features.

### 3.3.3 Réconciliations des associations

Nous avons déjà conçu un modèle de mise en œuvre des associations au sein du modèle CAO basé sur les contraintes d'assemblage. Ceci par la création des liens qui vont nous servir par la suite dans le processus industriel de gestion des changements. Le modèle met en évidence les informations qui nous permettent d'évaluer une modification proposée. Ces informations nous ont permis de créer des correspondances entre les Features du modèle initial et celles du modèle modifié. Nous disposons ainsi des *listes de Features associées* ainsi que les *listes des Features correspondantes*.

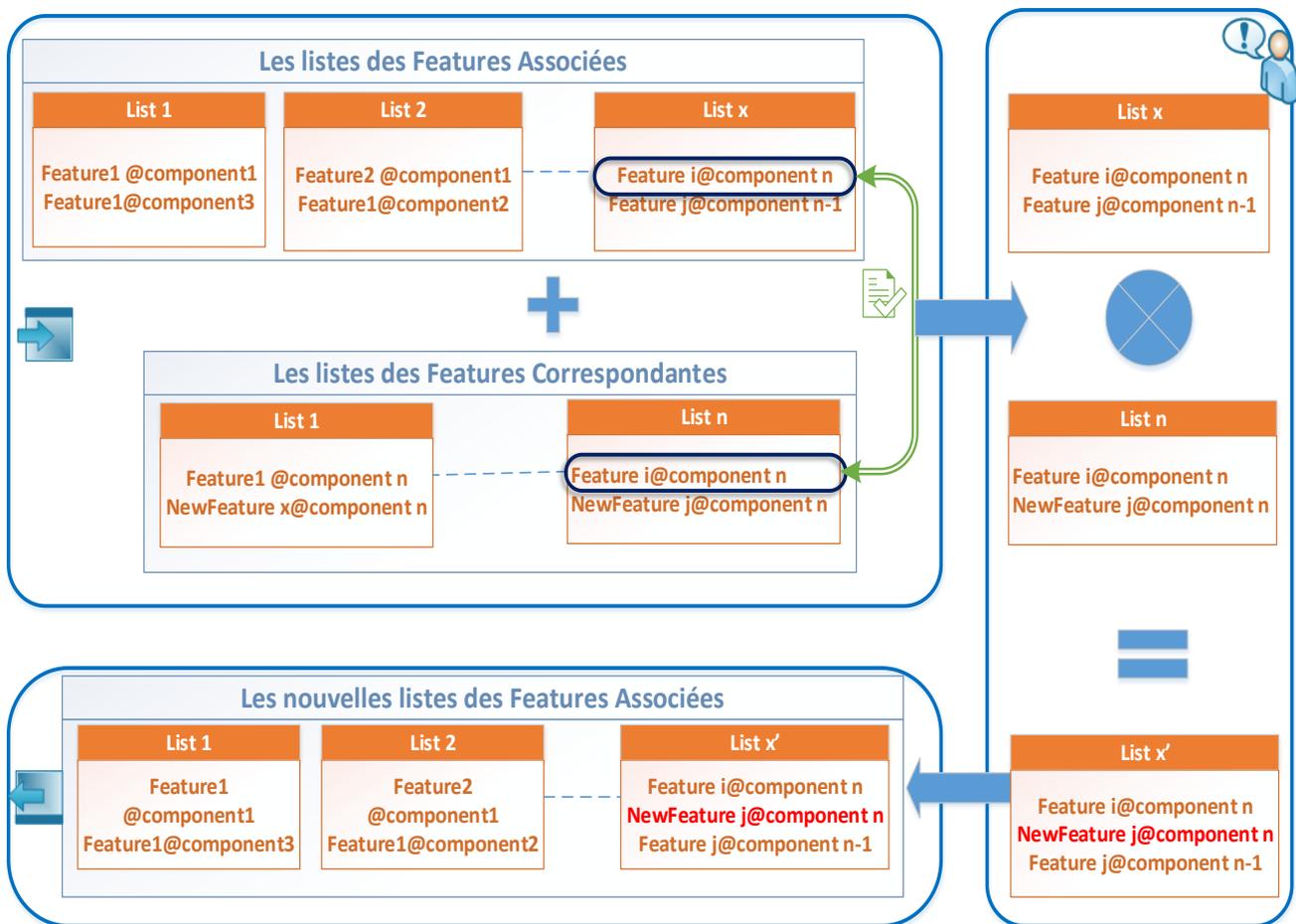


Fig.3. 23. Schéma explicatif de la réconciliation des associations.

La Réconciliation est par définition l'Action de Rétablissement des liens qui existaient auparavant. Dans notre cas, la modification apportée à une pièce de l'assemblage peut endommager certaines contraintes d'assemblages et certains liens (qui existent initialement) et par conséquent nous devons les rétablir. Subséquemment, les Listes des Features associées qui

existent au début vont être mise à jour (ou bien réconcilier), tout en tenant compte des listes de Features correspondantes. Si une Features appelée *Feature i* appartient à une liste des Features associées dite *liste x*, et au même temps cette même Feature (*Feature i*) a des Features correspondantes donc toutes la liste des Features correspondantes va être intercalé dans la *liste x* et on obtient ainsi la nouvelle liste de Features associées dite *listex'* (Fig.3. 23)

### 3.4. Algorithme de Propagation

#### 3.4.1 Analyse d'impact d'une modification

L'analyse de l'impact des changements (IA : Impact Analysis) est définie par Bohnner et al. [118] comme "l'identification des conséquences possibles d'un changement, ou l'estimation de ce qui doit être modifié afin accomplir un changement", et ils se concentrent sur l'IA en termes de conséquence des changements sur les détails de conception. En revanche, plusieurs auteurs se concentrent sur les risques associés aux changements et affirment que l'IA est : "l'évaluation des nombreux risques associés au changement, y compris l'estimation des effets sur les ressources, le budget, la main-d'œuvre et le planning". Durant le travail de cette thèse on s'est limité à déterminer les impacts prévisibles d'un changement et le faire propager tout en restant à l'intérieur du système CAO. On s'intéresse donc à estimer l'IA en termes de modifications de design uniquement.

L'analyse d'impact permet d'obtenir un ensemble d'impact estimé à partir d'un changement initial et de changements secondaires nécessaires. Il est vrai que l'analyse d'impact peut quelquefois surestimer ou sous-estimer les changements à effectuer, mais ça reste toujours un outil efficace qui nous permet d'évaluer un évènement (une modification dans notre cas) ainsi que son pouvoir d'impacter la totalité du système.

L'étude d'impacts en cadrans évalue pour un projet la nature des changements et l'importance de ceux-ci. Ce travail permet de quantifier et surtout de représenter les zones de changement dans un objectif de repérage ou de positionnement des causes réelles du changement. L'analyse quantitative recense les impacts et permet de visualiser de manière concrète population l'ampleur du changement, en se référant à plusieurs dimensions. Elle indique aussi les axes sur lesquels celui-ci sera le plus conséquent, ce qui sera capital pour déterminer les bons leviers d'accompagnement et ainsi optimiser la réussite du projet.

Durant la phase d'analyse d'impact d'une modification, cinq critères ont été pris en considération. D'une part, ces critères sont bien choisis, à la suite d'une recherche bibliographique approfondie. En effet ces éléments prennent en considération : la catégorie, l'urgence et aussi le nombre de pièces affectées. D'autre part on n'a pas négligé l'aspect cout (En termes de temps et d'argent). Cependant, chaque modification proposée sera évalué selon les valeurs associées à ces critères-là. Par conséquent, pour chacun d'entre eux, on associe une valeur notée  $c(i)$  ; avec  $c(i)$  est une grandeur allant de -2 à 2 selon le degré d'importance de l'évènement ou bien de la modification qui a eu lieu.

Commençant avec le premier critère : **la catégorie de modification**. Après avoir bien examiné les différentes catégories de modification existantes dans la littérature ; principalement 5 catégories de modification ont été définie. Les différentes alternatives sont les suivantes :

- Au sein du même composant ;
- 1 composant par 1 composant ;
- N composants par 1 composant ;
- N composants par N composants ;
- Ajout/ suppression d'une modification.

Pa conséquent, en se basant sur l'importance et la gravité de la modification ainsi que des effets indésirables possibles qu'elle peut engendrer, on associe les valeurs suivantes :

- Le même composant  $\rightarrow c(1) = 2$ ;
- 1 composant par 1 composant  $\rightarrow c(1) = 1$ ;
- N composants par 1 composant  $\rightarrow c(1) = 0$ ;
- N composants par N composants  $\rightarrow c(1) = -1$ ;
- Ajout/ suppression d'un composant  $\rightarrow c(1) = -2$ .

Le deuxième critère c'est la source de la modification ; ainsi une modification qui est faite dans le but de fixer ou corriger une erreur, va être considérée comme la plus importante. Le problème de sécurité peut aussi engendrer des modifications importantes. Également, une modification peut avoir comme objectif de changer le fonctionnement du système. Notamment, L'amélioration de la qualité du produit est un des facteurs les plus importants qui peuvent aussi engendrer des modifications. Mais ça n'empêche pas, que des simples améliorations du modèle sont parfois aussi demandées.

- Améliorer le design  $\rightarrow c(2) = 2$  ;
- Améliorer la qualité  $\rightarrow c(2) = 1$  ;
- Changer le fonctionnement  $\rightarrow c(2) = 0$  ;
- Améliorer la sécurité  $\rightarrow c(2) = -1$  ;
- Fixer une erreur  $\rightarrow c(2) = -2$  ;

L'urgence de la modification est considérée comme le troisième critère d'évaluation dans notre approche proposée. Une modification « Early », est une modification dans un stade (précoce) pas trop avancé dans le cycle de vie du produit et qui ne va pas causer par conséquent trop d'effets indésirables. Cependant, une modification dite « Mid-production » c'est une modification qui aura lieu après avoir lancé la production. Bien évidemment une modification appelée « Late » ou bien retardé est la modification qui va causer plus de dégâts. Conséquemment, on va affecter les valeurs suivantes :

- Early  $\rightarrow c(3) = 2$  ;
- Mid-production  $\rightarrow c(3) = 0$  ;
- Late  $\rightarrow c(3) = -2$  ;

En se basant sur le nombre de composant affecté et le nombre total des composants du système on peut déterminer le pourcentage des composants affectés à la suite de cette modification. En outre la gravité diffère d'un système à un autre. En effet une modification d'un seul composant pour un système à haute précision et dont les composants sont très coûteux (aéronautique) est parfois plus grave qu'une modification d'un ensemble de composant d'un autre système de caractéristiques différentes.

- % Pièces affectés  $\leq 5\%$   $\rightarrow c(4) = 2$  ;
- $5\% < \%$  Pièces affectés  $< 10\%$   $\rightarrow c(4) = 1$  ;
- $10\% < \%$  Pièces affectés  $< 25\%$   $\rightarrow c(4) = 0$  ;
- $25\% < \%$  Pièces affectés  $< 40\%$   $\rightarrow c(4) = -1$  ;
- Pièces affectés  $\geq 40\%$   $\rightarrow c(4) = -2$  .

De même pour le facteur coût, l'estimation du cout d'une modification et l'importance de ce coût la dépend énormément du système lui-même et de son coût global ainsi que des exigences du client. Donc, les intervalles de variation de prix seront fixés pour chaque modèle.

- $\text{Cout} \leq L_{\min} \rightarrow c(5) = 2$  ;
- $L_{\min} < \text{Cout} < L_1 \rightarrow c(5) = 1$  ;
- $L_1 < \text{Cout} < L_2 \rightarrow c(5) = 0$  ;
- $L_2 < \text{Cout} < L_{\max} \rightarrow c(5) = -1$  ;
- $\text{Cout} \geq L_{\max} \rightarrow c(5) = -2$  .

Vu qu'on a 5 critères différents nous devons attribuer à chacun d'entre eux un degré d'importance ou bien un poids. De ce fait, pour chaque critère  $c(i)$  on associe le poids  $W_i$  avec  $1 \leq i \leq 5$ . Ces coefficients nous permettent par la suite de calculer l'impact d'une modification quantifié par  $I_S$  appelé Score d'impact ou bien « Impact Score ». Ce dernier sera calculé en utilisant la formule montrée dans l'équation (3. 13). Selon des critères de validités bien définies à l'avant la décision de modification sera prise.

$$I_S = \frac{\sum_{i=1}^5 W_i \times c(i)}{\sum_{i=1}^5 W_i} \quad (3. 13)$$

avec  $\left\{ \begin{array}{l} c(i) \text{ c'est la valeur donnée au critère } i; \\ W_i \text{ c'est le poids donné à ce critère;} \end{array} \right.$

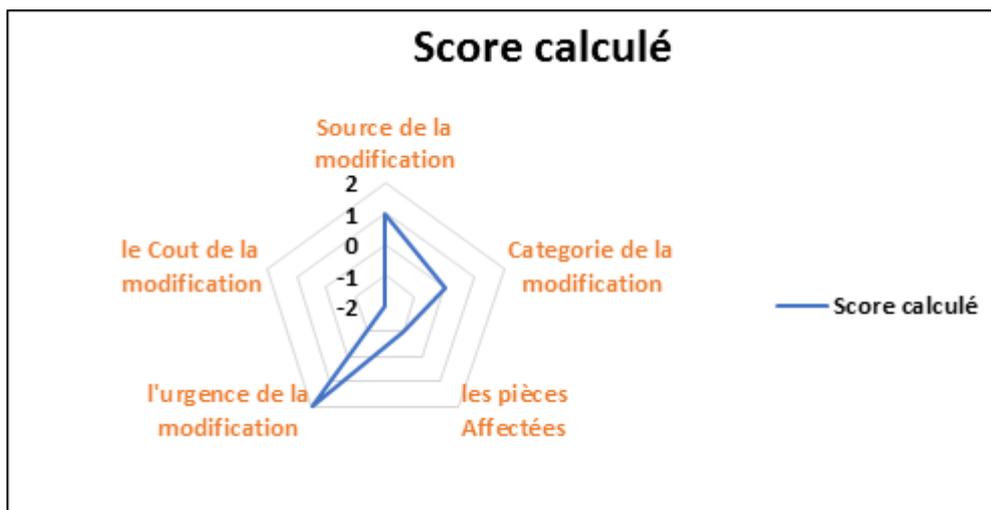


Fig.3. 24 . Exemple d'un diagramme d'araignée.

Tab.3. 1.Outil d'évaluation des conditions du changement.

Critère	Poids	Accepté	D'accord	Plutôt d'accord	Plutôt en désaccord	En désaccord
		+2	+1	0	-1	-2
Catégorie de modification	$W_1$					
Source de modification	$W_2$					
Urgence de modification	$W_3$					
Nombre de pièces affectés	$W_4$					
Cout de la modification	$W_5$					
<b>Total</b>		a	b	c	d	e

Total général :  $a+b+c+d+e$ , c'est la valeur calculer en utilisant l'équation (3. 13).

- $+7 < I_s < +10$  : Conditions favorables,
- $+1 < I_s < +6$  : Conditions plutôt favorables,
- $0 < I_s < -6$  : Conditions plutôt défavorables,
- $-7 < I_s < -10$  : Conditions défavorables,

### 3.4.2 Arbre et chemin de propagation de point de vu « Feature »

Grâce aux liens et aux associations qui ont été construits, il est possible de repérer les Features impactées par un changement. Une exploitation judicieuse de ces associations permettra une analyse d'impact et une propagation contrôlée et ciblée des modifications. Ainsi, seuls les Features réellement touchés par la modification d'une Feature d'un composant associé, seront traités. Cette section sera consacrée à la création d'un arbre de propagation des modifications. Cette dernière montrera clairement les chemins de propagation possible et les entités affectées par une modification proposée.

Les deux sections qui précèdent expliquent successivement l'algorithme de comparaison et de réconciliation. L'exécution de ces deux derniers nous permet d'avoir la liste des Features affectées par la modification et la liste des Features associées. Ce sont principalement ces deux types de liste qui vont nous servir dans cette section afin de déterminer le chemin de propagation d'une modification de point de vue Features.

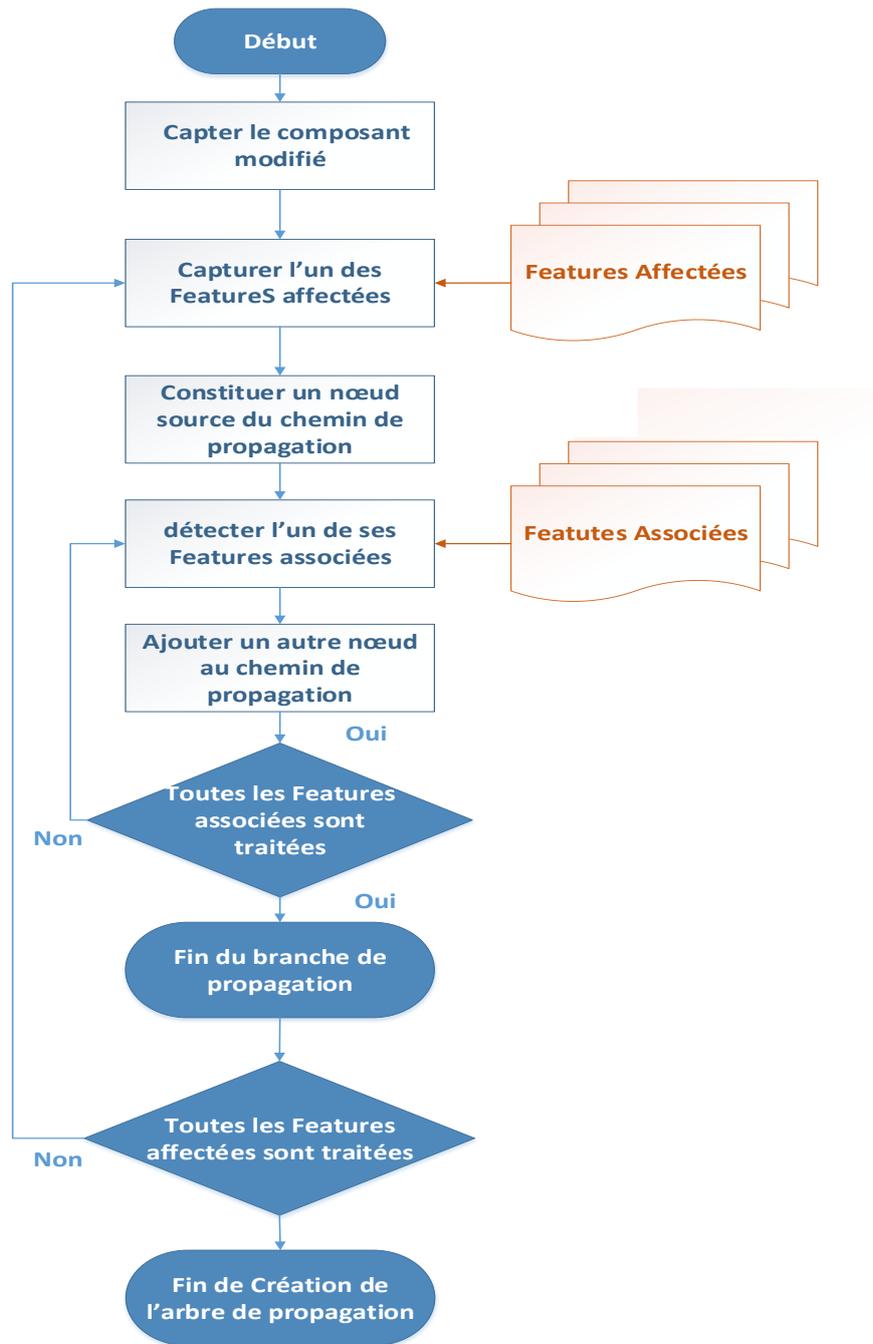


Fig.3. 25 . Algorithme de création de l'arbre de propagation.

Un chemin de propagation est défini par Masmoudi et al.[3] comme une succession d'arcs (ou chemin) telle que l'extrémité de chacun est l'origine du suivant. Ce chemin est bien caractérisé par sa propre longueur ou bien autrement par le nombre d'arcs qui le constituent. Par conséquent un arbre de propagation est défini par un graphe orienté non connexe et acyclique qui contient alors tous les chemins de propagation partant d'un même nœud instigateur de propagation. Selon

cette définition le chemin de propagation n'est qu'une branche appartenant à tout un arbre de propagation.

Ainsi, le chemin de propagation commence bien évidemment par un nœud (dans notre cas, le Feature) appelé "Source" de modification tout en passant par d'autres nœuds qui sont concernés par cette propagation appelés "Cibles". En effet, chacun de ces nœuds sera également considéré comme "Source" et aura donc des cibles. En fait, cela représente bien le phénomène de propagation.

Notre approche proposée se décompose principalement en trois sous cas, selon la nature de la Feature source de modification. En fait, si la Feature source est une Feature qui a subi une simple modification, cette dernière se propage automatiquement vers les Features qui lui sont directement associées. De même pour les Features qui ont été supprimés, leurs Features directement associées vont subir nécessairement la même action. Mais pour les Features récemment ajoutées dans la pièce source de modification, afin de propager la modification vers les Features associés on doit impérativement passer par les Features correspondantes à celles-ci.

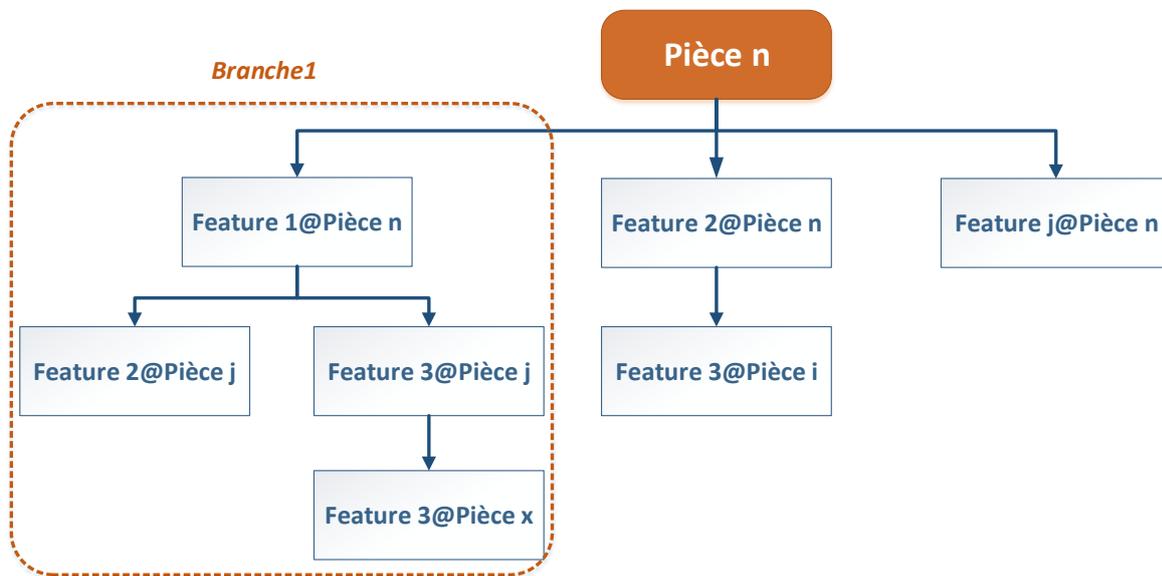


Fig.3. 26 . Exemple d'un arbre de propagation.

