



Faculté de génie  
Département de génie électrique et de génie informatique

## **Assistance à la prise de décision dans le processus de modification d'un produit en utilisant la technologie « Agent logiciel »**

**Thèse de doctorat présentée pour l'obtention  
du grade de Ph.D en génie électrique**

**Rédigé par : Dounia HABHOUBA**

### **Composition du jury**

**Dr. Soumaya Cherkaoui  
Dr. Alain Desrochers  
Dr. Jean-François Chatelain  
Dr. Ahmed Khoumsi  
Dr. Roland Maranzana**

---

**Dounia HABHOUBA**

**Sherbrooke (Québec), Canada**

**Août 2008**

IV-1883



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-42678-4*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-42678-4*

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux

**« Et dis: "Œuvrez, car Allah va voir votre œuvre, de même que Son messager et les croyants, et vous serez ramenés vers Celui qui connaît bien l'invisible et le visible. Alors Il vous informera de ce que vous faisiez" ».**

Sourate Le repentir (At-Tawba), verset [105]

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux

**« Que soit exalté Allah, le Vrai Souverain! Ne te hâte pas (de réciter) le Coran avant que ne te soit achevée sa révélation. Et dis: "Ô mon Seigneur, accroît mes connaissances!" »**

Sourate Tâ-Hâ, verset [114]

À mon très cher époux **Mehdi** et mes deux filles  
**Soukayna** et **Tasneem** qui sont ma joie de vivre

À mon père **Mohamed** que Dieu le tout puissant l'accueille  
dans son paradis

À ma mère **Khadija** que Dieu la protège

À mon beau père **Hadj Brahim** et ma belle mère **Malika**

À mon grand frère **Chaouki**

À mon adorable sœur **Siham** et mon beau frère **El Aid**

À mon frère **Hicham** et ma belle sœur **khadija**

À mon beau frère **Karim** et ma belle sœur **Batoul**

À mes très chers neveux; **Amine, Hajar, Anas** et **Selma**

À mes amies; **Nadia, Aya, Leila, Fatiha** et **Ghislaine**

## RESUMÉ

La gestion des changements d'ingénierie représente un défi pour les industriels dans le domaine de la conception mécanique assistée par ordinateur. Le système CAO est généralement entouré par d'autres systèmes tels que les SGDT les PGI ou autres. Pour que la gestion des changements soit entreprise efficacement, le système CAO doit être bien intégré avec les systèmes qui l'entourent. La communication entre les différentes équipes multidisciplinaires travaillant sur un projet a un impact positif sur le processus de gestion des changements dans sa globalité. Actuellement, la communication entre les disciplines se fait à l'aide d'échange de messages. Les experts sentent parfois le besoin de se rencontrer afin de se mettre d'accord sur le changement demandé. L'industrie de la CAO a donc besoin d'un système qui assiste l'humain dans la prise de décisions concernant les demandes de changements d'ingénierie. Un tel système diminuera considérablement le temps de traitement des demandes.

Ce projet de doctorat propose de réaliser un outil de collaboration qui assiste les concepteurs dans le processus de gestion des changements. Le système proposé assure une certaine cohérence à travers les différentes disciplines impliquées dans ce processus. C'est également un outil d'aide à la décision dans la mesure où il va proposer des solutions alternatives si jamais la demande de changement n'est pas validée. Nous avons également réalisé la propagation externe des modifications. Cette une étape qui était complètement intégrée à notre projet, car il fallait qu'on importe des données du système CAO pouvoir les traiter dans le système de gestion des changements. La technologie « Agent logiciel » est utilisée pour implémenter le système proposé. L'approche « Agent » présente certains avantages par rapport à l'approche classique Client/Serveur. Elle permet justement de minimiser le nombre de requêtes sur le réseau (avec la propriété de mobilité). Dans le système chaque agent représente une expertise donnée et il y a un Agent gestionnaire qui contrôle la communication entre les différentes disciplines.

**Mots clés :** Gestion des changements d'ingénierie, Conception mécanique assistée par ordinateur (CAO), ingénierie simultanée, processus de conception mécanique, Agents Logiciels.

## REMERCIEMENTS

Tout d'abord je remercie **DIEU** le tout puissant de m'avoir aidé à accomplir ce travail dans les meilleures conditions.

J'aimerais ensuite exprimer ma profonde gratitude :

Aux professeurs **Soumaya Cherkaoui** et **Alain Desrochers** pour leur patience, leur soutien et leurs encouragements. Leur encadrement et leurs conseils m'ont beaucoup aidé à bien réussir mes projets de maîtrise et de doctorat.