En exécutant l'algorithme de création de l'arbre de propagation (Fig.3. 25) chaque Features Affectées va être considérée comme source de modification, qui se propage vers la suite vers des Features Cibles qui vont être eux aussi par la suite considérées comme des nouvelles sources de

modification. Les chemins reliant la première Feature source et ses cibles de plus bas niveau forment une branche. Par conséquent l'assemblage de ces branches va nous donner l'arbre final de propagation de notre modification comme le montre la figure (Fig.3. 26).

### **3.4.3 Arbre pondéré de propagation des modifications**

La combinaison de la matrice de risque de propagation et l'arbre de propagation récemment créée va nous donner un arbre de propagation pondéré. En conséquence chaque chemin de propagation, va être pondéré avec une valeur qui mesure le risque de ce chemin.

## **4. Conclusion**

Les algorithmes développés dans ce chapitre ne sont que des sous algorithmes d'un algorithme général d'aide à la décision qui vise à faciliter la tâche de designer et ses différents partenaires dans le cadre de la conception collaborative. Quelques exemples illustratifs des différentes étapes sont aussi présentés. L'objectif de l'algorithme général développé est de pouvoir mesurer l'impact d'une modification et la détermination d'un arbre de propagation des modifications. La détermination, de l'arbre de propagation est principalement fondue sur l'algorithme de parcours de graphe en largeur qui a montré des bonnes performances. Dans ce chapitre aussi les différents critères sur les quel se base l'analyse d'impact ont été détaillés. Cette approche présentée permet de juger convenablement la faisabilité et l'impact d'une modification proposée. Ces algorithmes vont être par la suite traduits et implémenté dans le chapitre suivant à fin d'obtenir un outil efficace qui facilite la gestion des modifications et facile à utiliser.

**Chapitre 4 :**  
**Implémentation & Validation**

# Chapitre 4 : Implémentation et validation

## 1. Introduction

Dans ce chapitre on s'intéresse principalement par l'implémentation, qui est la réalisation, ou la mise en œuvre /pratique de l'approche proposée. La mise en œuvre désigne la création d'un « produit fini ». Dans notre cas le produit fini à obtenir est un outil qui sera intégré dans le logiciel de conception assisté par ordinateur SolidWorks. La première partie de ce chapitre, sera donc consacré à la présentation des outils informatiques utilisés afin de concevoir l'outil désiré. Nous commençons tout d'abord par une présentation des logiciels utilisés, le langage de programmation choisi, ainsi que les motivations de ce choix. La deuxième partie de chapitre consacré à la présentation de la mise en œuvre de notre application ainsi que ses différentes interfaces. Enfin deux cas d'études sont présentés pour montrer la validité de l'outil implémenté. Ce chapitre est terminé par une synthèse de nos résultats obtenus.

## 2. Environnement de développement

### 2.1 Logiciels & bibliothèques utilisés

#### 2.1.1 SolidWorks

SolidWorks est un logiciel de conception assistée par ordinateur, appartenant à la société Dassault Systèmes. Ce logiciel de conception mécanique est un outil de modélisation volumique paramétré, basée sur des « Features », qui tire parti des fonctionnalités de Windows, connu pour sa convivialité. On peut créer des modèles volumiques 3D entièrement intégrés avec ou sans contraintes tout en utilisant des relations automatiques ou définies par l'utilisateur pour saisir l'intention de conception. Un modèle SolidWorks est entièrement intégré par rapport aux mises en plan et aux assemblages qui le référence. Les changements introduits dans le modèle sont entièrement reflétés dans les mises en plan et les assemblages qui lui sont associés. Inversement si vous effectuez des changements dans le contexte d'une mise en plan ou d'un assemblage, ces changements sont reflétés dans le modèle. Les relations géométriques telles que les relations parallèles, perpendiculaires, horizontales, verticales, concentriques et coïncidentes sont des exemples des contraintes supportées par SolidWorks. Des équations

peuvent également utilisées pour établir des relations mathématiques entre les paramètres. Par ces moyens vous pouvez garantir que des concepts seront établis et conservés.

Un dossier complet contenant l'ensemble des relatifs à un même système constitue une maquette numérique. De nombreux logiciels viennent compléter l'éditeur SolidWorks comme les utilitaires orientés métiers (tôlerie, bois, BTP...), et les applications de simulation mécanique travaillent à partir des éléments de la maquette virtuelle.

### **2.1.1.1 Les API du SolidWorks**

Il faut tout d'abord définir le terme API : Une API (Application Programmable Interface, « interface de programmation » ou « interface pour l'accès programmé aux applications) est un ensemble de fonctions permettant d'accéder aux services d'une application, par l'intermédiaire d'un langage de programmation. Une interface de programmation d'application (API) est une interface mise en œuvre par un logiciel qui lui permet d'interagir avec d'autres logiciels. Il facilite l'interaction entre les différents logiciels comme le fait que l'interface utilisateur facilite l'interaction homme-machine.

Une API permet de fournir un certain niveau d'abstraction au développeur, c'est-à-dire qu'elle lui masque la complexité de l'accès à un système ou à une application en proposant un jeu de fonctions standard dont seuls les paramètres et les valeurs retournées sont connus. Ainsi, par analogie avec une voiture, le conducteur n'a pas à connaître le fonctionnement mécanique du moteur d'un véhicule pour pouvoir le conduire. Seule une interface composée d'un volant, de pédales (accélérateur, embrayage, frein), de manettes (clignotants, phares, boîte de vitesse) et de boutons (warning, antibrouillard, klaxon, etc.) lui sont accessibles : il s'agit d'une certaine façon de l'interface proposée à l'utilisateur.

Grâce aux API, un développeur n'a donc pas à se soucier de la façon dont une application distante fonctionne, ni de la manière dont les fonctions ont été implémentées pour pouvoir l'utiliser dans un programme. Une API peut être disponible pour un langage particulier ou bien être disponible pour plusieurs langages de programmation

Si on veut automatiser le travail, on doit répondre à cette question très importante : Est-ce que le problème est mieux résolu en utilisant une macro, add-ins, ou stand-alone ? Répondre à cette question nous oblige à poser plusieurs autres questions :

- Quelles sont les différences techniques entre ces trois types de programmes ?
- Combien de temps et d'argent puis-je investir dans ce programme ?
- Quelles sont les compétences (en termes de programmation et API) demandées pour écrire ce programme ? Quels langages dois-je apprendre ?
- Comment ce programme va être déployé ? Maintenu ? Modifié (si nécessaire) ? Sécurisé ?
- Est-ce que je veux ce programme pour être intégré à SolidWorks ?

On doit donc évaluer macros, add-ins, et stand-alone à partir d'une variété de points de vue afin qu'on puisse faire le meilleur choix pour notre projet d'automatisation.

### **2.1.1.2 Types d'applications de l'API SolidWorks**

#### **2.1.1.2.1 Les macros du SolidWorks**

Une macro est un programme qui s'exécute uniquement à l'intérieur d'un autre programme. C'est un procédé d'écriture permettant de définir des opérations composées à partir des instructions du répertoire de base d'une machine donnée, ou à partir du répertoire des commandes d'un programme ou d'un logiciel (SolidWorks ici). D'après l'aide de SolidWorks[119], les macros sont définis comme des scripts qui permettent d'exécuter automatiquement des opérations dans le logiciel (SolidWorks).

On peut créer une macro et la programmer à l'extérieur du logiciel SolidWorks ou enregistrer une macro capturant une séquence d'actions et de commandes qu'on effectue. Mais la façon la plus rapide et la plus simple de démarrer la programmation avec l'API SolidWorks est d'enregistrer une macro SolidWorks, qui contient les appels de l'API SolidWorks qui correspondent aux actions effectuées dans l'interface utilisateur. On peut également modifier la macro dans Microsoft Visual Basic for Applications (VBA) ou Microsoft Visual Studio Tools for Applications (VSTA) pour répondre aux besoins de l'espace de travail.

#### **2.1.1.2.2 Les add-ins**

Un add-in, aussi connu comme add-on ou plug-in, est aussi un programme qui fonctionne à l'intérieur d'un autre programme. Contrairement aux macros, ils doivent être enregistrés et chargés avant de pouvoir les exécutées correctement. Les add-ins peuvent être écrites en VB.NET, C # ou C ++, ce qui signifie qu'ils vont être écrites et éditées en utilisant le Visual

Studio, puis exécutez à partir d'une DLL compilée. Pour créer des add-ins SolidWorks on doit avoir SolidWorks API SDK et du Visual studio.

### 2.1.1.2.3 Les stand-alone

Un stand-alone est une application Windows qui s'exécute dans son propre processus, en dehors du processus SLDWORKS.EXE, d'où son nom (Les modules autonomes). Les stand-alone peuvent être écrites en VB.NET, C # ou C ++, ce qui signifie qu'ils vont être rédigés en utilisant les fichiers de projet Visual Studio, puis exécutez à partir d'un fichier EXE ou DLL compilé, selon le type de projet.

### 2.1.1.3 SDK de l'API SolidWorks

Le terme **SDK** signifie **S**oftware **D**éveloppement **K**it, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc., pour un terminal et/ou un système d'exploitation spécifique. Le SDK de l'API SOLIDWORKS contient toutes les fonctionnalités dont on a besoin pour développer des applications SOLIDWORKS API. Il est extrêmement important de comprendre la différence entre une API et un SDK et de savoir quand chacune d'elles doit être fournie pour la création d'un écosystème de développeurs. Dans le domaine du développement, ces deux outils représentent le moteur de la communication Web et de l'implémentation d'API externes. De plus, il est important de savoir exactement ce que l'on entend précisément quand on parle d'API et de SDK. Dans la présente section, nous tenterons d'établir une définition globale de ces deux concepts.

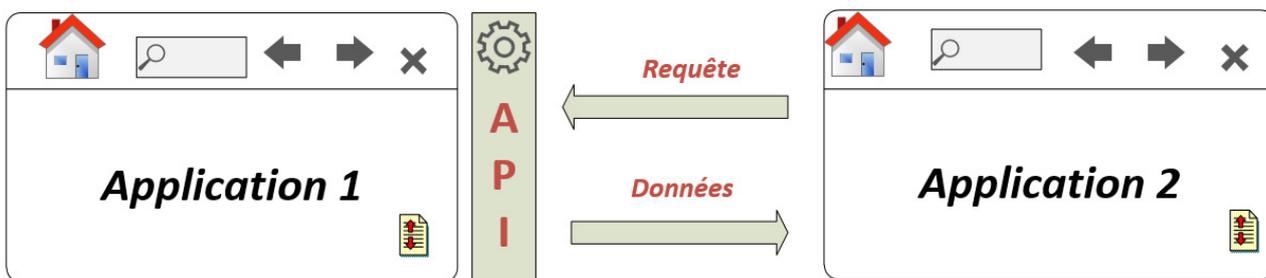


Fig.4. 1.Communication de deux applications via les API.

L'API est une interface qui donne au logiciel la possibilité d'interagir entre eux. Celui-ci définit un ensemble de règles que les programmes doivent respecter pour pouvoir communiquer entre eux. Les API décrivent généralement comment des routines, des structures de données, etc.

doivent être configurées pour que les deux applications communiquent. Les API doivent être mises en œuvre en affichant uniquement les fonctionnalités ou les données qui sont requises à l'extérieur, tout en gardant inaccessibles les autres parties de l'application. Le recours aux API est devenu très répandu.

Le SDK est un ensemble d'outils servant à développer des applications logicielles adaptées à une plate-forme spécifique. Les kits de développement comprennent des outils, des bibliothèques, de la documentation et des extraits de code qui sont susceptibles d'aider un programmeur à développer une application. La plupart des SDK peuvent être téléchargés sur Internet le plus souvent et un grand nombre de SDK sont gratuitement fournis pour inciter les programmeurs à utiliser le langage de programmation de ce même logiciel.

L'API est une interface qui permet au logiciel de communiquer avec les autres logiciels, contrairement au SDK qui peut être utilisé pour développer des applications logicielles ciblant une plate-forme spécifique. La plus simple version d'un SDK peut être une API qui contient quelques éléments essentiels pour interagir avec un langage de programmation donné. Par conséquent, une API peut être assimilable à un simple SDK sans support de débogage, etc.

#### **2.1.1.4 Personnalisation de l'interface utilisateur SOLIDWORKS**

On peut personnaliser les menus, raccourcis du clavier, barres d'outils et Volet des tâches mais on peut également affecter des macros et y accéder ensuite rapidement depuis l'interface utilisateur ou le clavier. Donc après avoir créé la macro désirée on peut y affecter un bouton qui sera accessible facilement depuis l'interface utilisateur SolidWorks.

#### **2.1.2 Visual studio**

Afin de rassembler au sein d'un même environnement de développement plusieurs de ses outils de programmation, Microsoft a publié la première version de Visual Studio en 1997. D'où, Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications webASP.NET, des services web XML, des applications bureautiques ainsi que des applications mobiles. Cet environnement supporte différents langages de programmation : C++, C, C#, F#, Visual Basic, Python, Q#. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages

permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Développeur.

## 2.2 Langage de programmation

En générale, un langage de programmation est un ensemble d'instructions (un ensemble de mots) qui permettent de construire des applications informatiques. Les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par le dispositif informatique, et d'autre part d'indiquer comment s'effectuent les manipulations, conformément à quel algorithme. Ils servent de moyen de communication par lequel le programmeur communique avec son ordinateur, mais aussi avec d'autres programmeurs.

On a choisi de travailler avec le langage de programmation C# (C Sharp). Il s'agit d'un langage de programmation moderne, polyvalent et orienté objet, développé par Microsoft afin de développer des applications sur Windows. Le langage de programmation C# est un langage très facile à apprendre. Il est entièrement basé sur les langages C et C++. Pourtant, la syntaxe du langage C# est très expressive, mais elle est aussi simple et facile à apprendre.

C# simplifie de nombreuses complexités du C++ et offre aussi des fonctionnalités puissantes telles que les types de valeurs nulles, les énumérations et l'accès direct à la mémoire qui ne se trouvent pas dans d'autre langage. Ce langage nous permet de nous concentrer sur les fonctionnalités de notre programme plutôt que sur les aspects techniques

Aussi de point de vue performance, C# est relativement rapide car c'est langage qui est fortement typé. Le système n'a pas à calculer la place de la mémoire, il la connaît déjà.

## 3. Implémentation de l'outil « CAD Change propagation Tool »

Notre approche proposée est traduite en un outil qui facilite la tâche du designer, et l'aide à gérer et propager les changements, appelé : « CAD change propagation Tool » ou CAD CPPT. Comme nous l'avons déjà mentionné, nous avons utilisé le langage de programmation C# et les Api du SolidWorks pour réaliser cet outil. C'est un outil facile à utiliser est bien guider. La première interface utilisateur qui apparait dans la figure (Fig.4. 2 ) offre à l'utilisateur la possibilité de choisir le fichier sur lequel il va travailler, soit il utilise le fichier qui est déjà ouvert ou bien il importe son propre fichier désiré. Dès que le choix de fichier est fait la deuxième interface,

permettant l'extraction des données et la construction de toutes les matrices nécessaires ainsi que l'affectation de tous les arbres de propagation possibles, apparaisse (Fig.4. 3). Le designer dispose ainsi de toutes les informations qui le permettent de juger n'importe quelle modification proposée.

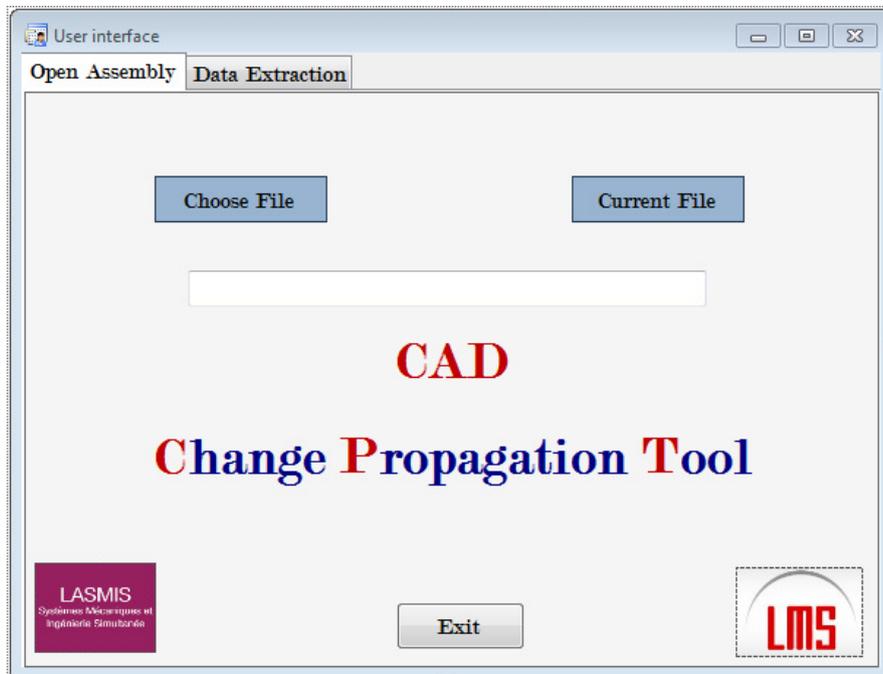


Fig.4. 2. La première interface utilisateur de l'outil « CAD CPT ».

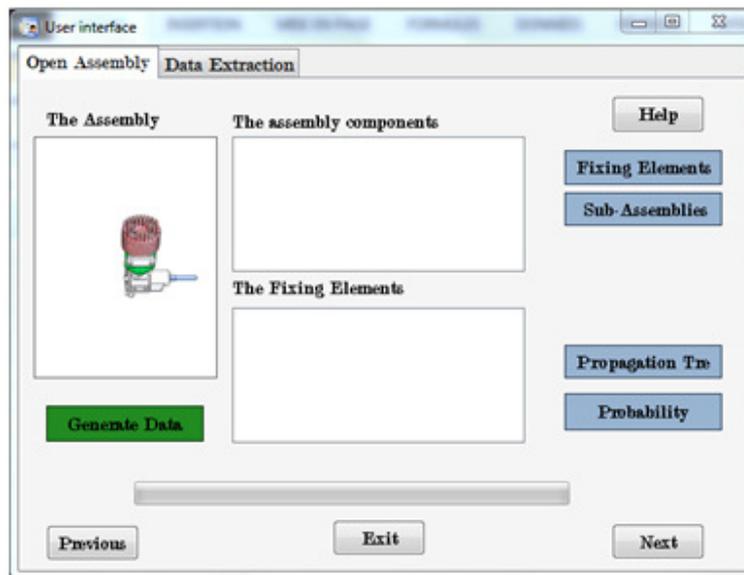


Fig.4. 3. L'interface de l'extraction des données de l'outil « CAD CPT ».

L'interface appelée « l'extracted Data » (Fig.4. 4) offre à l'utilisateur la possibilité d'examiner toutes les matrices construites, les sous assemblages existants et les listes des Features associées avant de choisir le composant qu'il désire le modifié. Elle permet d'ouvrir d'autre interface montrant plus de détails (Fig.4. 5) et aussi d'aller directement dans la base de données créée. Une fois le choix de la cible est affirmé, l'outil CAD CPT permet à l'utilisateur de mesurer l'impact prévisible de la modification proposée (Fig.4. 6) (avant d'entrer dans les détails de la modification).

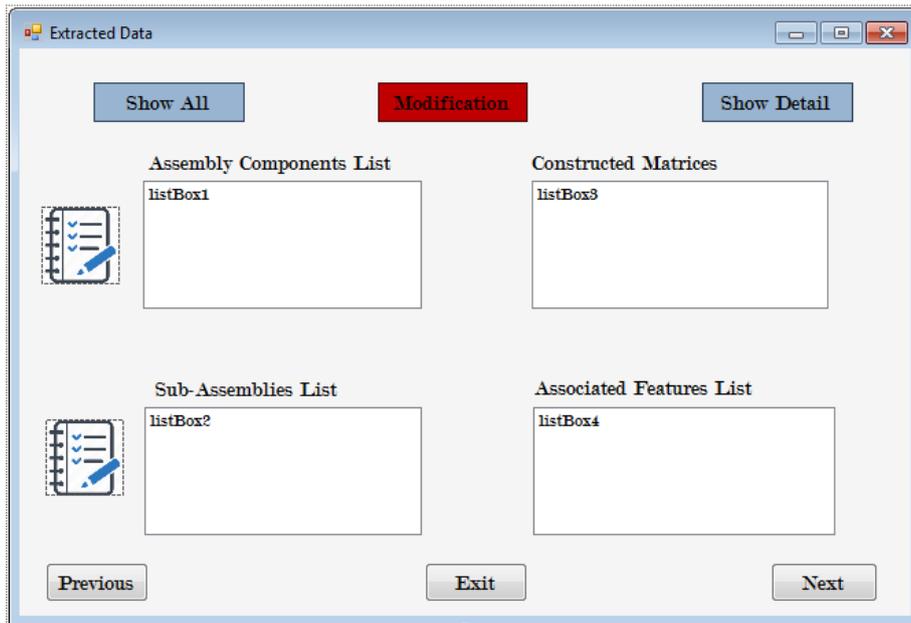


Fig.4. 4. L'interface « Extracted Data » de l'outil « CAD CPT ».

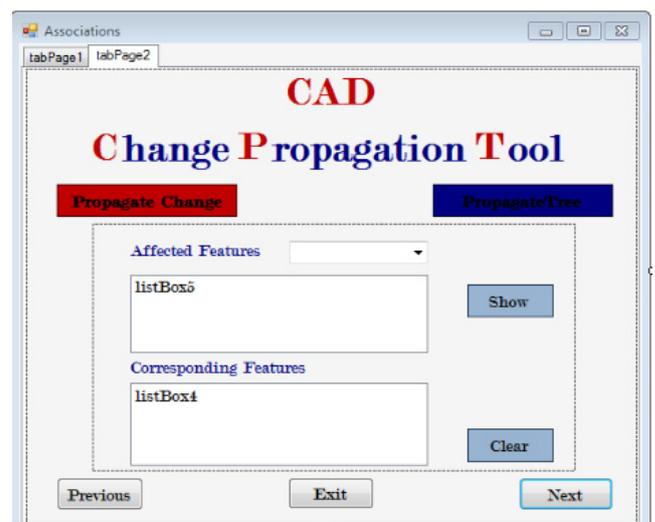
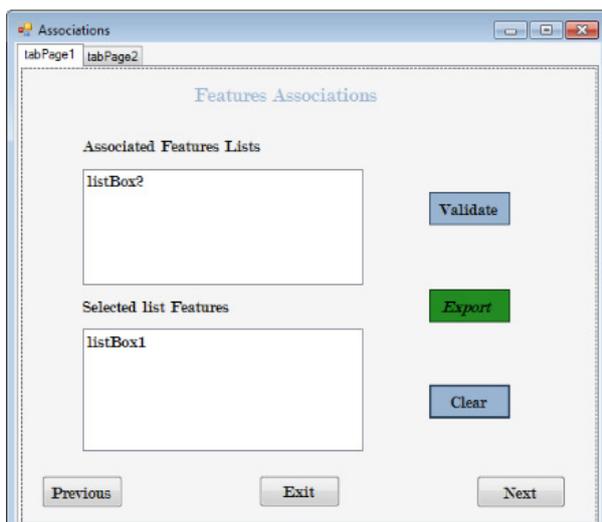


Fig.4. 5 . L'interface « Associations » de l'outil « CAD CPT »

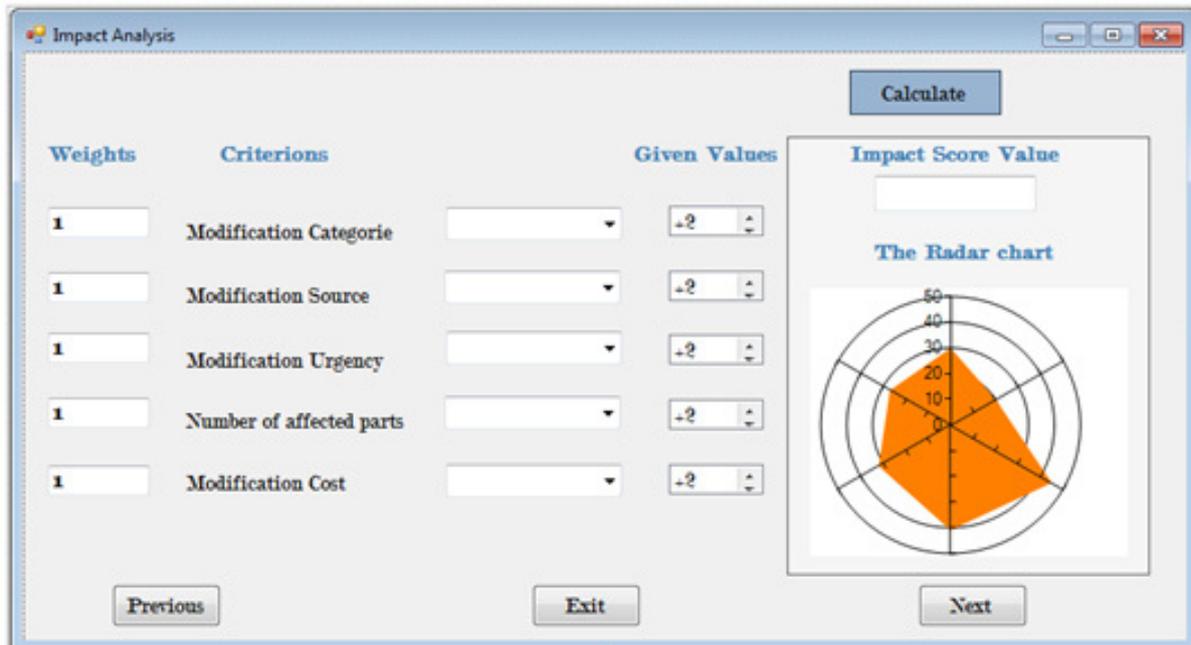


Fig.4. 6. Interface d'analyse d'impact de la modification proposée de l'outil « CAD CPT ».

#### 4. Implémentation de l'outil « CAD Part Comparaison »

Notre propre outil de Comparaison a été implémenté afin de comparer deux versions différentes d'un modèle CAO. C'est un outil qui peut être utilisé non seulement dans le cadre de la propagation des modifications. Mais aussi ça peut être utilisé seule comme un SolidWorks add-in pour comparer n'importe quelles deux pièces CAO afin de détecter la similarité et la différence existantes entre elles.

L'interface de visualisation des résultats de comparaison (Fig.4. 7) est l'interface qui s'affiche automatiquement une fois la comparaison des deux pièces en question est terminée. Elle se compose principalement de quatre « TabControl » ou onglets. La première liste les Features affectées : Celles qui sont modifiées, supprimées ou nouvellement créées. Un bouton appelé « Export » c'est le premier bouton de la figure (Fig.4. 8) est un bouton qui permet l'exportation de ce résultat vers un fichier Excel. Le deuxième onglet c'est un onglet supplémentaire qui a pour but simplement la mise en valeur des résultats de la comparaison toute en colorant les faces concernées avec des couleurs qui facilitent la visualisation de la modification mise en œuvre, et ceci en utilisant le bouton « View Faces » qui correspond au deuxième bouton de la figure (Fig.4. 8). Tandis que le troisième onglet appelé « Détails » est un onglet facultatif qui vise à calculer le taux de modification d'une pièce après avoir exploité les résultats de la comparaison. Cependant le dernier onglet « Correspondante Features » est un onglet très important qui exécute

l'algorithme de correspondance afin de créer des correspondances entre les Feature récemment créés et les anciennes (qui existe initialement avant la modification). Le bouton « Export » de cet onglet (bouton 4 de la (Fig.4. 8)) permet bien évidemment l'enregistrement de ces listes des Features correspondantes dans la base de données (Fichier Excel).

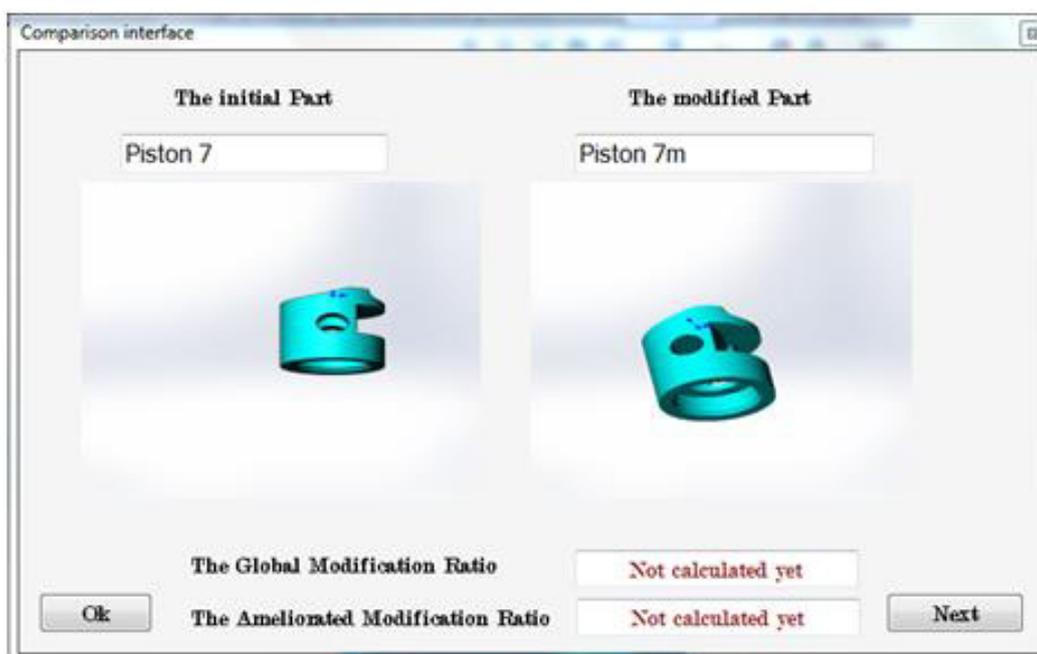


Fig.4. 7 . Interface de démarrage de l'outil de Comparaison des Pièces CAO « **CAD Part Comparison** ».

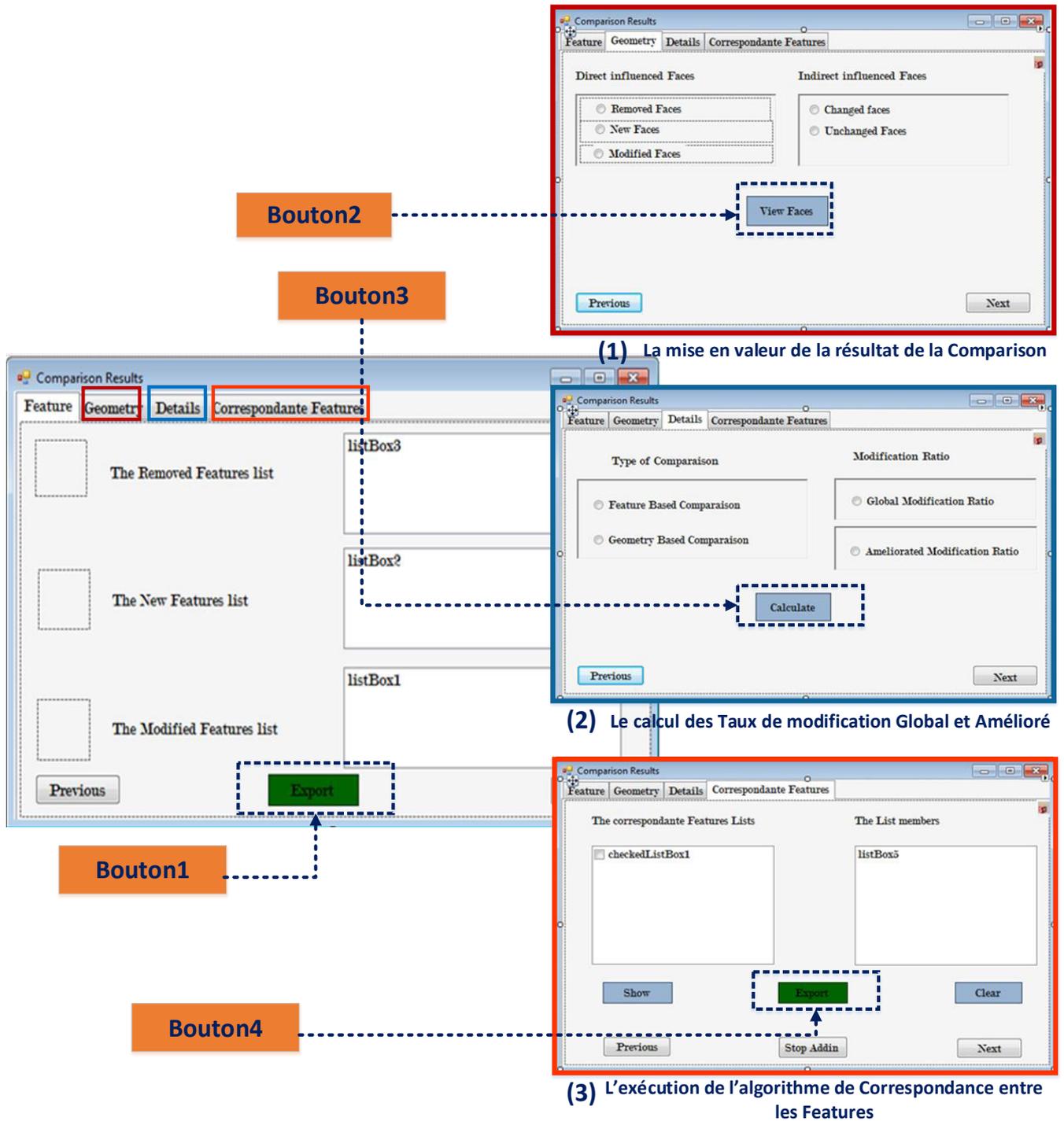


Fig.4. 8.Interface de visualisation des résultats de l'outil de Comparaison des Pièces CAO

« CAD Part Comparison ».

## 5. Validation

### 5.1 Premier cas d'étude : Roue de Remorque

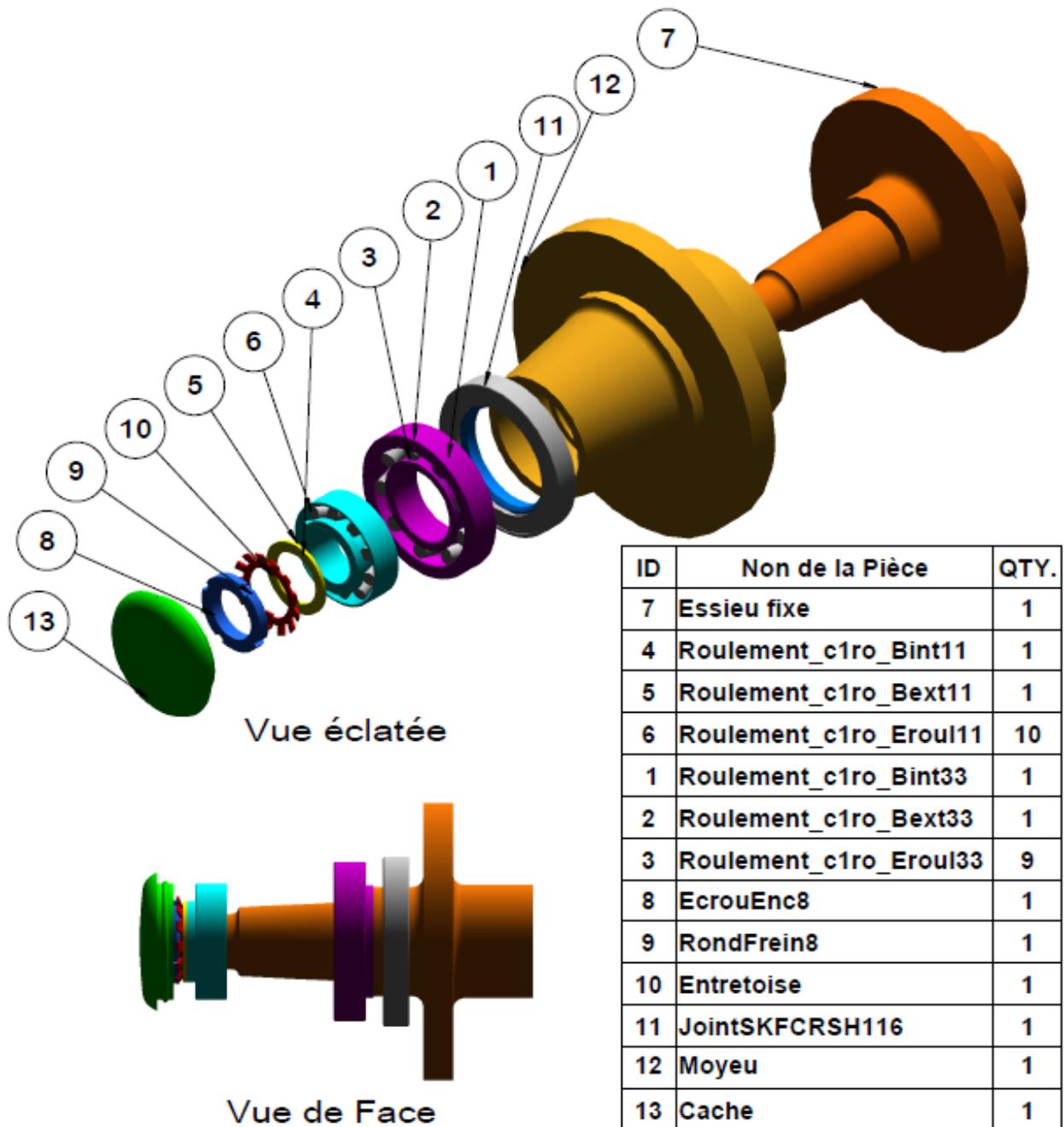


Fig.4. 9.Modèle CAO d'une Roue de Remorque.

Note premier cas d'étude est un modèle CAO d'une Roue de Remorque constitué de 13 composants comme le montre la figure (Fig.4. 9). Durant de la phase de *la simulation numérique* et plus précisément lors de la validation et optimisation du dimensionnement il a été prouvé qu'une augmentation du diamètre de l'arbre C7 (Essieu fixe) améliore les caractéristiques

du système (sa résistance aux charges, sa durée de vie ...), D'où vient la proposition de la modification du modèle CAO développée dans la figure (Fig.4. 10). En fait c'est une augmentation du diamètre de l'essieu de 1 mm qui nécessite l'augmentation des esquisses 3, 4, 5,6 et 8 tous de 1 mm

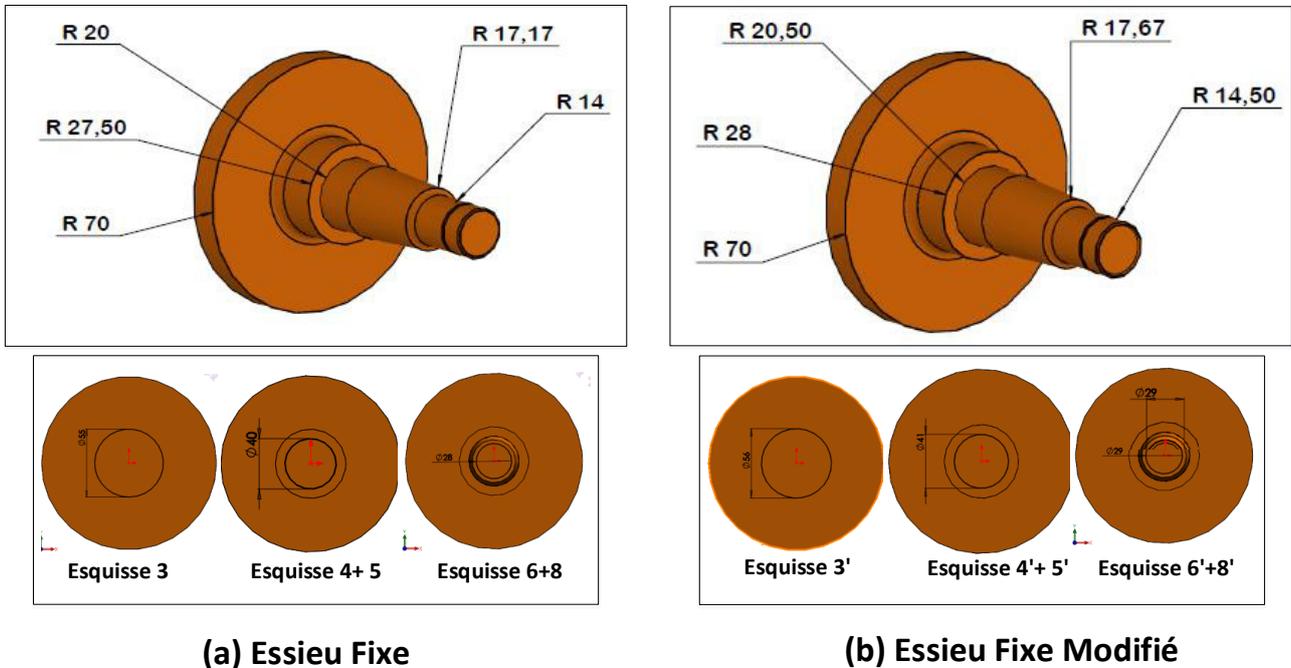


Fig.4. 10.Modification apporté à l'Essieu fixe.

### 5.1.1 L'extraction des données

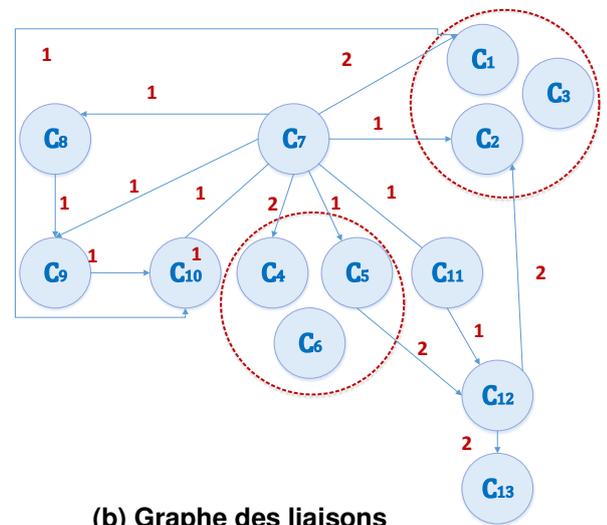
La première étape c'est l'extraction des données d'assemblage et des données topologiques et géométriques des pièces qui sont enregistrés dans la base de données spécifique à notre assemblage qui est créée dès l'exécution de notre add-in (juste après avoir choisi le modèle CAO souhaité). Notre modèle CAO d'une Roue de Remarque principalement constitué de 13 composants comme le montre la figure (Fig.4. 9). Ces derniers sont reliés entre eux par 26 contraintes d'assemblages standards au totale. Ces contraintes sont exploitées afin de construire la matrice des Relations. De même, les interférences volumiques et les interférences de coïncidence sont détectées et extraient pour construire la matrice des Interférences. La matrice des Relations construite représente la matrice d'adjacence du graphe des liaisons du mécanisme, qui est parcourus afin de déterminer les différents arbres et chemins de propagation possibles.

Initial Assembly	first level	second level	Area	Volume	Nbr	FObjective
Roue de Remorque3	Roulement_c1ro33-1	Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	1,52259E-05	0,00764297	0	0,01583681
		Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	2,07425E-05	0,00983172	2	0,820569
		Roulement_c1ro33-1/Roulement_c1ro_Eroul33-1			0	0
	Roulement_c1ro11-1	Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	8,24264E-06	0,0050046	1	0,41007576
		Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	1,21169E-05	0,00688412	1	0,41399239
		Roulement_c1ro11-1/Roulement_c1ro_Eroul11-1			0	0
	Roulement_c1ro11-1/Roulement_c1ro_Eroul11-1	Essieu fixe-1	0,000494899	0,05925044	4	1,78695299
		EcrouEnc8-1	4,81473E-06	0,00293399	2	0,805904
		RondFrein8-1	9,48259E-07	0,0019499	3	1,20354028
		Entretoise-1	1,31947E-06	0,00175929	3	1,20327322
		JointSKFCRSH116-1	1,80186E-05	0,01994082	2	0,83762218
		Moyeu-1	0,000574178	0,10392837	6	2,67787263
		Cache-1	2,38597E-05	0,01171358	2	0,82436076
			<b>0,001174365</b>	<b>0,2308398</b>	<b>26</b>	

Fig.4. 11. Les données extraites.