Au professeur **Jean-François Châtelain** d'avoir accepté d'être mon co-directeur dans mon projet de doctorat. Je le remercie pour ses conseils et ses suggestions.

À messieurs **Maxime Nicole** et **Jean-François Pépin** qui m'ont beaucoup aidé à bien assimiler certaines notions de la CAO.

Aux techniciens des départements de génie électrique et d'informatique et de génie mécanique d'avoir été très serviables.

À tous les membres du groupe de recherche **INTERLAB** pour avoir formé tous ensemble un groupe très dynamique et enrichissant.

Je remercie aussi très vivement mon très cher mari **Mehdi Haitami** pour sa patience et ses encouragements continus tout au long de mes années d'études.

Un gros merci à mes deux filles **Soukayna** et **Tasneem**, qui me donnent toujours le courage et l'énergie d'exceller dans tout ce que je fais.

Je remercie aussi toute ma famille au Maroc pour leur grand soutien moral même s'ils sont si loin de moi.

## TABLE DES MATIÈRES

<b>RÉSUMÉ.....</b>	<b>3</b>
<b>REMERCIEMENTS.....</b>	<b>4</b>
<b>LISTE DES FIGURES ET DES TABLEAUX.....</b>	<b>12</b>
<b>INTRODUCTION .....</b>	<b>15</b>
<b>I. PROBLÉMATIQUE COUVERTE PAR LA THÈSE .....</b>	<b>18</b>
I.1. MISE EN CONTEXTE : PRÉSENTATION DES DIFFÉRENTS OUTILS UTILISÉS DANS LE PROCESSUS DE CONCEPTION D'UN PRODUIT .....	18
<i>I.1.1. Systèmes CAO (Conception assistée par ordinateur) /FAO (Fabrication assistée par ordinateur)....</i>	<i>18</i>
<i>I.1.1.1. Systèmes CAO .....</i>	<i>18</i>
<i>I.1.1.2. Systèmes FAO.....</i>	<i>21</i>
<i>I.1.1.3. Intégration des systèmes CAO et FAO .....</i>	<i>21</i>
<i>I.1.2. Systèmes PGI (Progiciels de gestion intégrée).....</i>	<i>23</i>
<i>I.1.2.1. Les PGIs .....</i>	<i>23</i>
<i>I.1.2.2. Difficultés d'implémentation des PGIs.....</i>	<i>23</i>
<i>I.1.2.3. Les PGIs et les systèmes CAO .....</i>	<i>24</i>
<i>I.1.3. Logiciels de Systèmes de gestion de données techniques (SGDT).....</i>	<i>25</i>
<i>I.1.3.1. Les SGDTs.....</i>	<i>25</i>

1.1.3.1. <i>L'implantation du SGDT dans l'organisation</i> .....	26
I.2. ÉCHANGE DE DONNÉES ENTRE DES SYSTÈMES HÉTÉROGÈNES .....	27
1.2.1. <i>Échange de données entre des systèmes CAO différents</i> .....	27
1.2.2. <i>Échange de données entre des systèmes SGDT différents</i> .....	28
1.2.3. <i>Échange de données entre des systèmes PGI différents</i> .....	29
I.3. PRÉSENTATION GÉNÉRALE DES NORMES D'ÉCHANGE DE DONNÉES D'UN PRODUIT .....	29
1.3.1. <i>STEP</i> .....	29
1.3.2. <i>IGES</i> .....	30
1.3.3. <i>DXF</i> .....	31
1.3.4. <i>XML</i> .....	31
1.3.4. <i>Conclusion</i> .....	32
I.4. INGÉNIERIE SIMULTANÉE .....	32
I.5. PROBLÉMATIQUE : GESTION DU CHANGEMENT ET ASSISTANCE À LA PRISE DE DÉCISION DANS LE PROCESSUS DE CONCEPTION ET DE MODIFICATION D'UN PRODUIT .....	33
<b>II. ÉTAT DE L'ART</b> .....	<b>36</b>
II.1. INGÉNIERIE SIMULTANÉE ET ENVIRONNEMENT MULTIDISCIPLINAIRE .....	36
11.1.1. <i>L'ingénierie simultanée dans le domaine de la CAO/FAO</i> .....	36
11.1.2. <i>L'implémentation de l'ingénierie simultanée</i> .....	37
II.2. SYSTÈMES DE GESTION DE DONNÉES TECHNIQUES (SGDT) .....	40
11.2.1. <i>SGDT indépendants</i> .....	40
11.2.2. <i>SGDT liés aux systèmes CAO</i> .....	41