		Roult	Rou	Roult	Roult	Roult	Roult	Roult	Roult	Essie	Ecrou	Rond	Entre	Joint	Moye	Cache
C1	Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
C2	Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0
C3	Roulement_c1ro11-1/Roulement_c1ro_Eroul11-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C4	Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
C5	Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0
C6	Roulement_c1ro33-1/Roulement_c1ro_Eroul33-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C7	Essieu fixe-1	1	1	0	2	1	0	0	0	1	1	1	1	1	0	0
C8	EcrouEnc8-1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
C9	RondFrein8-1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
C10	Entretoise-1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0
C11	JointSKFCRSH116-1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
C12	Moyeu-1	0	2	0	0	2	0	0	0	0	0	0	0	1	0	2
C13	Cache-1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0

(a) Matrice de Relation



(b) Graphe des liaisons

Fig.4. 12. Graphe des liaisons du système Roue de Remorque et sa matrice d'adjacence.

Puisque notre système est composé de 13 pièces le parcours de graphe des liaisons en largeur nous permet d'obtenir donc 13 arbres de propagations différentes, en changeant à chaque fois de Source de modification (La pièce qui a engendré la propagation). La pièce qu'on va la modifier c'est le composant C7« essieu fixe », donc on s'intéresse principalement à l'arbre de propagation des modifications de celui-là (Fig.4. 12) dans le but de déterminer le risque provoqué par une modification qui le touche. En effet l'obtention de cet arbre nous a permis de déterminer les deux matrices : d'influence et de probabilité de propagation (Fig.4. 14 ) qui vont nous servir par la suite pour le calcul de Risque engendré par la modification.

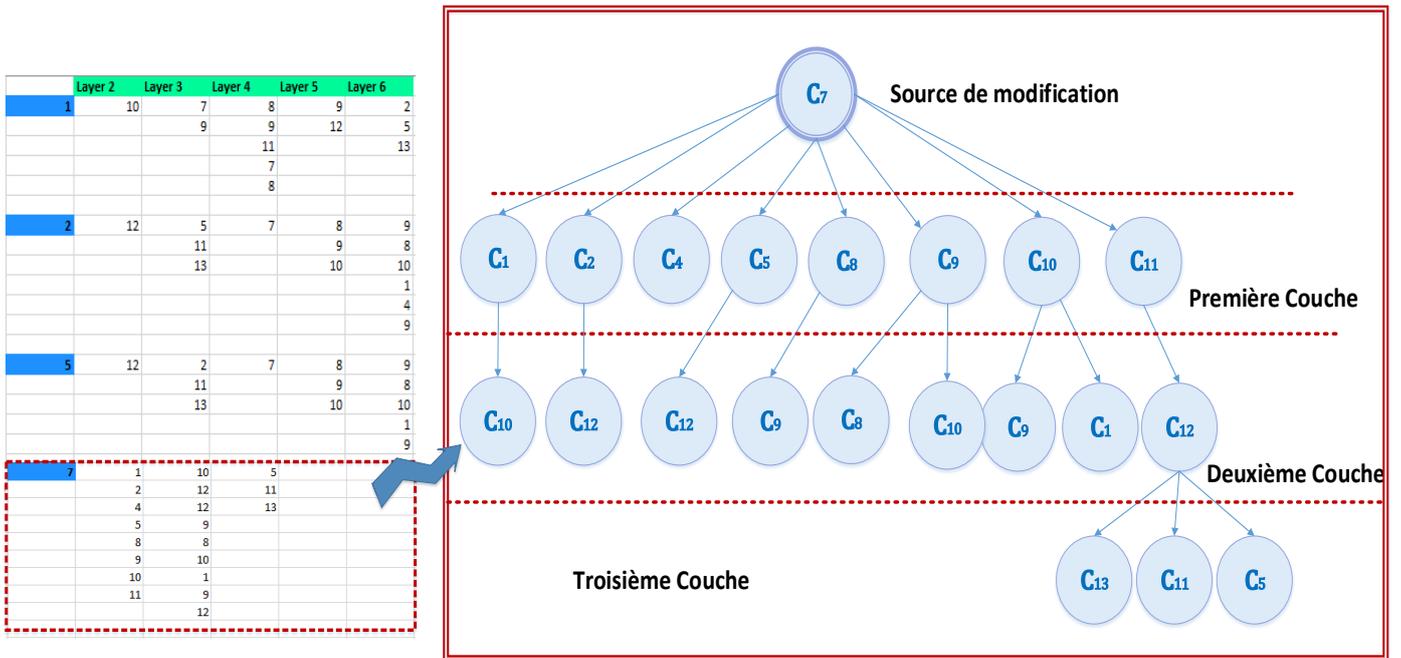


Fig.4. 13. Arbre de propagation des modifications du Composant C7 (Essieu fixe).

	Roulem	Roulem	Roulem	Roulem	Roulem	Roulem	Essieu fixe-1	EcrouEn	RondFre	Entreto	JointSKF	Moyeu-	Cache-1
Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	0	0,135	0	0,09	0,144	0	1,755	0,71379	0,36963	0,61677	0,00673	0,0945	0
Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	0,06	0	0	0,06	0,09	0	0,9	0	0	0,027	0,06729	0,763	0,06872
Roulement_c1ro11-1/Roulement_c1ro_Eroul11-1	0	0	0	0	0	0	0	0	0	0	0	0	0
Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	0,09	0,135	0	0	0,135	0	1,35	0	0	0,0405	0	0,0945	0
Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	0,06	0,09	0	0,06	0	0	0,9	0	0	0,027	0,06729	0,763	0,06872
Roulement_c1ro33-1/Roulement_c1ro_Eroul33-1	0	0	0	0	0	0	0	0	0	0	0	0	0
Essieu fixe-1	0,06667	0,1	0	0,06667	0,1	0	0	0,06845	0,04844	0,03	0	0,07	0
EcrouEnc8-1	3,52661	5,18097	0	3,45398	0	0	217,2685891	0	0,96987	1,91744	0	3,62668	0
RondFrein8-1	15,5882	23,3823	0	15,5882	0	0	347,6225037	2,1965	0	7,52787	0	16,3676	0
Entretoise-1	0,26	0,09	0	0,06	0	0	1,17	1,58619	0,72215	0	0	0	0
JointSKFCRSH116-1	0,71775	1,07662	0	0,71775	1,24636	0	10,76621552	0	0	0,32299	0	10,4475	0,95171
Moyeu-1	0,01543	0,12857	0	0,00771	0,14014	0	0,115714286	0	0	0,027	0,09613	0	0,09818
Cache-1	0,00771	1,06351	0	0,00771	1,06351	0	0,957159705	0	0	0	0,80395	8,27175	0

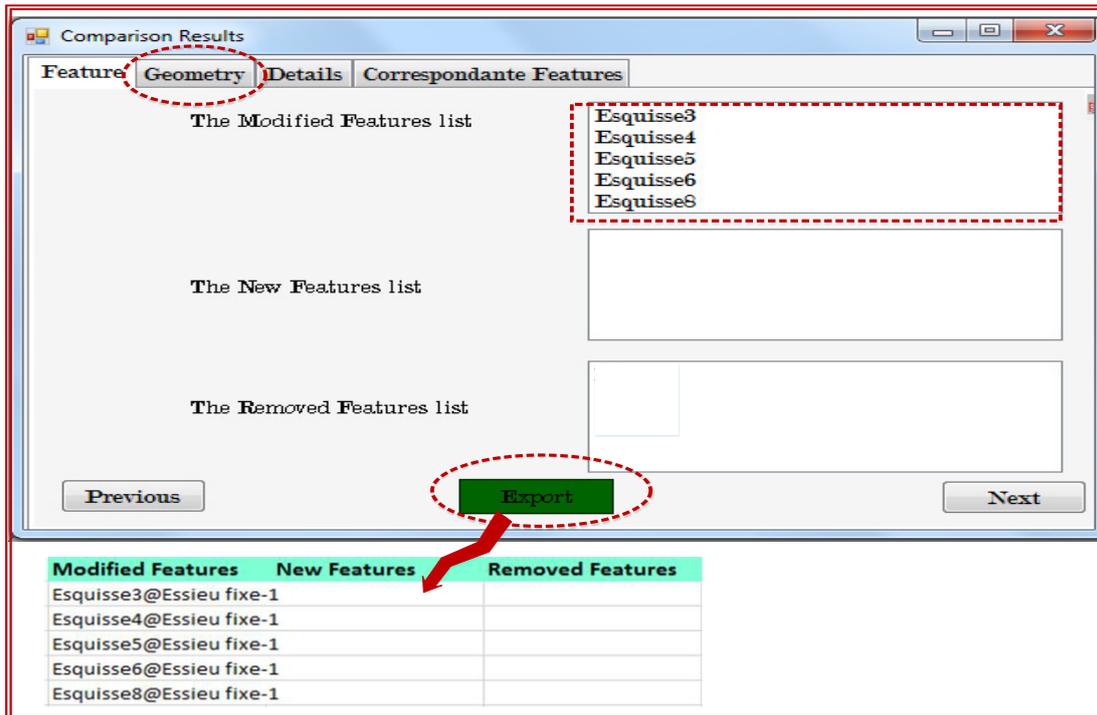
(a) Matrice des Influences

	Roulem	Roulem	Roulem	Roulem	Roulem	Roulem	Essieu fixe-1	EcrouEn	RondFre	Entreto	JointSKF	Moyeu-	Cache-1
Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	0	0,03704	0	0,11111	0,03704	0	0,111111111	0,05556	0,14815	0,33333	0,05556	0	0
Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	0,05556	0	0	0,11111	0,22751	0	0,111111111	0,05556	0,03704	0,03704	0,19841	0,28571	0,28571
Roulement_c1ro11-1/Roulement_c1ro_Eroul11-1	0	0	0	0	0	0	0	0	0	0	0	0	0
Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	0,11111	0,07407	0	0	0,07407	0	0,222222222	0,11111	0,07407	0,07407	0,11111	0	0
Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	0,05556	0,22751	0	0,11111	0	0	0,111111111	0,05556	0,03704	0,03704	0,19841	0,28571	0,28571
Roulement_c1ro33-1/Roulement_c1ro_Eroul33-1	0	0	0	0	0	0	0	0	0	0	0	0	0
Essieu fixe-1	0,5	0,33333	0	1	0,33333	0	0	0,5	0,33333	0,33333	0,5	0,2619	0,19048
EcrouEnc8-1	0,05556	0,03704	0	0,11111	0,03704	0	0,111111111	0	0,33333	0,14815	0,05556	0,0291	0,02116
RondFrein8-1	0,25	0,05556	0	0,16667	0,05556	0	0,111111111	0,5	0	0,33333	0,08333	0,04365	0,03175
Entretoise-1	0,5	0,07407	0	0,22222	0,07407	0	0,111111111	0,25	0,33333	0	0,11111	0,04365	0,03175
JointSKFCRSH116-1	0,05556	0,13228	0	0,11111	0,13228	0	0,111111111	0,05556	0,03704	0,03704	0	0,14286	0,16402
Moyeu-1	0,10185	0,66667	0	0,2037	0,66667	0	0,203703704	0,10185	0,0679	0,0679	0,5	0	1
Cache-1	0,0291	0,19048	0	0,0582	0,19048	0	0,058201058	0,0291	0,0194	0,0194	0,14286	0,28571	0

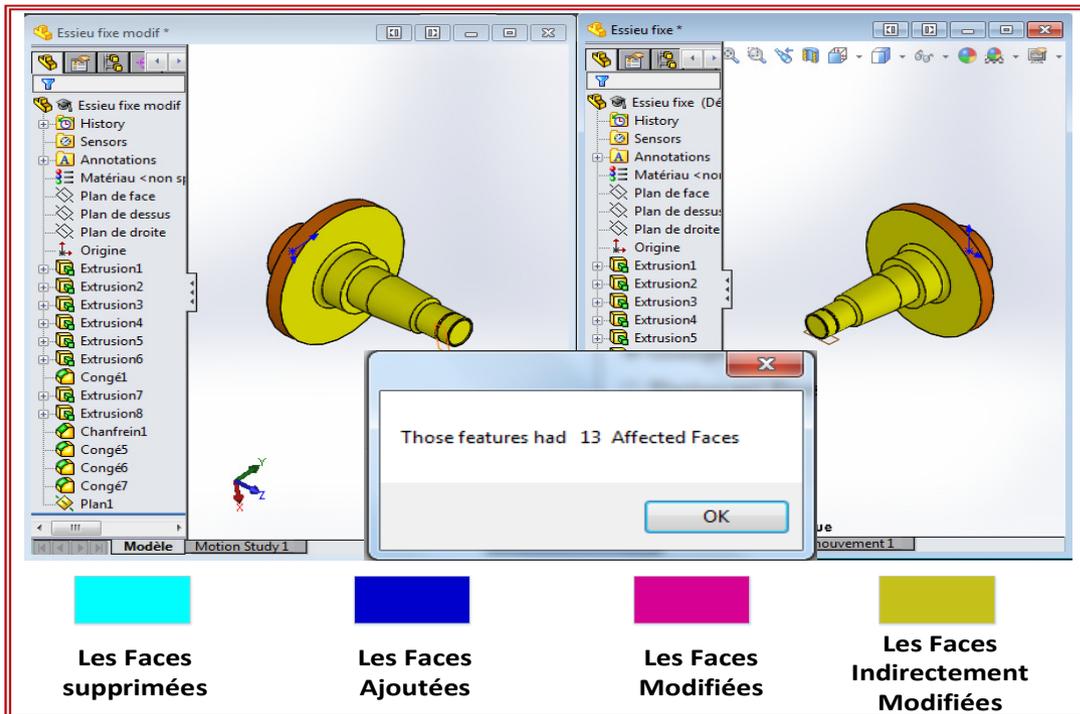
(b) Matrice de Probabilité de Propagations

Fig.4. 14. Matrice d'influence/ Matrice de Probabilité.

### 5.1.2 Comparaison et localisation des modifications



(a) Comparaison des Features



(b) Comparaison de la géométrie

Fig. 4. 15. Résultats de la comparaison des deux versions de l'essieu fixe (Avant/Après modification).

Les résultats de la comparaison montrent que les esquisses : Esquisse 3, Esquisse 4, Esquisse 5, Esquisse 6 et Esquisse 8 de l'Essieu fixe ont été modifiées ce qui a causé la modification indirecte de 13 faces du corps de la pièce. La liste des Features supprimées et celle des Features récemment ajoutées sont toutes les deux vides.

### 5.1.3 Réconciliation des Associations

Pour pouvoir Propager ces modifications apportées à la pièce vers le reste du système on doit passer par l'étape de réconciliation qui nous permet rétablir les liens existents initialement au sein de l'assemblage CAO. Ces liens sont traduits en des associations entre les Features qui forment les différents composants du système. Dans notre cas la Figure (Fig.4. 16) montre les listes des Features Associées obtenues après l'exécution de notre outil développé.

	Niveau 1	Niveau 2	Niveau 3	Niveau 4
List 1	Extrusion3@Essieu fixe-1	Base-Révolution@Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	Base-Révolution@JointSKFCRSH116-1	
List 2	Extrusion4@Essieu fixe-1	Base-Révolution@Roulement_c1ro33-1/Roulement_c1ro_Bint33-1		
List 3	Extrusion6@Essieu fixe-1	Base-Révolution@Roulement_c1ro11-1/Roulement_c1ro_Bint11-1		
List 4	Base-Révolution@Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	Extrusion1@Entretoise-1		
List 5	Extrusion1@Essieu fixe-1	Enlèv. mat.-Extru.1@Entretoise-1	Base-Extrusion@RondFrein8-1	
List 6	Base-Extrusion@RondFrein8-1	Extrusion1@Entretoise-1	Base-Révolution@EcrrouEnc8-1	
List 7	Base-Révolution@EcrrouEnc8-1	Extrusion8@Essieu fixe-1		
List 8	Extrusion8@Essieu fixe-1			
List 9	Base-Révolution@Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	Enlèv. mat.-Extru.3@Moyeu-1	Enlèv. mat.-Extru.3@Moyeu-1	
List 10	Base-Révolution@JointSKFCRSH116-1	Enlèv. mat.-Extru.1@Moyeu-1		
List 11	Base-Révolution@Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	Enlèv. mat.-Extru.6@Moyeu-1	Base-Révolution@Moyeu-1	Base-Révolution@Essieu fixe-1
List 12	Enlèv. mat.-Extru.6@Moyeu-1	Extrusion1@Cache-1		
List 13	Extrusion4@Moyeu-1	Extrusion2@Cache-1		
List 14	Extrusion2@Essieu fixe-1	Extrusion2@Roulement_c1ro33-1/Roulement_c1ro_Bext33-1		
List 15	Base-Révolution@Essieu fixe-1			

SubList 1	Esquisse3@Essieu fixe-1	Esquisse1@Roulement_c1ro33-1/Roulement_c1ro_Bint33-1	Esquisse1@JointSKFCRSH116-1	
SubList 2	Esquisse4@Essieu fixe-1	Esquisse1@Roulement_c1ro33-1/Roulement_c1ro_Bint33-1		
SubList 3	Esquisse6@Essieu fixe-1	Esquisse1@Roulement_c1ro11-1/Roulement_c1ro_Bint11-1		
SubList 4	Esquisse1@Roulement_c1ro11-1/Roulement_c1ro_Bint11-1	Esquisse1@Entretoise-1		
SubList 5	Esquisse1@Essieu fixe-1	Esquisse2@Entretoise-1	Esquisse1@RondFrein8-1	
SubList 6	Esquisse1@RondFrein8-1	Esquisse1@Entretoise-1	Esquisse1@EcrrouEnc8-1	
SubList 7	Esquisse1@EcrrouEnc8-1	Esquisse8@Essieu fixe-1		
SubList 8	Esquisse8@Essieu fixe-1			
SubList 9	Esquisse1@Roulement_c1ro33-1/Roulement_c1ro_Bext33-1	Esquisse6@Moyeu-1	Esquisse6@Moyeu-1	
SubList 10	Esquisse1@JointSKFCRSH116-1	Esquisse5@Moyeu-1		
SubList 11	Esquisse1@Roulement_c1ro11-1/Roulement_c1ro_Bext11-1	Esquisse10@Moyeu-1		
SubList 12	Esquisse10@Moyeu-1	Esquisse1@Cache-1		
SubList 13	Esquisse4@Moyeu-1	Esquisse2@Cache-1		
SubList 14	Esquisse2@Essieu fixe-1			

Fig.4. 16. Les listes des Associations.

### 5.1.4 Risque et Chemin de Propagation de la modification

#### 5.1.4.1. Impact prévisible et risque de la modification

Dans la partie précédente nous avons déterminé deux matrices dites de Probabilité d'Interférence. Ainsi, leur produit nous donne la matrice de risque du système. On s'intéresse principalement au risque engendré par une modification qui touche l'Essieu fixe (Pièce 7).

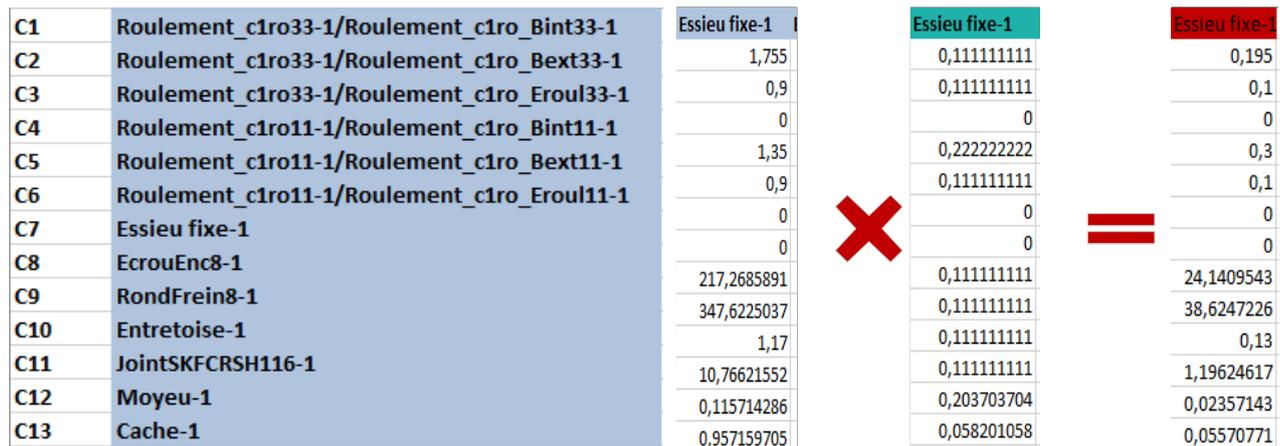


Fig.4. 17. Risque de propagation de modification au sein du système Roue de remorque.

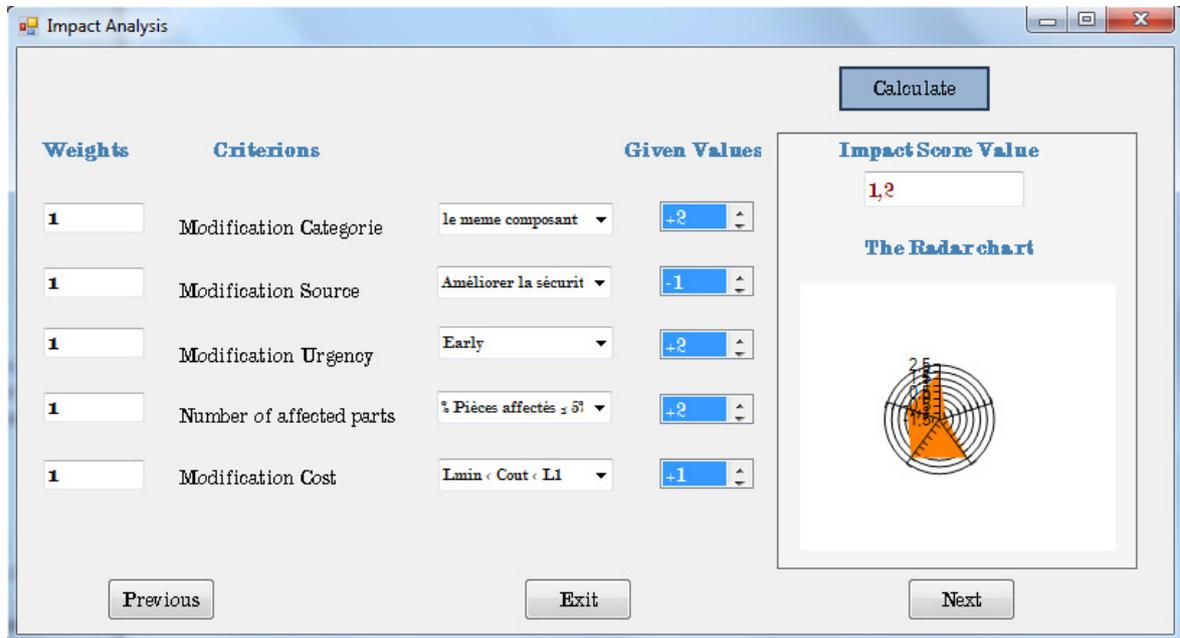


Fig.4. 18. Analyse d'impact de la modification.

La figure (Fig.4. 17) présente les différents facteurs de risque de propagation d'une modification au niveau de l'Essieu fixe ou autre composant du système. On remarque que les pièces C8

(EcrouEnc8) et C9 (RondFrein8) présente un risque très élevé en présence d'une modification. De même pour la pièce C11(JointSKFCRH116) et les deux roulements (avec leurs bagues extérieur et intérieur) mais avec un risque moins élevé. Cette étape nous permet de prévoir les composants qui peuvent être affectés par une modification proposée. Ces résultats sont aussi utilisés pour analyser l'impact de cette dernière.

#### 5.1.4.2. Arbre et chemin de propagation de la modification

Après avoir visualisé l'impact et le risque possible de cette modification, nous avons décidé quand même de continuer. L'étape suivante est donc la détermination des chemins de propagation.

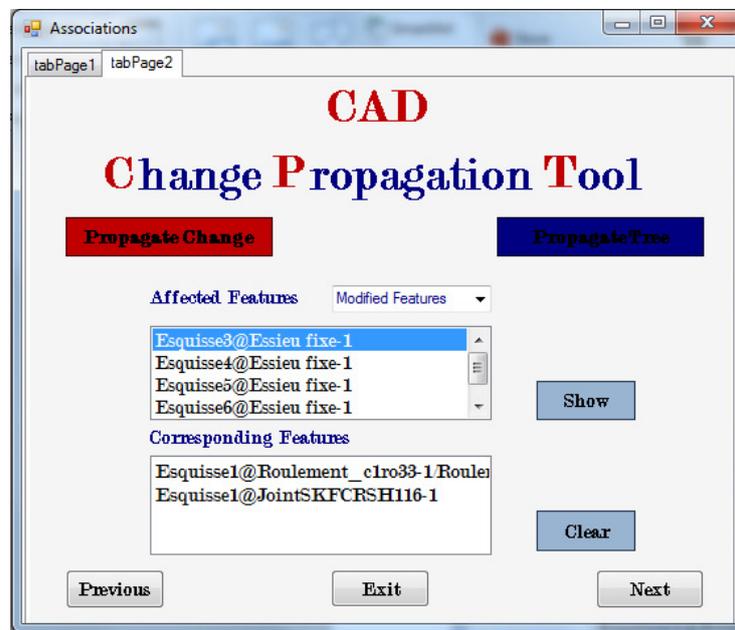


Fig.4. 19. Capture de l'exécution de l'outil "CAD Propagation Tool".

En examinant l'arbre de propagation qui apparaît dans la figure (Fig.4. 20) on remarque l'apparition des composants qui présentent un risque très élevé dans la matrice de risque élaboré précédemment. Par conséquent, la matrice de risque prévoit correctement les propagations possibles. Notre outil « CAD propagation Toll » permet la visualisation des chemins de propagation et des Features susceptibles d'être affectées par cette modification comme le montre la figure (Fig.4. 19).

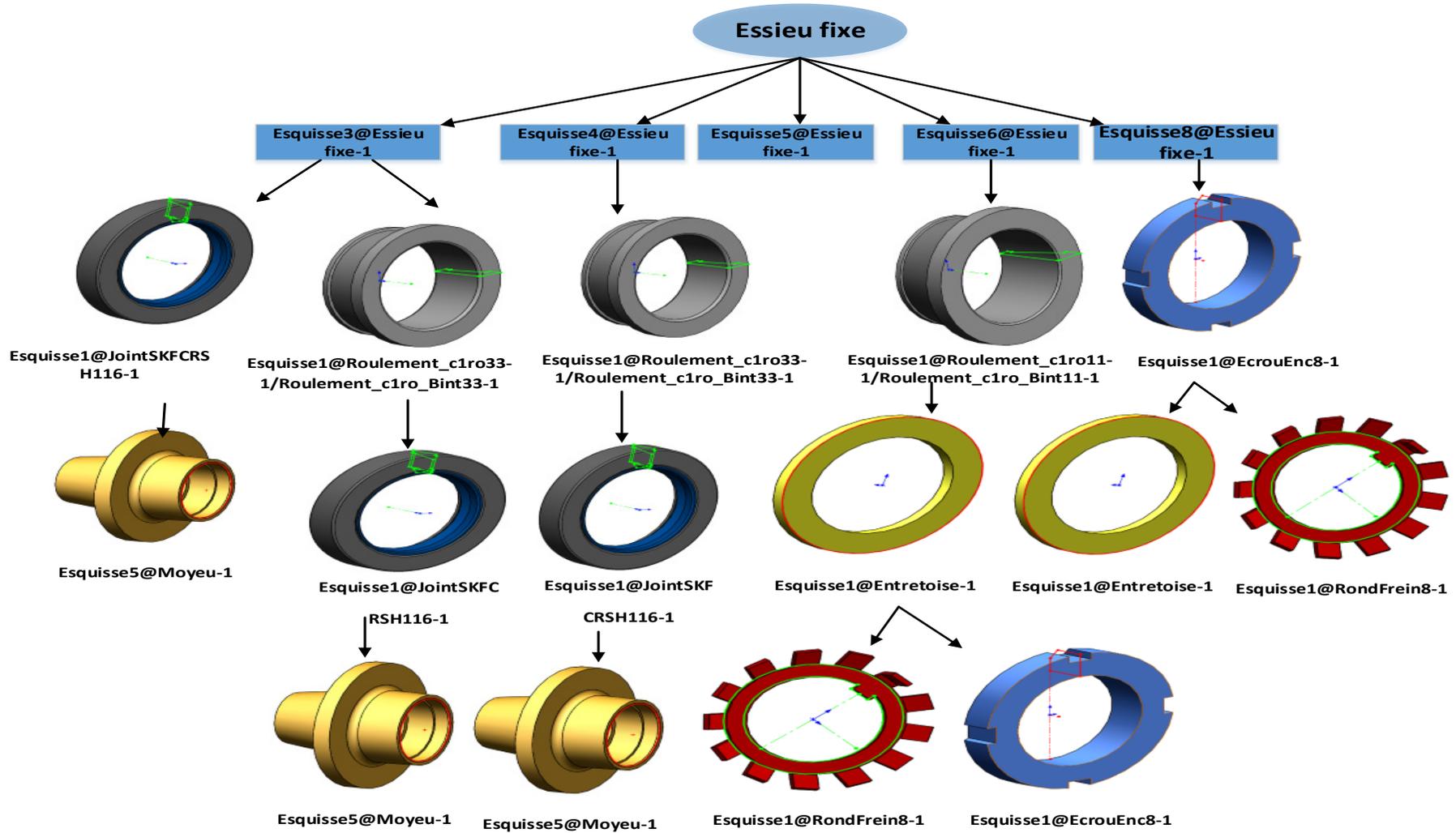


Fig.4. 20. Arbre de propagation de la modification de l'Essieu fixe vers le reste du système.

## 5.2 Deuxième cas d'étude : Mécanisme d'un micromoteur

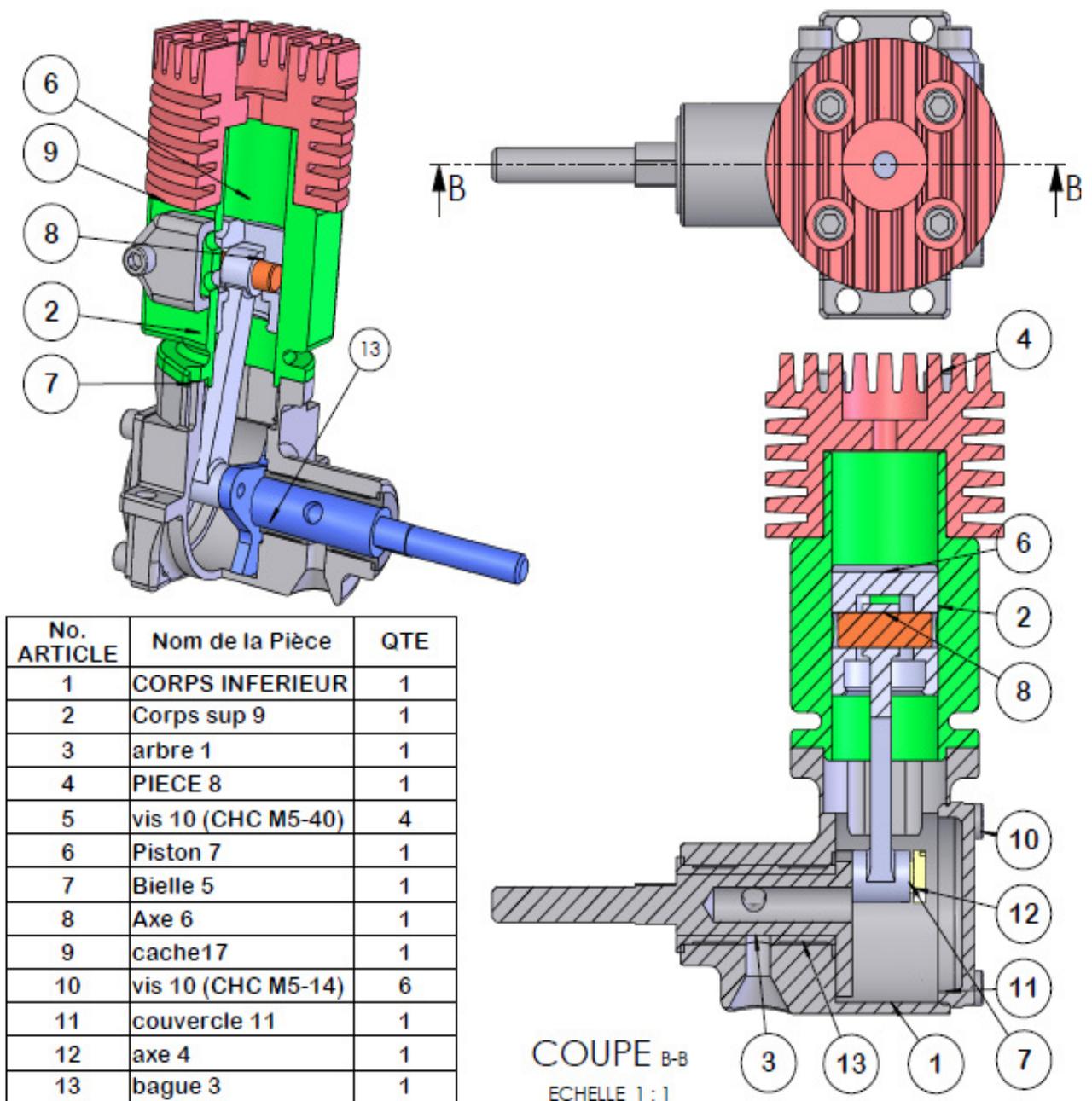
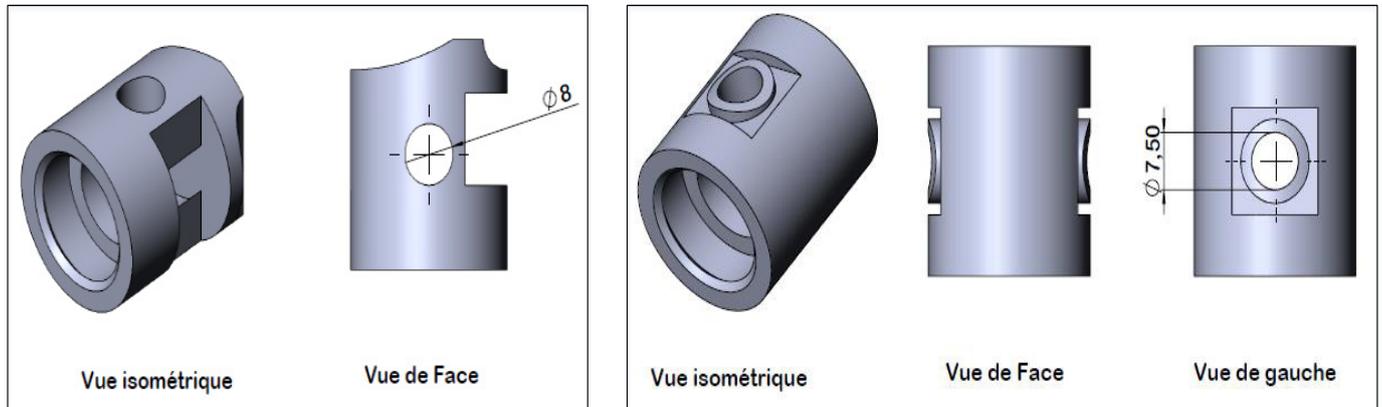


Fig.4. 21. Modèle CAO d'un micromoteur.

Notre deuxième cas d'étude est micromoteur qui est déjà en cours d'utilisation. Ce produit est déjà lancé sur le marché et il est en phase de maturation. En effet, les demandes et les profits sont très importants, mais aussi la concurrence est la plus exacerbée. Afin de se différencier dans cet environnement concurrentiel, une analyse de satisfaction clients a été relancée. D'où il a été détecté qu'après un x temps d'utilisation et durant le fonctionnement du moteur, le piston peut

subir une rupture en charge ou une rupture de fatigue (formation d'une rupture par fissuration jusqu'à la tête du piston). Cette analyse a donc montré la nécessité de modifier la conception du Piston 7 du micromoteur. Les modifications proposées sont montrées dans la figure (Fig.4. 22).



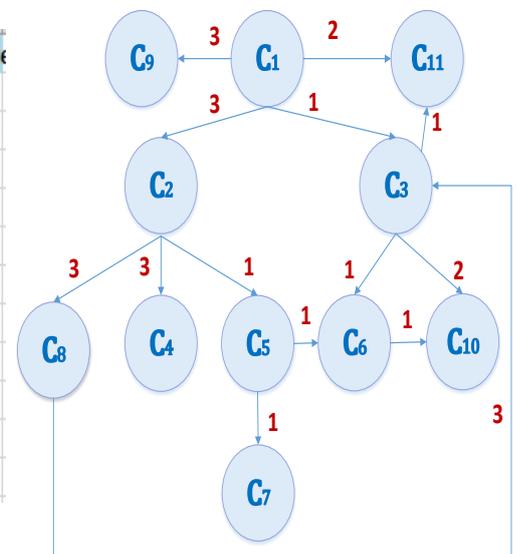
(a) Piston

(b) Piston Modifié

Fig.4. 22. Schéma explicatif de la modification apportée au Piston.

### 5.2.1 L'extraction des données

		CORP	Corps	arbre	PIECE	Pisto	Bielle	Axe	cache	couve	axe 4	bague
C1	CORPS INFERIEUR-1	0	3	1	0	0	0	0	0	3	0	2
C2	Corps sup 9-1	3	0	0	3	1	0	0	3	0	0	0
C3	arbre 1-1	1	0	0	0	0	1	0	0	0	2	1
C4	PIECE 8-1	0	3	0	0	0	0	0	0	0	0	0
C5	Piston 7-1	0	1	0	0	0	1	1	0	0	0	0
C6	Bielle 5-1	0	0	1	0	1	0	0	0	0	1	0
C7	Axe 6-1	0	0	0	0	1	0	0	0	0	0	0
C8	cache17-1	0	3	0	0	0	0	0	0	0	0	0
C9	couvercle 11-1	3	0	0	0	0	0	0	0	0	0	0
C10	axe 4-2	0	0	2	0	0	1	0	0	0	0	0
C11	bague 3-2	2	0	1	0	0	0	0	0	0	0	0



(a) Matrice des Relations

(b) Graphe des liaisons

Fig.4. 23. Graphe des liaisons du Micromoteur et sa matrice d'adjacence.

Puisque notre système est composé de 22 pièces le parcours de graphe des liaisons en largeur nous permet d'obtenir donc le même nombre d'arbres de propagations différentes, en changeant à chaque fois de Source de modification. Pour alléger le calcul dans le cas des mécanismes ayant

un nombre de composant très élevé, nous avons ajouté la possibilité de ne pas prendre les éléments de fixation en considération lors de l'exécution de notre algorithme. Pour le cas du micromoteur on a remarqué la présence de différents types de vis, du coup on a choisi de ne les pas prendre en considération. Le nombre des composants du système et donc réduit à 11 composants (sans considération des vis) dont le graphe de liaison est présenté dans la figure (Fig.4. 23).

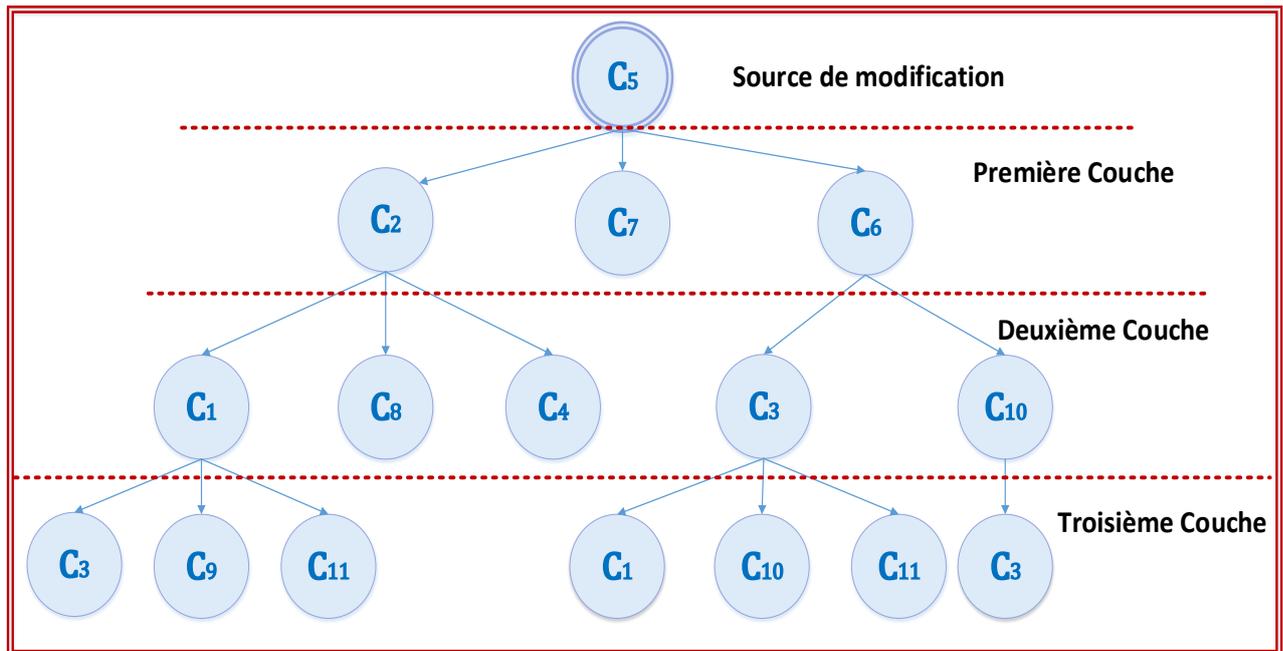


Fig.4. 24. Arbre de propagation des modifications du Piston.

La pièce qui va être modifiée c'est le composant C5« Piston », donc on s'intéresse principalement à l'arbre de propagation des modifications de celui-là (Fig.4. 24) dans le but de déterminer le risque provoqué par une modification qui le touche. On a déterminé par la suite les deux matrices : d'influence et de probabilité de propagation (Fig.4. 29) qui nous a permis de calculer la matrice de risque. Cette dernière à montre que les composant qui ont le risque le plus élevé sont : l'axe 6(axe du piston), la bielle et l'axe4 (axe qui relie la bielle avec le vilebrequin).

	CORPS II	Corps su arbre 1-1	PIECE 8-1	Piston 7-1	Bielle 5-1	Axe 6-1	cache17-1	couvercl	axe 4-2	bague 3-2	
CORPS INFERIEUR-1	0	0,6127	0,48005	0,55807	0,18572	0,1142	0,01178	0,07341	0,22356	0,07625	0,23352109
Corps sup 9-1	0,59215	0	0,28426	0,91082	0,13601	0,11981	0,01923	0,11982	0,13238	0,0325	0,13827994
arbre 1-1	4,12212	1,30976	0	1,19296	0,63727	1,01746	0	0,15694	0,47789	0,34193	0,64116783
PIECE 8-1	0,87567	1,4788	0,34748	0	0,20113	0,15903	0,02844	0,17719	0,19576	0,0325	0,15150827
Piston 7-1	3,47617	2,95615	1,19763	2,69253	0	0,60221	0,1414	0,35421	0,39133	0,39967	0,30286884
Bielle 5-1	9,93153	6,48364	3,57096	0	1,59933	0	0,22614	0	0,64062	0,66367	0,85950006
Axe 6-1	7,10697	9,08768	2,87204	8,27726	3,07416	1,85128	0	1,08889	0,39133	0,81653	0
cache17-1	2,37933	4,01811	0,6828	3,65978	0,5465	0,36701	0,07727	0	0,53191	0,0325	0,41166983
couvercle 11-1	3,99821	2,44972	1,91932	2,23126	0,43557	0,30614	0,01178	0,29353	0	0,24081	0,93366684
axe 4-2	27,9488	8,75337	10,6613	1,19296	4,92433	4,88618	0	0	3,19381	0	4,28504381
bague 3-2	10,4902	4,48152	3,11674	4,08187	0,60953	0,65482	0,01178	0,53698	1,63516	0,36565	0

(a) Matrice des Influences

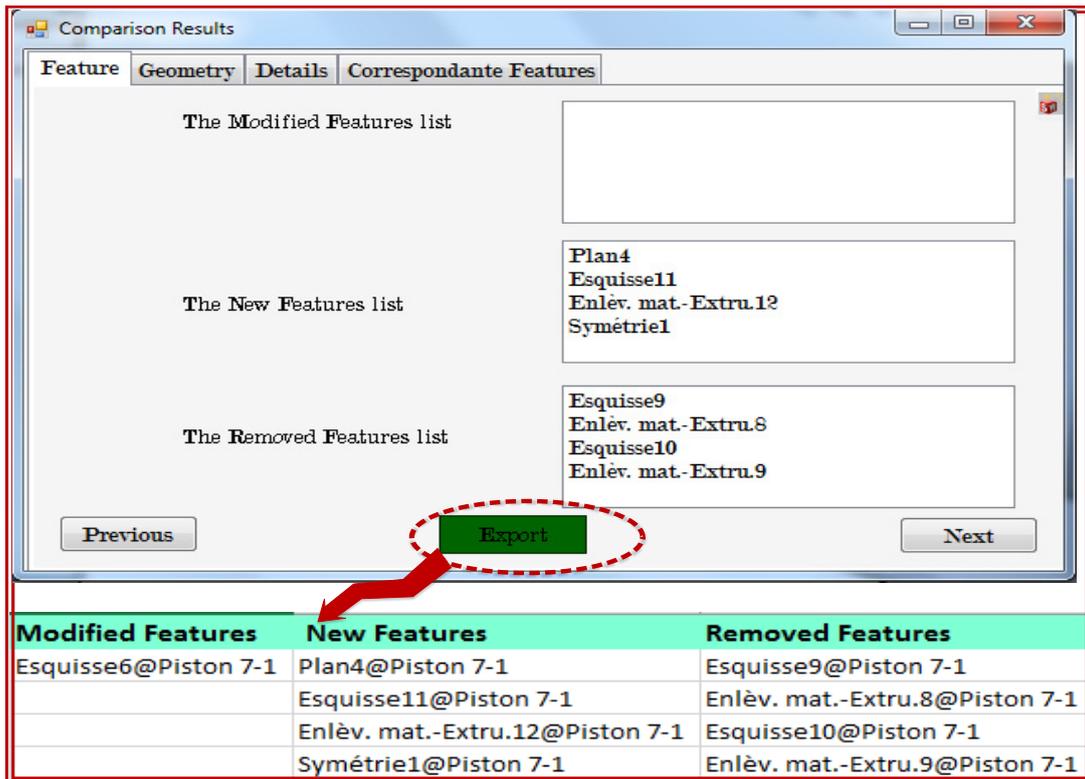
	CORPS II	Corps su arbre 1-1	PIECE 8-1	Piston 7-1	Bielle 5-1	Axe 6-1	cache17-1	couvercl	axe 4-2	bague 3-2	
CORPS INFERIEUR-1	0	0,3	0,2	0,3	0,1	0,06667	0	0,3	1	0,13333	0,66667
Corps sup 9-1	0,33333	0	0,06667	1	0,33333	0,13333	0,33333	1	0,33333	0,04444	0,22222
arbre 1-1	0,11111	0,03333	0	0,03333	0,12222	0,33333	0	0,03333	0,11111	0,66667	0,33333
PIECE 8-1	0,1	0,3	0,02	0	0,1	0,04	0,1	0,3	0,1	0,01333	0,06667
Piston 7-1	0,03333	0,1	0,07333	0,1	0	0,33333	1	0,1	0,03333	0,11556	0,02222
Bielle 5-1	0,03333	0,04	0,2	0,04	0,33333	0	0,33333	0,04	0,03333	0,33333	0,07407
Axe 6-1	0,01111	0,03333	0,02444	0,03333	0,33333	0,11111	0	0,03333	0,01111	0,03852	0,00741
cache17-1	0,1	0,3	0,02	0,3	0,1	0,04	0,1	0	0,1	0,01333	0,06667
couvercle 11-1	0,33333	0,1	0,06667	0,1	0,03333	0,02222	0	0,1	0	0,04444	0,22222
axe 4-2	0,05556	0,02667	0,4	0,02667	0,16	0,33333	0,11111	0,02667	0,05556	0	0,15802
bague 3-2	0,22222	0,07333	0,2	0,07333	0,04667	0,08148	0	0,07333	0,24444	0,16296	0

(b) Matrice de Probabilité de Propagation

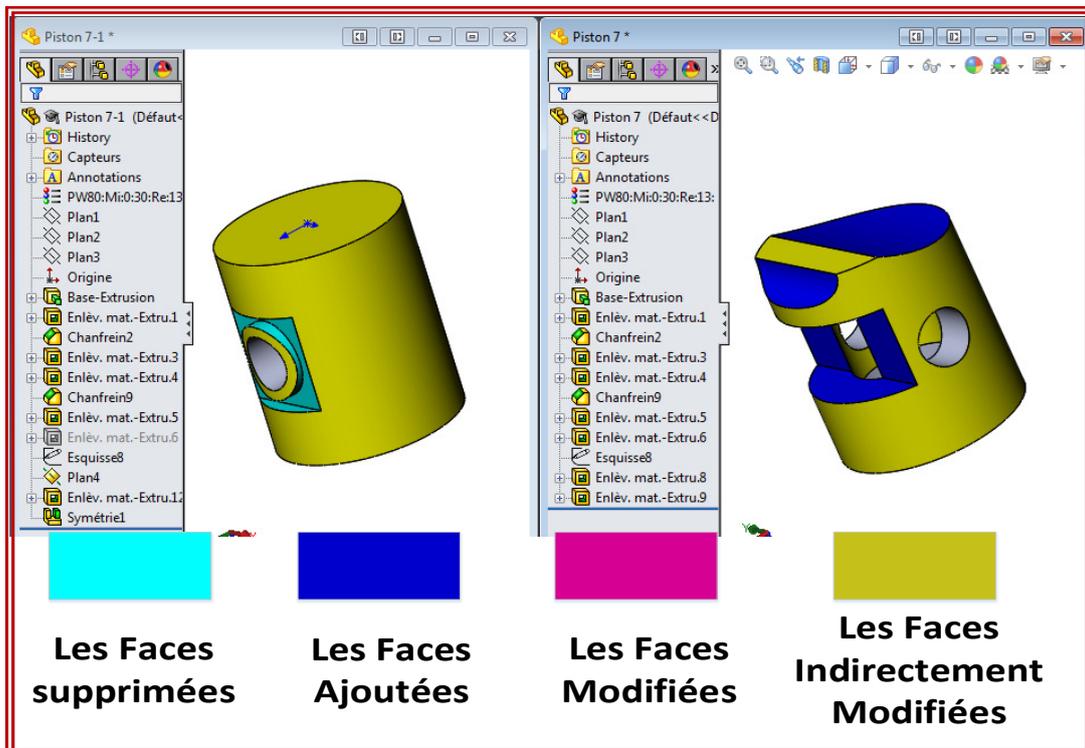
Fig.4. 25. Matrice des Influences et Matrice de Probabilité de Propagation des modifications au sein du mécanisme micromoteur.

### 5.2.2 Comparaison et localisation des modifications

Dans ce deuxième exemple les modifications qui ont été faites sur le Piston-7 se manifestent par la suppression des deux Features « Enlèv. mat.-Extru.8@Piston 7-1 » et « Enlèv. mat.-Extru.9@Piston 7-1 » ainsi que leurs esquisses parentes. Mais en contrepartie la création d'un nouveau Feature « Enlèv. mat.-Extru.12@Piston 7-1 » qui a été symétrisé « Symétrie1@Piston 7-1 ». Des correspondances doivent être faites entre ces nouvelles Features et les Features initiales de la pièce.



(a) Comparaison des Features



(b) Comparaison de la géométrie

Fig.4. 26. Les résultats de comparaison des deux versions du Piston-7.

### 5.2.3 Réconciliation des associations

#### 5.2.3.1. Correspondance des Features

La figure (Fig.4. 27) montre que les deux Features récemment créés « Enlèv. mat.-Extru.12@Piston 7-1 » et « Symétrie1@Piston 7-1 » sont tous les deux associées aux Features « Base-Extrusion@Piston 7-1 » et « Enlèv. mat.-Extru.5@Piston 7-1 ». En effet les nouvelles faces qui ont apparues ont causé des modifications aux faces des Features cités précédemment.

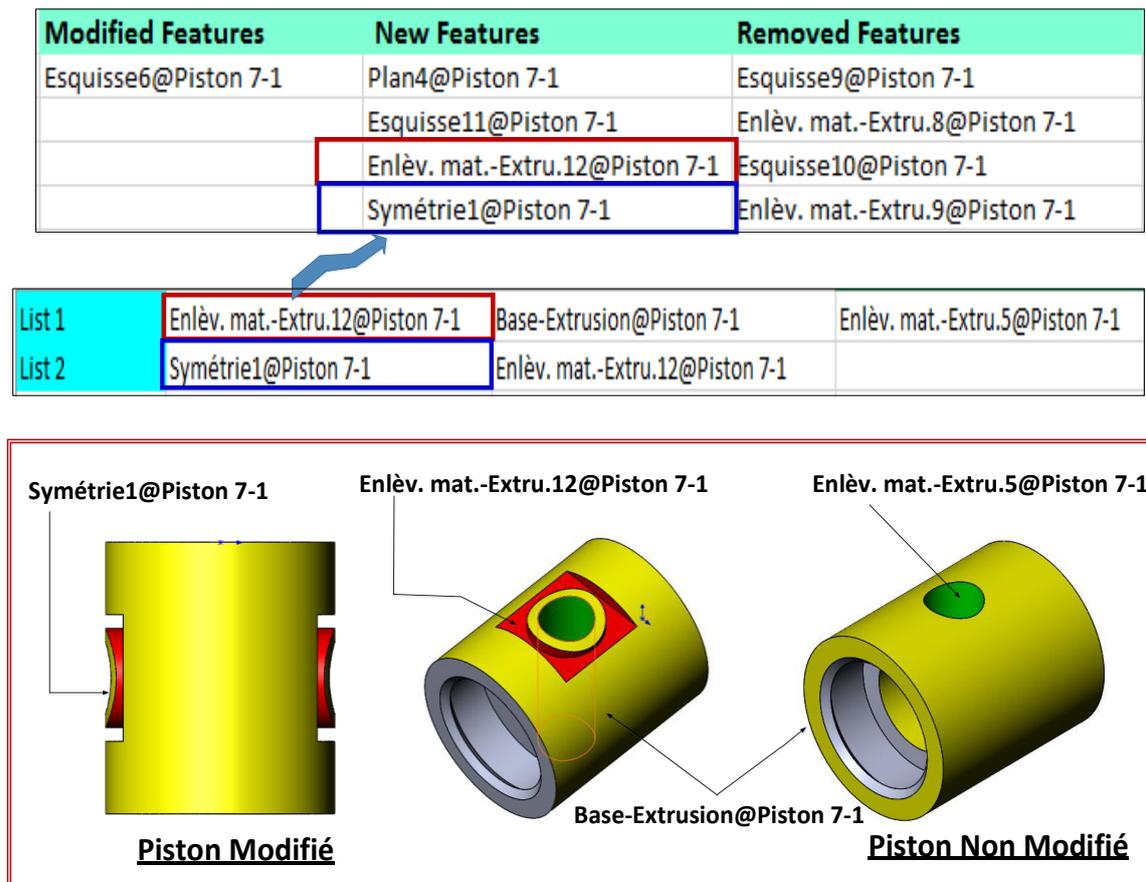


Fig.4. 27. Correspondance des Features.

#### 5.2.3.2. Associations des Features

La figure (Fig.4. 28) montre les différentes associations obtenues entre les Features du système micromoteur. Ces associations en plus des correspondances réalisées, nous permet par la suite d'obtenir l'arbre de propagation des modifications.

List 1	Base-Extrusion@CORPS INFERIEUR-1	Boss.-Extru.3@Corps sup 9-1	Boss.-Extru.3@Corps sup 9-1	
List 2	Enlèv. mat.-Extru.9@CORPS INFERIEUR-1	Enlèv. mat.-Extru.1@Corps sup 9-1		
List 3	Enlèv. mat.-Extru.5@CORPS INFERIEUR-1	Base-Extrusion@arbre 1-1	Boss.-Extru.2@couvercle 11-1	
List 4	Base-Extrusion@Corps sup 9-1	Enlèv. mat.-Extru.5@PIECE 8-1	Base-Extrusion@Piston 7-1	
List 5	Boss.-Extru.1@Corps sup 9-1	Base-Révolution@PIECE 8-1	Base-Révolution@PIECE 8-1	Base-Extrusion@cache17-1
List 6	Répétition circulaire2@PIECE 8-1	Boss.-Extru.1@vis 10 (CHC M5-40)-1	Base-Extrusion@vis 10 (CHC M5-40)-1	
List 7	Enlèv. mat.-Extru.5@Piston 7-1	Enlèv. mat.-Extru.1@Bielle 5-1		
List 8	Base-Extrusion@arbre 1-1	Boss.-Extru.3@Bielle 5-1		
List 9	Boss.-Extru.3@Piston 7-1	Base-Extrusion@Axe 6-1		
List 10	Perçage7@Corps sup 9-1	Enlèv. mat.-Extru.2@cache17-1		
List 11	Perçage6@Corps sup 9-1	Enlèv. mat.-Extru.2@cache17-1		
List 12	Boss.-Extru.3@cache17-1	Boss.-Extru.1@vis 10 (CHC M5-14)-1		
List 13	Base-Extrusion@cache17-1	Base-Extrusion@vis 10 (CHC M5-14)-1	Boss.-Extru.1@vis 10 (CHC M5-14)-2	
List 14	Boss.-Extru.1@cache17-1	Base-Extrusion@vis 10 (CHC M5-14)-2		
List 15	Boss.-Extru.2@CORPS INFERIEUR-1	Base-Extrusion@couvercle 11-1	Base-Extrusion@couvercle 11-1	
List 16	Base-Extrusion@couvercle 11-1	Boss.-Extru.1@vis 10 (CHC M5-14)-3	Base-Extrusion@vis 10 (CHC M5-14)-3	
List 17	Enlèv. mat.-Extru.1@arbre 1-1	Boss.-Extru.1@axe 4-2		
List 18	Boss.-Extru.1@arbre 1-1	Boss.-Extru.1@axe 4-2	Enlèv. mat.-Extru.1@bague 3-2	
List 19	Enlèv. mat.-Extru.2@Bielle 5-1	Enlèv. mat.-Extru.2@axe 4-2		
List 20	Enlèv. mat.-Extru.7@CORPS INFERIEUR-1	Boss.-Extru.2@bague 3-2		
List 21	Enlèv. mat.-Extru.11@CORPS INFERIEUR-1	Enlèv. mat.-Extru.2@bague 3-2		

SubList 1	Esquisse6@CORPS INFERIEUR-1	Esquisse4@Corps sup 9-1	Esquisse4@Corps sup 9-1	
SubList 2	Esquisse23@CORPS INFERIEUR-1	Esquisse6@Corps sup 9-1		
SubList 3	Esquisse19@CORPS INFERIEUR-1	Esquisse1@arbre 1-1	Esquisse4@couvercle 11-1	
SubList 4	Esquisse1@Corps sup 9-1	Esquisse5@PIECE 8-1	Esquisse1@Piston 7-1	
SubList 5	Esquisse2@Corps sup 9-1	Esquisse1@PIECE 8-1	Esquisse1@PIECE 8-1	Esquisse1@cache17-1
SubList 6		Esquisse3@vis 10 (CHC M5-40)-1	Esquisse1@vis 10 (CHC M5-40)-1	
SubList 7	Esquisse6@Piston 7-1	Esquisse5@Bielle 5-1		
SubList 8	Esquisse1@arbre 1-1	Esquisse4@Bielle 5-1		
SubList 9		Esquisse1@Axe 6-1		
SubList 10	Esquisse29@Corps sup 9-1	Esquisse4@cache17-1		
SubList 11	Esquisse27@Corps sup 9-1	Esquisse4@cache17-1		
SubList 12		Esquisse3@vis 10 (CHC M5-14)-1		
SubList 13	Esquisse1@cache17-1	Esquisse1@vis 10 (CHC M5-14)-1	Esquisse3@vis 10 (CHC M5-14)-2	
SubList 14		Esquisse1@vis 10 (CHC M5-14)-2		
SubList 15	Esquisse9@CORPS INFERIEUR-1	Esquisse1@couvercle 11-1	Esquisse1@couvercle 11-1	
SubList 16	Esquisse1@couvercle 11-1	Esquisse3@vis 10 (CHC M5-14)-3	Esquisse1@vis 10 (CHC M5-14)-3	
SubList 17	Esquisse2@arbre 1-1	Esquisse2@axe 4-2		
SubList 18	Esquisse3@arbre 1-1	Esquisse2@axe 4-2	Esquisse7@bague 3-2	
SubList 19	Esquisse6@Bielle 5-1	Esquisse11@axe 4-2		
SubList 20	Esquisse21@CORPS INFERIEUR-1	Esquisse3@bague 3-2		
SubList 21	Esquisse32@CORPS INFERIEUR-1	Esquisse8@bague 3-2		

Fig.4. 28. La liste des Features Associées du système Micromoteur.

## 5.2.4 Risque et Chemin de Propagation de la modification

### 5.2.4.1 Impact prévisible et risque de la modification

La matrice de Risque de Propagation de Modification au sein du micromoteur (Fig.4. 29) montre bien que le composant le plus risqué par une modification affectant le composant « Piston7 » c'est l'axe du piston « Axe-6 ». Également, les composants « Bielle 5 » et « Axe 4 » sont aussi susceptibles d'être affectés d'après la matrice de risque avec des degrés moins importants. Aussi, les autres composants d'assemblage peuvent être affectés.

		Piston 7-		Piston 7-		Piston 7-1
C1	CORPS INFERIEUR-1	0,18572		0,1		0,01857188
C2	Corps sup 9-1	0,13601		0,33333		0,04533622
C3	arbre 1-1	0,63727		0,12222		0,07788851
C4	PIECE 8-1	0,20113		0,1		0,02011293
C5	Piston 7-1	0		0		0
C6	Bielle 5-1	1,59933		0,33333		0,53311165
C7	Axe 6-1	3,07416		0,33333		1,02471883
C8	cache17-1	0,5465		0,1		0,05464973
C9	couvercle 11-1	0,43557		0,03333		0,01451897
C10	axe 4-2	4,92433		0,16		0,78789216
C11	bague 3-2	0,60953		0,04667		0,04469854

Fig.4. 29. Risque de propagation de modification au sein du système d'un micromoteur.

#### 5.2.4.2 Arbre et chemin de propagation de la modification du Piston 7

Bien évidemment l'arbre de propagation des modifications de la figure (Fig.4. 30) montre que les composants ayant le risque le plus élevé d'être touché (Axe 6, Bielle 5, Corps supérieur 9 et la Pièce 8) par une modification au niveau du piston sont effectivement risqués et la propagation va les touchés. D'autre composant du système présente aussi un risque important mais n'ont pas être touché par la propagation car la modification proposée n'a pas touché les Features avec lesquelles leurs Feature à eux sont associées.

Les Features du Piston qui ont été supprimées n'ont pas de propagation possible car ce sont des Feature qui ne sont pas engagés dans des contraintes d'assemblages. Par conséquent on ne trouve pas ces dernières dans la liste des Features associées. Ce qui signifie que leur suppression ne va affecter aucun autre composant du système.

## 6. Conclusion

La première partie de ce chapitre a été consacrée pour présenter les outils utilisés (logiciels et langage de programmation). La deuxième partie est dédiée à la présentation des interfaces qui ont été implémentés. Tandis que la troisième partie est une partie de validation ou deux cas d'études sont présentés afin de tester le fonctionnement de l'outil développé. L'implémentation de notre approche proposée nous a permis de créer un add-in qui a été ajouté à SolidWorks et qui est facile à manipuler. Les deux cas d'étude utilisés ont bien montré le bon fonctionnement de toutes les interfaces de l'outil implémenté. Ainsi que la validité des algorithmes sur lesquels l'approche est fondue. En effet les arbres de propagation obtenues à la fin de chaque exemple et

la matrice de risque déterminée sont parfaitement en accord. Ainsi nous sommes capables de prévoir l'impact d'une modification sur le reste des composants d'un assemblage. Par conséquent la tâche de gestion des modifications est devenue plus efficace (sans la nécessité d'utiliser aucun autre logiciel ou outil de gestion). L'outil développé a été intégré dans SolidWorks mais elle peut être adaptée et intégrée sous n'importe quel autre Système de conception assistée par ordinateur comme CATIA par exemple. Ainsi, l'add-in implémenté peut être intégré dans les systèmes PLM et ajoute ainsi une fonction de plus à ces systèmes et facilite la gestion des données CAO.

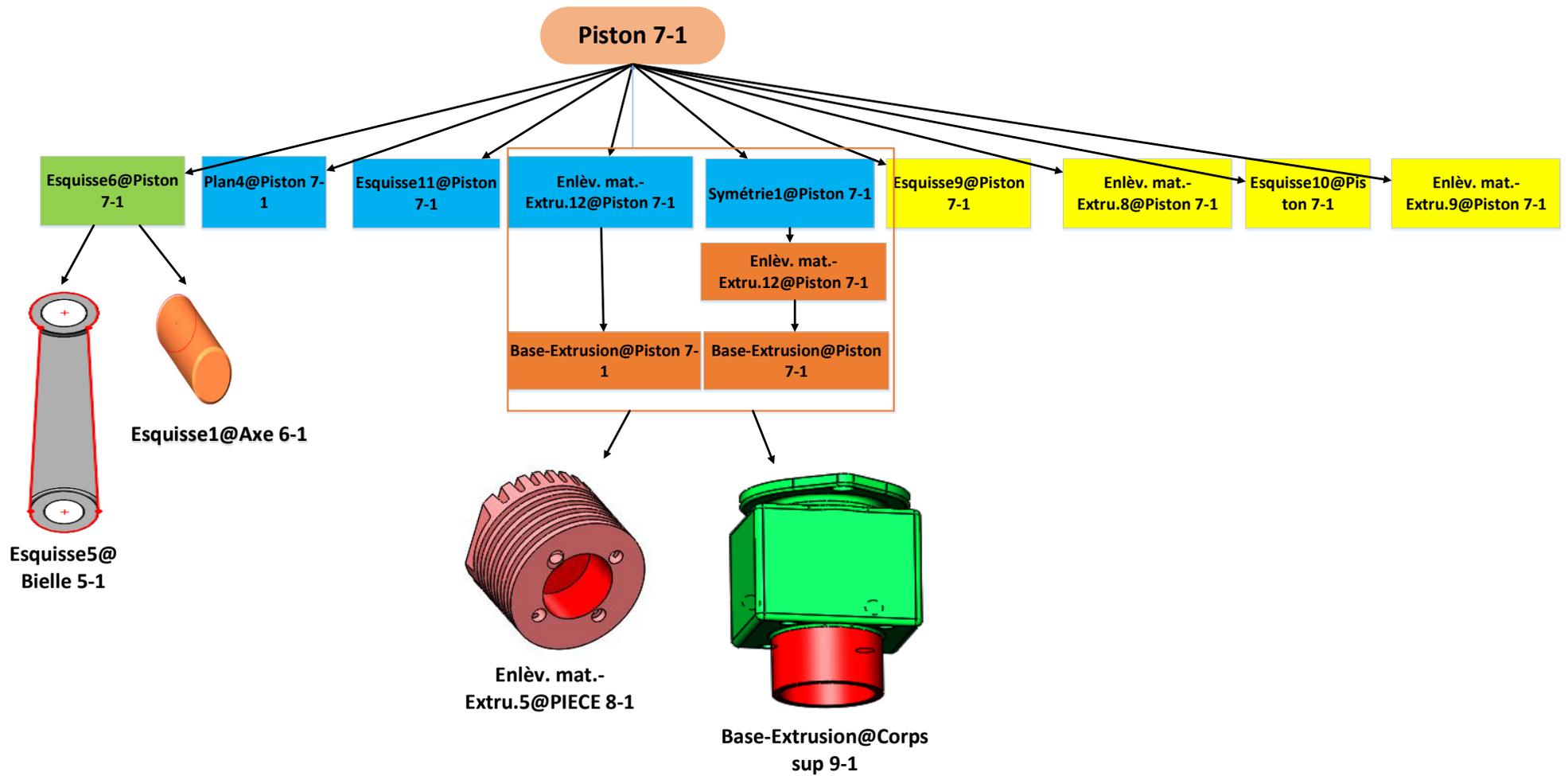


Fig.4. 30. Arbre de propagation d'une modification qui a touché le piston vers le reste du système micromoteur.

## Conclusion générale

L'objectif principal d'un concepteur est de trouver une solution qui répond aux besoins et aux attentes du client. Ceci est le cas de la ré-conception ou de l'amélioration d'un produit déjà existant, en apportant des modifications à son modèle actuel (appelés changements d'ingénierie). D'où la nécessité d'une gestion adéquate afin d'avoir des résultats bénéfiques sur le produit en améliorant son coût de revient, sa qualité mais aussi en respectant le délai de sa mise sur le marché. La gestion de ces changements dites d'ingénierie est considérée parmi les problématiques les plus compliquées. En effet le changement réalisé doit nécessairement être propagé dans le reste du produit pour avoir la cohérence des données. Cette thèse rentre dans ce cadre par la proposition d'un outil de gestion et de propagation des modifications dans un produit mécanique. Le but est d'évaluer l'impact d'un changement d'ingénierie apporté à un produit et de prédire sa propagation.

Dans ce mémoire de thèse, on a répondu à Cinq différentes questions qui ont été déjà posées dans le premier chapitre. La motivation industrielle, la problématique et la structure de la thèse ont été discutées et détaillées grâce à un schéma explicatif dans la première partie du premier chapitre. Dans la deuxième partie de ce chapitre, des concepts différents ont été définis. On a présenté les différentes étapes de la conception du produit, la définition de la conception assistée par ordinateur (CAO) ainsi que les différents modèles tridimensionnels. Cette section a été suivie d'une revue des différentes définitions des changements d'ingénierie et de la propagation des changements d'ingénierie qui existent dans la littérature. Cette réflexion a permis de bien comprendre la notion de propagation de modification et de concrétiser les différents contextes d'apparition de ce phénomène. À la fin de cette partie on a pu formuler une idée complète de la définition des changements d'ingénierie et du type de modélisation CAO tridimensionnelle qu'on va les utiliser dans le reste du travail.

Dans le deuxième chapitre nous avons abordé une étude bibliographique sur la gestion des changements d'ingénierie (ECM). Nous avons commencé par les systèmes PLM leur utilité et leurs limitations qui ont mené à l'apparition des nouveaux systèmes et méthodes de gestion des modifications. Ces dernières commencent généralement par la caractérisation des changements. En effet l'évaluation de l'impact d'un changement n'est possible que si la caractérisation des

changements, quantitativement et/ou qualitativement est faite. Les méthodes de gestion présentée dans ce chapitre sont classées en trois catégories : Méthodes traditionnelles (ou méthodes qui se concentrent principalement sur une seule couche de produit), Méthodes multi-compagnies et Méthodes multicouches. Aussi les méthodes/modèles de propagation des changements présentés sont aussi classé en : méthodes matricielles, méthodes nécessitant une base de données et méthodes basées sur d'autres concepts sous-jacents.

Nous avons constaté que tous les éléments des systèmes présentés dans la partie bibliographique sont généralement liés par des relations déjà connues grâce à un modèle théorique ou obtenues grâce aux experts par exemple. Cependant, l'obtention de ce modèle de dépendance est difficile si le modèle théorique n'est pas disponible ou s'il est incomplet. Cela nous a conduits à réfléchir au recours aux données internes propres du modèle CAO pour en faire un modèle de dépendance. La dernière partie de ce chapitre été totalement dédié aux outils de comparaison des modèle CAO et leur utilité, ainsi qu'une description des outils disponible sous SolidWorks et Catia V5. Cette partie s'est terminée par une conclusion par rapport aux deux outils examinés. Nous nous sommes alors trouvés dans la nécessité de développer notre propre outil de comparaison afin de compenser les limitations des outils déjà disponibles.

Le troisième chapitre a été consacré au développement des différents algorithmes qui constitue la structure de base de notre outil élaboré. L'algorithme de dépendance, l'algorithme de comparaison, l'algorithme de réconciliation et l'algorithme propagation. Le premier algorithme dit de dépendance commence tout d'abord par l'extraction des données du modèle CAO. Toutes les données extraites : Contraintes d'assemblage (type de contrainte, entité de contrainte, nature des entités...), interférences, volumes et surfaces, sont conservés dans une base de données. Afin de donner un aspect visuel à ces données là et de simplifier la compréhension du modèle un graphe de liaisons a été construit. Ce graphe nous à servit par la suite aussi pour construire l'arbre de propagation d'un changement. Supposons qu'un composant est affecté par une modification donnée qu'elles sont les autres composant affectés par cette modification ? L'arbre de propagation nous permet de répondre à cette question.

L'algorithme de parcours en largeur ou le BFS permet de parcourir un graphe ou un arbre en commençant par un nœud source puis ses successeurs qui vont eux aussi être considéré comme des nouvelles sources. Le BFS s'est avéré performant et sa définition s'adapte bien avec le

phénomène de propagation nous avons donc adopté par la suite de notre travail le principe de la recherche en largeur pour construire notre arbre de propagation à partir du graphe de liaison.

Pour pouvoir gérer un changement et bien évaluer l'impact de sa propagation, nous avons déterminé le degré d'influence d'une modification affectant un composant sur le reste des composants de l'assemblage. C'est la caractérisation quantitative du changement. Notre modèle de dépendance crée est principalement basé sur la dépendance surfacique, la dépendance volumique et la dépendance relationnelle. Cependant, trois types d'influence sont distingués : influence directe, influence indirecte et influence mixte. Ce modèle de dépendance crée nous a servis par la suite pour déterminer la matrice d'influence entre les différents composant d'un assemblage. Cette dernière a été multipliée par la matrice de probabilité de propagation pour avoir la matrice de risque d'un assemblage.

Le deuxième algorithme qui a été développé dans le deuxième chapitre c'est l'algorithme de comparaison. Le troisième algorithme c'est l'algorithme de réconciliation qui se base principalement sur la mise en œuvre des associations initiales (première étape) existantes entre les Features des différents composants du système. Les contraintes d'assemblage sont traduites en associations. La deuxième étape c'est la mise en œuvre des correspondances entre les nouvelles Features et celles déjà existantes. C'est une étape primordiale qui facilite la réconciliation des associations qui se fait en exploitant les deux étapes précédentes ainsi que les résultats de comparaison. La dernière partie c'est la propagation qui commence par l'analyse d'impact d'une modification qui aide le concepteur à décider par rapport à la modification. Une fois la modification est validée, nous créons alors l'arbre pondéré de propagation des modifications. Cet arbre donne les Features susceptibles d'être affectées par la modification proposée. Les arcs de l'arbre représentant les chemins de propagation sont pondérés avec des valeurs qui mesurent le risque de cette propagation.

Le quatrième chapitre est un chapitre qui montre bien les différentes interfaces de notre outil proposé ainsi que les outils utilisés dans une première partie. La deuxième partie est consacrée pour la validation de notre outil appelé « CAD CPT ». La validation est faite sur différents assemblages mécaniques avec un nombre de pièce et une complexité variante.

- Les principales contributions et apports de nos travaux de recherche sont les suivants :

- La définition d'un modèle de dépendance : le modèle de dépendance créé est un modèle qui se base purement sur les données extraites du modèle CAO. Il ne dépend d'aucun facteur ou élément extérieur par exemple expertise ou historique.
  - La création d'une matrice d'influence qui décrit les dépendances existantes entre les différents composants d'un système est en elle-même une contribution qui donne une idée claire sur les influences existantes dans un système avant même de proposer une modification.
  - La matrice de probabilité de propagation peut aussi être considérée comme une contribution car elle peut être utilisée indépendamment des autres paramètres pour juger l'impact prévisible d'une modification qui touche un composant.
  - La création d'un outil de comparaison basé sur la combinaison de la comparaison des éléments géométriques et celle des Features donne des résultats très satisfaisants. Cet outil permet aussi le calcul du taux de modification d'une pièce après sa modification. L'algorithme de comparaison proposé peut aussi servir à la détection des similarités, c'est une approche très importante qui peut servir dans le cas de la ré-conception et qui évite la conception de certains modèles de zéro.
  - L'implémentation de l'approche proposée et la création de l'outil « CAD CPT » est un grand ajout dans le domaine de gestion des changements d'ingénierie et de propagation des modifications. C'est un outil qui peut être utilisé comme un SolidWorks add-in et qui facilite énormément la tâche du designer, lui offre une aide à la décision et lui permet de gagner du temps qui coûte bien évidemment de l'argent.
- Néanmoins, ces travaux de recherche présentent aussi des limitations que nous visons améliorer dans les travaux futurs.
- Le modèle de dépendance est basé uniquement sur 3 types de dépendance surfacique, volumique et relationnelle. On peut ajouter d'autres types de dépendance (exp : fonctionnel, tolérance.) afin d'améliorer le modèle de dépendance et de le rendre plus performant.
  - Les assemblages testés ne présentent pas un nombre de pièces très important. Tandis que l'augmentation du nombre de pièces de l'assemblage rend le graphe plus complexe et le calcul plus lent.
  - Le code implémenté peut être allégé (certaines procédures et fonctions peuvent être utiles) pour rendre la réponse plus rapide et le calcul plus précis.

## Références

- [1] S. St-Martin, “Maîtrise Du Changement Dans Les Modèles Cao [Maîtrise En Technologie Des Systèmes],” L’école De Technologie Supérieure, Université Du Québec, 2001.
- [2] B. Louhichi and L. Rivest, “Maintaining consistency between CAD elements in collaborative design using association management and propagation,” *Comput. Ind.*, vol. 65, no. 1, pp. 124–135, 2014.
- [3] M. Masmoudi, “Contribution au développement d’une méthode analytique pour l’identification des dépendances et la propagation des changements d’ingénierie en re-conception de produits [ thèse de doctorat],” L’université Paris-Saclay, 2016.
- [4] L. Yin, D. Tang, Y. Kang, and S. Leng, “Topology face-based change propagation analysis in aircraft assembly tooling design,” *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 230, no. 1, pp. 120–135, 2016.
- [5] “AIAG.org - Automotive Industry Action Group.” [Online]. Available: <https://www.aiag.org/>. [Accessed: 13-Jun-2019].
- [6] A. Maier and S. Langer, *Engineering change management report 2011: Survey results on causes and effects, current practice, problems, and strategies in Denmark*. Technical University of Denmark (DTU), 2011.
- [7] E. I. Potoczak, “Gestion du changement d’ingénierie: réalisez des économies grâce à des processus améliorés.” [Online]. Available: <https://erpblog.iqms.com/engineering-change-management-savings/>. [Accessed: 15-Sep-2020].
- [8] Martin Fleurent, “Amélioration du Processus de Gestion Des Demandes de Changement D’ingénierie dans Le Domaine De La Construction [Maîtrise Ès Sciences Appliquées],” Université De Montréal/ École Polytechnique De Montréal, 2013.
- [9] W. F. Rowell, A. H. B. Duffy, I. M. Boyle, and N. Masson, “The nature of engineering change in a complex product development cycle,” in *Systems Engineering Research*, 2009, p. 8.
- [10] E. Fricke, B. Gebhard, H. Negele, and E. Igenbergs, “Coping with changes: Causes, findings, and strategies,” *Syst. Eng.*, vol. 3, no. 4, pp. 169–179, 2000.
- [11] F. Demoly, O. Dutartre, X. T. Yan, B. Eynard, D. Kiritsis, and S. Gomes, “Product relationships management enabler for concurrent engineering and product lifecycle management,” *Comput. Ind.*, vol. 64, no. 7, pp. 833–848, 2013.
- [12] C. H. Loch and C. Terwiesch, “Communication and Uncertainty in Concurrent Engineering,” *Manage. Sci.*, vol. 44, no. 8, pp. 1032–1048, Aug. 1998.
- [13] université de valenciennes, “Conception,” 2013. [Online]. Available: <http://stockage.univ-valenciennes.fr/EcoPEM/BoiteK/co/Introduction.html>. [Accessed: 06-Jul-2020].

- [14] “ISO 9000:2005 - Systèmes de management de la qualité -- Principes essentiels et vocabulaire.” [Online]. Available: <https://www.iso.org/fr/standard/42180.html>. [Accessed: 12-Feb-2019].
- [15] “Les étapes du développement de produit : conception, simulation, prototypage et essais.” [Online]. Available: <http://www.cetim-ceremat.com/fr/conception/>. [Accessed: 02-Jul-2019].
- [16] Lyrfac, “Conception assistée par ordinateur.”
- [17] D. Givert, “Modélisation géométrique d’objets en environnement robotique [thèse de doctorat],” Université de Paris-Sud. Faculté des Sciences d’Orsay (Essonne), 1988.
- [18] B. Louhichi, “Intégration CAO / Calcul par reconstruction du modèle CAO / Calcul par reconstruction du modèle CAO à partir des résultats éléments finis [thèse de doctorat],” Ecole Nationale d’Ingénieurs de Monastir, 2008.
- [19] G. G. Dominique Taraud, *Le guide de la CAO*, vol. 192. Paris: Dunod. Print, 2008.
- [20] T. A. W. Jarratt, C. M. Eckert, N. H. M. Caldwell, and P. J. Clarkson, “Engineering change: An overview and perspective on the literature,” *Res. Eng. Des.*, vol. 22, no. 2, pp. 103–124, 2011.
- [21] I. C. Wright, “A review of research into engineering change management: implications for product design,” *Des. Stud.*, vol. 18, no. 1, pp. 33–42, 1997.
- [22] C. Terwiesch and C. H. Loch, “Managing the process of engineering change orders: the case of the climate control system in automobile development,” *J. Prod. Innov. Manag.*, vol. 16, no. 2, pp. 160–172, 1999.
- [23] G. Q. Huang, W. Y. Yee, and K. L. Mak, “Development of a web-based system for engineering change management,” *Robot. Comput. Integr. Manuf.*, vol. 17, no. 3, pp. 255–267, 2001.
- [24] G. Q. Huang and K. L. Mak, “Current practices of engineering change management in UK manufacturing industries,” *International Journal of Operations and Production Management*, vol. 19, no. 1, pp. 21–37, 1999.
- [25] G. Q. Huang, W. Y. Yee, and K. L. Mak, “Current practice of engineering change management in Hong Kong manufacturing industries,” *J. Mater. Process. Technol.*, vol. 139, no. 1–3 SPEC, pp. 481–487, 2003.
- [26] E. Fricke and A. P. Schulz, “Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle,” *Syst. Eng.*, vol. 8, no. 4, pp. 342–359, 2005.
- [27] J. Tavčar and J. Duhovnik, “Engineering change management in individual and mass production,” *Robot. Comput. Integr. Manuf.*, vol. 21, no. 3, pp. 205–215, 2005.
- [28] A. Siddiqi, G. Bounova, O. L. de Weck, R. Keller, and B. Robinson, “A Posteriori Design Change Analysis for Complex Engineering Projects,” *J. Mech. Des.*, vol. 133, no. 10, p. 101005, 2011.

- [29] G. Vianello and S. Ahmed-Kristensen, “A comparative study of changes across the lifecycle of complex products in a variant and a customised industry,” *J. Eng. Des.*, vol. 23, no. 2, pp. 99–117, Feb. 2012.
- [30] B. Hamraz, “Engineering Change Modelling Using a Function-Behaviour-Structure Scheme [thèse de doctorat],” the University of Cambridge, 2013.
- [31] C. Eckert, P. J. Clarkson, and W. Zanker, “Change and customisation in complex engineering domains,” *Res. Eng. Des.*, vol. 15, no. 1, pp. 1–21, 2004.
- [32] M. DiPrima, “Engineering change control and implementation considerations,” *Prod. Invent. Manag.*, vol. 23, no. 1, pp. 81–87, 1982.
- [33] M. A. Reidelbach, “Engineering change management for long-lead-time production environments,” vol. 32, no. 2, pp. 84–88, Dec. 1991.
- [34] E. C. Y. Koh, N. H. M. Caldwell, and P. J. Clarkson, “A method to assess the effects of engineering change propagation,” *Res. Eng. Des.*, vol. 23, no. 4, pp. 329–351, 2012.
- [35] K. Soufi, “Gestion De La Propagation Des Modifications À Travers Des Documents Méthodes,” vol. 2006, 2006.
- [36] F. Giguère, L. Rivest, and A. Desrochers, “Improving Design Productivity and Product Data Consistency,” in *Feature Based Product Life-Cycle Modelling*, Boston, MA: Springer US, 2003, pp. 77–91.
- [37] J. Tan, “Self-organizing assembly modeling based on relational constraints,” *Chinese J. Mech. Eng. (English Ed.)*, vol. 13, no. 02, p. 145, 2000.
- [38] Thomas Guillaume Tremblay, “La Propagation du Changement dans Un Contexte PLM : Vers La Maîtrise du Rôle des Associations dans Le Développement de Produits Et des Processus Associés [Maîtrise En Génie De La Production Auto Ma Tisée],” École de Technologie Supérieure Université Du Québec, 2006.
- [39] O. W. Salomons, F. van Slooten, G. W. F. de Koning, F. J. A. M. van Houten, and H. J. J. Kals, “Conceptual Graphs in CAD,” *CIRP Ann.*, vol. 43, no. 1, pp. 125–128, Jan. 1994.
- [40] O. Yildiz, “Atelier de conception pour l ’ évolution des systèmes PLM : une approche d ’ ingénierie dirigée par les modèles,” *Univ. Grenoble*, no. Thèse soutenue publiquement le 21 septembre 2015, 2016.
- [41] T. G. Tremblay, L. Rivest, O. Msaaf, and R. Maranzana, “The role of associations in CAD and PLM for handling change propagation during product development.”
- [42] A. Rogiez, “Le processus de conception intégré : principes et accompagnement vers le succès - Strategia Conseil.” [Online]. Available: <https://www.strategiaconseil.ca/2018/07/31/le-processus-de-conception-integre-principes-et-accompagnement-vers-le-succes/>. [Accessed: 16-Sep-2020].
- [43] D. Brissaud and S. Tichkiewitch, “Innovation and manufacturability analysis in an integrated design context,” *Comput. Ind.*, vol. 43, no. 2, pp. 111–121, Oct. 2000.

- [44] M. Z. Ouertani and L. Gzara, "Tracking product specification dependencies in collaborative design for conflict management," *CAD Comput. Aided Des.*, vol. 40, no. 7, pp. 828–837, 2008.
- [45] V. Kocar, "Modeling engineering change management process in virtual collaborative design environments."
- [46] K. R. Reddi and Y. B. Moon, "System dynamics modeling of engineering change management in a collaborative environment."
- [47] F. Demoly, X. T. Yan, B. Eynard, S. Gomes, and D. Kiritsis, "Integrated product relationships management: A model to enable concurrent product design and assembly sequence planning," *J. Eng. Des.*, vol. 23, no. 7, pp. 544–561, 2012.
- [48] P. G. M. Hugh D. Bradley, "A relation-based product model for computer-supported early design assessment," *J. Mater. Process. Technol.*, vol. 76, pp. 88–95, 1998.
- [49] J. Brown, "Managing Product Relationships: Enabling Iteration and Innovation in Design," 2006.
- [50] F. Demoly, L. Toussaint, B. Eynard, D. Kiritsis, and S. Gomes, "Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning," *CAD Comput. Aided Des.*, vol. 43, no. 12, pp. 1654–1673, Dec. 2011.
- [51] B. He, P. Zhang, N. Zhu, J. Cao, S. Huang, and W. Tang, "Skeleton model-based approach to integrated engineering design and analysis," *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 5–8, pp. 1105–1115, 2016.
- [52] T. A. W. Jarratt, "A Model-Based Approach To Support The Management Of Engineering Change," Cambridge University, 2004.
- [53] K. Rouibah and K. R. Caskey, "Change management in concurrent engineering from a parameter perspective," *Comput. Ind.*, vol. 50, no. 1, pp. 15–34, 2003.
- [54] Michel Maurino, *La Gestion Des Donnees Techniques. Technologie du concurrent engineering*. 1995.
- [55] W. F. Rowell, A. H. B. Duffy, I. M. Boyle, and N. Masson, "The nature of engineering change in a complex product development cycle," *Syst. Eng.*, vol. 2009, no. April, pp. 1–8, 2009.
- [56] M. Ouertani, L. Gzara-yesilbas, and L. Lossent, "Engineering Change Process : State of the Art , a Case Study and Proposition of an Impact Analysis Method," in *5th International Conference on IDMME*, pp. 1–12.
- [57] A. Kusiak and J. Wang, "Dependency analysis in constraint negotiation," *IEEE Trans. Syst. Man. Cybern.*, vol. 25, no. 9, pp. 1301–1313, 1995.
- [58] D. K. H. Chua and M. A. Hossain, "Predicting change propagation and impact on design schedule due to external changes," *IEEE Trans. Eng. Manag.*, vol. 59, no. 3, pp. 483–493, 2012.

- [59] T. Cohen, S. B. Navathe, and R. E. Fulton, "C-FAR, change favorable representation," *CAD Comput. Aided Des.*, vol. 32, no. 5, pp. 321–338, 2000.
- [60] H. Cheng and X. Chu, "A network-based assessment approach for change impacts on complex product," *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1419–1431, 2012.
- [61] A. Kusiak and J. Wang, "Dependency analysis in constraint negotiation," *IEEE Trans. Syst. Man. Cybern.*, vol. 25, no. 9, pp. 1301–1313, 1995.
- [62] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting Change Propagation in Complex Design," *J. Mech. Des.*, vol. 126, no. 5, p. 788, 2004.
- [63] B. Hamraz, N. H. M. Caldwell, and P. J. Clarkson, "A Matrix-Calculation-Based Algorithm for Numerical Change Propagation Analysis," vol. 60, no. 1, pp. 186–198, 2013.
- [64] R. Keller, T. Eger, C. Eckert, and P. J. Clarkson, "Visualising change propagation," *Int. Conf. Eng. Des.*, pp. 1–12, 2005.
- [65] R. Rutka, Guenov, Lemmens, Schmidt-Schäffer, Coleman, "Methods For Engineering Change Propagation Analysis," in *25TH International Congress Of The Aeronautical Sciences*, p. 8.
- [66] K. Reddi and Y. Moon, "A framework for managing engineering change propagation," *Mech. Aerosp. Eng.*, vol. 6, no. 5, pp. 461–476, 2009.
- [67] A. Brière-coté, L. Rivest, and R. Maranzana, "3D CAD model comparison: An evaluation of model difference identification technologies," *Comput. Aided. Des. Appl.*, vol. 10, no. 2, pp. 173–195, 2013.
- [68] G. A. Ollinger and T. F. Stahovich, "Redesignit - A Constraint-Based Tool For Managing," in *Proceedings of DETC '01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 1–11.
- [69] J. Clarkson, C. Simons, and C. Eckert, "Predicting Change Propagation In Complex Design," in *Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.
- [70] T. Jarratt, C. Eckert, P. J. Clarkson, and L. Schwankl, "Product architecture and the propagation of engineering change," *Des. 2002 Proc. 7Th Int. Des. Conf.*, pp. 75 – 80, 2002.
- [71] O. O. . E. C. M. . C. P. J. Ariyo, "Hierarchical decompositions for complex product representation," *Proc. Int. Des. Conf. Des. 2008*, no. 2, pp. 1–8, 2008.
- [72] B. Hamraz, N. H. M. Caldwell, and P. John Clarkson, "A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design," *J. Mech. Des.*, vol. 134, no. 10, p. 100905, 2012.
- [73] R. Rutka, Guenov, Lemmens, Schmidt-Schäffer, Coleman, "Methods For Engineering Change Propagation Analysis," in *25TH International Congress Of The Aeronautical*

- Sciences*, 2006, p. 8.
- [74] Y.-M. Chen, W.-S. Shir, and C.-Y. Shen, “Distributed engineering change management for allied concurrent engineering,” *Int. J. Comput. Integr. Manuf.*, vol. 15, no. 2, pp. 127–151, Jan. 2002.
- [75] V. Kocar and A. Akgunduz, “ADVICE: A virtual environment for Engineering Change Management,” *Comput. Ind.*, vol. 61, no. 1, pp. 15–28, 2010.
- [76] S. Li and L. Chen, “Towards Rapid Redesign: Pattern-based Redesign Planning for Large-Scale and Complex Redesign Problems,” *J. Mech. Des.*, vol. 129, no. 2, p. 227, 2007.
- [77] L. Chen and S. Li, “Pattern-Based Reasoning In Critical Parameter Management (Cpm) For Rapid Redesign,” in *Engineering Technical Conferences & Computers and Information in Engineering Conference Proceedings of DETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 1–10.
- [78] L. Chen, A. Macwan, and S. Li, “Model-based Rapid Redesign Using Decomposition Patterns,” *J. Mech. Des.*, vol. 129, no. 3, p. 283, 2006.
- [79] S. Li and E. Rajinia, “Path-Based And Pattern-Based Approaches For Change Management,” *Inf. Syst.*, no. July, pp. 279–292, 2010.
- [80] S. Li and L. Chen, “Pattern-based reasoning for rapid redesign: A proactive approach,” *Res. Eng. Des.*, vol. 21, no. 1, pp. 25–42, 2010.
- [81] Y. Ma, G. Chen, and G. Thimm, “Change propagation algorithm in a unified feature modeling scheme,” *Comput. Ind.*, vol. 59, no. 2–3, pp. 110–118, 2008.
- [82] M. C. Pasqual and O. L. de Weck, “Multilayer Network Model For Analysis And Management Of Change Propagation,” *DS 68-10 Proc. 18th Int. Conf. Eng. Des. (ICED 11), Impacting Soc. through Eng. Des. Vol. 10 Des. Methods Tools pt. 2, Lyngby/Copenhagen, Denmark, 15.-19.08.2011*, pp. 126–138, 2011.
- [83] A. Albert and Z. Christian, “Interdisciplinary systems modelling using the contact and channel model for SysML,” *Int. Conf. Eng. Des.*, no. August, 2011.
- [84] P. Boersting, R. Keller, T. Alink, C. M. Eckert, A. Albers, and P. J. Clarkson, “Design Methods 309 International Design Conference-Design,” in *International Design Conference - Design 2008*, pp. 309–316.
- [85] K. Rahmani and V. Thomson, “Managing subsystem interfaces of complex products,” *Int. J. Prod. Lifecycle Manag.*, vol. 5, no. 1, p. 73, 2011.
- [86] K. Rahmani and V. Thomson, “Ontology based interface design and control methodology for collaborative product development,” *Comput. Des.*, vol. 44, pp. 432–444, 2012.
- [87] J. R. Hauser and D. Clausing, “The House of Quality,” *Harv. Bus. Rev.*, vol. Volume 66, pp. 63–73, 1988.

- [88] S. Helms, F. G. H. Behncke, L. Lindlöf, M. C. Wickel, and U. Lindemann, "Classification of Methods Me Thods for the Indication of Change Propagation – a Literature Review," *Proc. 13th Int. Des. Conf. Des. 2014*, pp. 211–220, 2014.
- [89] M. Smith, J. Eger, and T. J. Clarkson, "A functional analysis of change propagation," in *Proceedings of the 14th International Conference on Engineering Design (ICED'03)*, 2003, p. 19–21.
- [90] L. Yin, D. Tang, Y. Kang, and S. Leng, "Topology face-based change propagation analysis in aircraft-assembly tooling design," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 230, no. 1, pp. 120–135, 2016.
- [91] K. G. Lough, R. B. Stone, and I. Tumer, "The Risk in Early Design (RED) method: Likelihood and consequence formulations," in *Proceedings of the ASME Design Engineering Technical Conference*, 2006, vol. 2006.
- [92] F. Yang and G. J. Duan, "Developing a parameter linkage-based method for searching change propagation paths," *Res. Eng. Des.*, vol. 23, no. 4, pp. 353–372, 2012.
- [93] D. K. H. Chua and M. A. Hossain, "Predicting change propagation and impact on design schedule due to external changes," *IEEE Trans. Eng. Manag.*, vol. 59, no. 3, pp. 483–493, 2012.
- [94] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting change propagation in complex design," *J. Mech. Des. Trans. ASME*, vol. 126, no. 5, pp. 788–797, Sep. 2004.
- [95] A. Brière-Côté, L. Rivest, and R. Maranzana, "Comparing 3D CAD models: Uses, methods, tools and perspectives," *Comput. Aided. Des. Appl.*, vol. 9, no. 6, pp. 771–794, 2012.
- [96] T. Hong, K. Lee, S. Kim, C. Chu, and H. Lee, "Similarity Comparison of Mechanical Parts," *Comput. Aided. Des. Appl.*, vol. 2, no. 6, pp. 759–768, 2005.
- [97] S. Chowdhury and Z. Siddique, "Development and Implementation of Shape Comparison of 3D CAD Models," in *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 1–10.
- [98] R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei, "Shape-Similarity Search of Three-Dimensional Models using paramterized statistics," in *proceedings of the Pacific Graphics 2002*, pp. 1–10.
- [99] A. Elinson, D. S. Nau, and W. C. Regli, "Feature-based similarity assessment of solid models," *Proc. fourth ACM Symp. Solid Model. Appl. - SMA '97*, pp. 297–310, 1997.
- [100] L. Zehtaban, O. Elazhary, and D. Roller, "A framework for similarity recognition of CAD models," *J. Comput. Des. Eng.*, vol. 3, no. 3, pp. 274–285, 2016.
- [101] D. Bepalov, A. Shokoufandeh, W. C. Regli, and W. Sun, "Scale-Space Representation of 3D Models and Topological Matching," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*, 2003, pp. 208–215.

- [102] Y. Lu, K. Kaneko, and A. Makinouchi, "Using a Partial Geometric Feature for Similarity Search of 3D Objects," *Ipsj Digit. Cour. ER*, vol. 3, pp. 674–682, 2007.
- [103] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proceedings International Conference on Shape Modeling and Applications*, 2002, pp. 154–166.
- [104] C. C. Yiu Ip Daniel Lapadat Leonard Sieger William Regli, "Using Shape Distributions to Compare Solid Models \* General Terms," in *Proceedings of the seventh ACM symposium on Solid modeling and applications*.
- [105] C. H. Chu and Y. C. Hsu, "Similarity assessment of 3D mechanical components for design reuse," *Robot. Comput. Integr. Manuf.*, vol. 22, no. 4, pp. 332–341, 2006.
- [106] Z. Siddique, K. Viswanathan, and S. Chowdhury, "Shape Comparison Of 3d Models Based On Features And Parameters Parameters," in *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference Information in En*, 2008, pp. 1–11.
- [107] X. Zhang, "Contribution to engineering change management in product development projects : reference models and multi-agent-based simulation," L'Université Bordeaux 1, 2013.
- [108] L. Yin, D. Tang, I. Ullah, Q. Wang, H. Zhang, and H. Zhu, "Analyzing engineering change of aircraft assembly tooling considering both duration and resource consumption," *Adv. Eng. Informatics*, vol. 33, pp. 44–59, 2017.
- [109] Y. G. Ma A Britton S B Tor L Y Jin, "Associative assembly design features: concept, implementation and application," *Int J Adv Manuf Technol*, vol. 32, pp. 434–444, 2007.
- [110] T. G. Tremblay, L. Rivest, O. Msaaf, and R. Maranzana, "The role of associations in CAD and PLM for handling change propagation during product development," in *13th ISPE International Conference On Concurrent Engineering: Research And Applications*, pp. 507–514.
- [111] B. Louhichi and L. Rivest, "Management of associations within digital mock-ups for improved collaboration," *IFIP Adv. Inf. Commun. Technol.*, vol. 388 AICT, pp. 411–423, 2012.
- [112] Y. M. Chen, W. S. Shir, and C. Y. Shen, "Distributed engineering change management for allied concurrent engineering," *Int. J. Comput. Integr. Manuf.*, vol. 15, no. 2, pp. 127–151, 2002.
- [113] A. T. Mathew and C. S. P. Rao, "A Novel Method of Using API to Generate Liaison Relationships from an Assembly," *Softw. Eng. Appl.*, vol. 2010, no. February, pp. 167–175, 2010.
- [114] J.-C. Yu and Y.-M. Li, "Structure representation for concurrent analysis of product assembly and disassembly," *Expert Syst. Appl.*, vol. 31, no. 4, pp. 705–714, Nov. 2006.

- [115] B. Hamraz and P. J. Clarkson, "Industrial evaluation of FBS Linkage – a method to support engineering change management," *J. Eng. Des.*, vol. 26, no. 1–3, pp. 24–47, 2015.
- [116] J. Clarkson, C. Simons, and C. Eckert, "Change Prediction for Product Redesign," *Int. Conf. Eng. Des. ICED 01*, vol. August, no. February 2014, 2001.
- [117] J. De Kleer and J. S. Brown, "A qualitative physics based on confluences," *Artif. Intell.*, vol. 24, no. 1–3, pp. 7–83, 1984.
- [118] Y. Fu, M. Li, and F. Chen, "Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix," *Int. J. Proj. Manag.*, vol. 30, pp. 363–373, 2012.
- [119] S. A. Bohner, "Impact analysis in the software change process: a year 2000 perspective," in *Conference on Software Maintenance*, 1996, pp. 42–51.
- [120] "2018 Aide de SOLIDWORKS - Enregistrer et lire les macros." [Online]. Available: [http://help.solidworks.com/2018/french/SolidWorks/sldworks/c\\_recording\\_playing\\_macros.htm](http://help.solidworks.com/2018/french/SolidWorks/sldworks/c_recording_playing_macros.htm). [Accessed: 07-May-2019].



# Annexes

## Traverse Assembly at Component and Feature Levels Using Recursion Example (C#)

```
using SolidWorks.Interop.sldworks;
using SolidWorks.Interop.swconst;
using System;
using System.Diagnostics;
namespace RecursiveTraverseAssemblyCSharp.csproj
{
    public partial class SolidWorksMacro
    {
        public void TraverseFeatureFeatures(Feature swFeat, long nLevel)
        {
            Feature swSubFeat;
            Feature swSubSubFeat;
            Feature swSubSubSubFeat;
            string sPadStr = " ";
            long i = 0;

            for (i = 0; i <= nLevel; i++)
            {
                sPadStr = sPadStr + " ";
            }
            while ((swFeat != null))
            {
                Debug.Print(sPadStr + swFeat.Name + " [" + swFeat.GetTypeName2() + "]");
                swSubFeat = (Feature)swFeat.GetFirstSubFeature();

                while ((swSubFeat != null))
                {
                    Debug.Print(sPadStr + " " + swSubFeat.Name + " [" + swSubFeat.GetTypeName() + "]");
                    swSubSubFeat = (Feature)swSubFeat.GetFirstSubFeature();

                    while ((swSubSubFeat != null))
                    {
                        Debug.Print(sPadStr + " " + swSubSubFeat.Name + " [" + swSubSubFeat.GetTypeName() + "]");
                        swSubSubSubFeat = (Feature)swSubSubFeat.GetFirstSubFeature();

                        while ((swSubSubSubFeat != null))
                        {
                            Debug.Print(sPadStr + " " + swSubSubSubFeat.Name + " [" + swSubSubSubFeat.GetTypeName() + "]");
                            swSubSubSubSubFeat = (Feature)swSubSubSubFeat.GetNextSubFeature();
                        }

                        swSubSubSubFeat = (Feature)swSubSubSubFeat.GetNextSubFeature();
                    }

                    swSubSubFeat = (Feature)swSubSubFeat.GetNextSubFeature();
                }

                swSubFeat = (Feature)swSubFeat.GetNextSubFeature();
            }

            swFeat = (Feature)swFeat.GetNextFeature();
        }
    }
}
```

```
public void TraverseComponentFeatures(Component2 swComp, long nLevel)
{
    Feature swFeat;

    swFeat = (Feature)swComp.FirstFeature();
    TraverseFeatureFeatures(swFeat, nLevel);
}
public void TraverseComponent(Component2 swComp, long nLevel)
{
    object[] vChildComp;
    Component2 swChildComp;
    string sPadStr = " ";
    long i = 0;

    for (i = 0; i <= nLevel - 1; i++)
    {
        sPadStr = sPadStr + " ";
    }

    vChildComp = (object[])swComp.GetChildren();
    for (i = 0; i < vChildComp.Length; i++)
    {
        swChildComp = (Component2)vChildComp[i];
        Debug.Print(sPadStr + "+" + swChildComp.Name2 + " <" + swChildComp.ReferencedConfiguration + ">");

        TraverseComponentFeatures(swChildComp, nLevel);
        TraverseComponent(swChildComp, nLevel + 1);
    }
}
public void TraverseModelFeatures(ModelDoc2 swModel, long nLevel)
{
    Feature swFeat;

    swFeat = (Feature)swModel.FirstFeature();
    TraverseFeatureFeatures(swFeat, nLevel);
}
... ..
```

```
public void Main()
{
    ModelDoc2 swModel;
    ConfigurationManager swConfMgr;
    Configuration swConf;
    Component2 swRootComp;

    swModel = (ModelDoc2)swApp.ActiveDoc;
    swConfMgr = (ConfigurationManager)swModel.ConfigurationManager;
    swConf = (Configuration)swConfMgr.ActiveConfiguration;
    swRootComp = (Component2)swConf.GetRootComponent();

    System.Diagnostics.Stopwatch myStopwatch = new Stopwatch();
    myStopwatch.Start();

    Debug.Print("File = " + swModel.GetPathName());

    TraverseModelFeatures(swModel, 1);

    if (swModel.GetType() == (int)swDocumentTypes_e.swDocASSEMBLY)
    {
        TraverseComponent(swRootComp, 1);
    }

    myStopwatch.Stop();
    TimeSpan myTimespan = myStopwatch.Elapsed;
    Debug.Print("Time = " + myTimespan.TotalSeconds + " sec");
}

/// <summary>
/// The SldWorks swApp variable is pre-assigned for you.
/// </summary>
public SldWorks swApp;
}
}
```

## Get Data for Fillet Feature Example (C#)

---

This example shows how to get the data for the selected fillet feature.

```
//-----  
// Preconditions:  
// 1. Open a model document with a simple fillet feature.  
// 2. Select the simple fillet feature.  
// 3. Open the Immediate window.  
// 4. Run the macro.  
//  
// Postconditions: Inspect the Immediate window.  
//-----  
  
using Microsoft.VisualBasic;  
using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Data;  
using System.Diagnostics;  
using SolidWorks.Interop.sldworks;  
using SolidWorks.Interop.swconst;  
using System.Runtime.InteropServices;  
  
namespace SimpleFilletFeatureData_CSharp.csproj  
{  
    partial class SolidWorksMacro  
    {  
  
        public void Main()  
        {  
            ModelDoc2 swModel = default(ModelDoc2);  
            SelectionMgr swSelMgr = default(SelectionMgr);  
            Feature swFeat = default(Feature);  
            SimpleFilletFeatureData2 swFillet = default(SimpleFilletFeatureData2);  
        }  
    }  
}
```

```
swModel = (ModelDoc2)swApp.ActiveDoc;
swSelMgr = (SelectionMgr)swModel.SelectionManager;
swFeat = (Feature)swSelMgr.GetSelectedObject(1, -1);
swFillet = (SimpleFilletFeatureData2)swFeat.GetDefinition();

Debug.Print("File = " + swModel.GetPathName());
Debug.Print("Feature = " + swFeat.Name);
Debug.Print(" Constant width? " + swFillet.ConstantWidth);
Debug.Print(" Curvature continuous? " + swFillet.CurvatureContinuous);
Debug.Print(" Default radius = " + swFillet.DefaultRadius);
Debug.Print(" Number of fillet items = " + swFillet.FilletItemsCount);
Debug.Print(" Multiple radii? " + swFillet.IsMultipleRadius);
Debug.Print(" Keep existing features? " + swFillet.KeepFeatures);
Debug.Print(" Apply fillet to attachment edges? " + swFillet.OmitAttachedEdges);
Debug.Print(" Overflow type = " + swFillet.OverflowType);
Debug.Print(" Extend fillet to all affected parts? " + swFillet.PropagateFeatureToParts);
Debug.Print(" Extend fillet to all tangent faces? " + swFillet.PropagateToTangentFaces);
Debug.Print(" Reverse normal? " + swFillet.get ReverseFaceNormal(0));
Debug.Print(" Round fillet corners? " + swFillet.RoundCorners);
Debug.Print(" Trim and attach to surfaces? " + swFillet.TrimAndAttachSurfaces);
Debug.Print(" Type of fillet? " + swFillet.Type);
Debug.Print(" Number of faces associated with this fillet = " + swFillet.GetFaceCount((int)swSimpleFilletWhichFaces_e.swSimpleFilletSingleRadius));

}

public StdWorks swApp;
}

}
```

# Ameni ELTAIEF

Doctorat : Matériaux, Mécanique, Optique, Nanotechnologie

Année 2020

## Modèle de réconciliation des associations et propagation des modifications au sein d'une maquette numérique

Un système mécanique est un ensemble de pièces interconnectées. Ainsi, une modification de l'une des pièces de l'assemblage entraîne des modifications affectant également les autres pièces. Cela peut entraîner des retards importants, voire des erreurs, ce qui augmente les coûts de conception/ fabrication du produit. Le développement d'un outil de propagation automatique des modifications permettrait donc de réduire les coûts et les délais qui peuvent survenir dans un modèle de propagation manuelle. L'objectif principal de cette thèse est d'améliorer et de faciliter la gestion des modifications pendant la phase de conception/ré-conception. En particulier, on s'est positionné à l'intérieur des systèmes CAO utilisés par le designer pour concevoir le modèle du produit. On part d'un modèle CAO existant (contenant tous les paramètres et éléments) et on extrait tous les données (en particulier les données de contraintes d'assemblage). Toutes ces données sont alors utilisées pour établir des associations entre les différentes entités géométriques. Ensuite, si une pièce de l'assemblage est modifiée, l'approche proposée permet de capturer les entités géométriques qui sont susceptibles d'être affectées et de vérifier l'impact prévisible de cette modification ainsi que l'arbre de propagation de celle-ci vers le reste de l'assemblage. Finalement, l'approche présentée est mise en œuvre à l'aide de Visual Studio et cela donne un outil de gestion et de propagation des modifications intégré dans SolidWorks. Différentes cas d'études sont utilisés pour tester la validité de notre outil.

Mots clés : conception assistée par ordinateur – modèles entité-association – graphes de lien – évaluation du risque.

## Association Reconciliation and Change Propagation Model within a Digital Mock-up

A mechanical product is essentially an assembly of interconnected parts. As a result, a modification that affects only one part of the assembly causes modifications affecting other parts as well. This can lead to significant delays or even errors, which increases the costs of the product's design and manufacture. The development of an automatic change propagation tool would therefore reduce the costs and delays associated with the risk of errors that may occur in a manually propagation model. The main objective of this thesis is to improve and facilitate the management of engineering changes during the design or re-design phase. In particular, we are interested in the CAD systems used by the designer to design the product model. We will start from an already existing CAD model (containing all the parameters and elements of a given product) and extract existing data (particularly assembly constraints data). Afterwards, all of this data is used to establish associations between the different geometrical entities. Then, if a part of the assembly is modified by one of the partners, our proposed approach captures the geometric entities that are likely to be affected and allows us to verify the expected impact of this modification as well as the propagation tree of the intended modification towards the whole assembly. Eventually, the presented approach is implemented using Visual Studio and the result is a simple change management and propagation tool that is integrated into SolidWorks. Different case studies are used to test the validity of our tool.

Keywords: computer-aided design – entity-relationship modeling – bond graphs – risk assessment.

Thèse réalisée en partenariat entre :