II.2.3. <i>SGDT intégrés dans des PGI</i> .....	42
II.2.3. <i>Comparaison entre les différents SGDT</i> .....	42
II.2.4. <i>Conclusion</i> .....	45
II.3. <b>MÉCANISMES DE GESTION DES CHANGEMENTS D'INGÉNIERIE DANS L'INDUSTRIE</b> .....	46
II.3.1. <i>Travaux déjà réalisés dans le domaine de la gestion du changement</i> .....	48
II.3.2 <i>Conclusion</i> .....	69
II.4. <b>L'INTÉGRATION DES AGENTS LOGICIELS AUX SYSTÈMES CAO POUR LA GESTION DU CHANGEMENT</b> .....	70
II.4.1. <i>PACT ou PACE (Palo Alto Collaborative Testbed/Environment)</i> .....	70
II.4.2. <i>DIDE (Distributed Intelligent Design Environment)</i> .....	71
II.4.3. <i>SiFA (Single Function agent)</i> .....	72
II.4.4. <i>SHARE</i> .....	74
II.4.5. <i>Processus distribué de planification et d'allocation de ressources basé sur les agents logiciels</i> .....	75
II.4.6. <i>Les contraintes de sécurité pour les dispositifs distribués sur le réseau</i> .....	76
II.4.7. <i>Des agents coopératifs en conception distribuée</i> .....	79
II.4.7. <i>Un système CAO collaboratif basé sur les agents logiciels</i> .....	81
II.4.8. <i>D'autres recherches</i> .....	82
II.4.6. <i>Position de notre travail de recherche</i> .....	84
<b>III. PROPOSITION D'UNE SOLUTION D'AIDE À LA GESTION DU CHANGEMENT À BASE D'AGENTS LOGICIELS : REVUE DE LITTÉRATURE</b> .....	<b>86</b>
III.1. <b>INTRODUCTION</b> .....	86

III.2. DÉFINITION DES CONCEPTS DE LA SOLUTION PROPOSÉE .....	86
<i>III.2.1. Environnement multi-vues et gestion du changement</i> .....	86
<i>III.2.2. Les Agents logiciels</i> .....	87
<i>III.2.2.1. Les agents intelligents</i> .....	87
<i>III.2.2.2. Les agents mobiles</i> .....	88
<i>III.2.3. Les techniques de négociations</i> .....	93
<i>III.2.3.1. Théorie de la vente aux enchères (VAE)</i> .....	93
<i>III.2.3.2. Contract-Net</i> .....	94
<i>III.2.3.3. Formation et restructuration de coalition</i> .....	96
<i>III.2.3.4. Théorie de jeux</i> .....	96
<i>III.2.3.5. Les systèmes à tableau noir</i> .....	97
<i>III.2.3.6. La négociation basée sur des plans</i> .....	98
<i>III.2.3.7. La négociation basée sur l'intelligence artificielle</i> .....	98
<i>III.2.3. Typage</i> .....	98
<b>IV. PROPOSITION D'UNE SOLUTION D'AIDE À LA GESTION DU CHANGEMENT À BASE D'AGENTS LOGICIELS : ÉTAPES D'IMPLÉMENTATION</b> .....	<b>99</b>
<i>IV.1 Définition des paramètres à changer dans le document de conception</i> .....	99
<i>IV.2 Architecture multi-agents proposée</i> .....	100
<i>IV.2.1. Architecture multi-agents</i> .....	100
<i>IV.2.2. Architecture détaillée du système</i> .....	103

IV.3. Algorithme de négociation.....	105
IV.4. L'originalité de l'architecture proposée.....	106
IV.5. EXEMPLE D'APPLICATION .....	107
IV.5.1. Algorithme de validation.....	108
IV.6. PERFORMANCE DU SYSTÈME PROPOSÉ.....	109
IV.7. ENVIRONNEMENT DE DÉVELOPPEMENT .....	111
IV.7.1. Langage Java.....	111
IV.7.2. Grasshopper.....	111
IV.7.3. Langage Visual Basic .....	112
IV.7.4. Système de CAO CATIA.....	112
IV.8. CONCLUSION.....	112
IV.9. INTÉGRATION DE LA SOLUTION PROPOSÉE AVEC LE SYSTÈME À ENTITÉS PARLANTES DÉVELOPPÉ À L'ETS .....	112
IV.9.1. Complémentarité.....	113
IV.9.2. Comparaison.....	114
<b>V. LE PROTOTYPE DE GESTION DES CHANGEMENTS PROPOSÉ : MAS-ECM (MULTI-AGENT SYSTEM-ENGINEERING CHANGE MANAGEMENT) .....</b>	<b>115</b>
V.1. INTRODUCTION.....	115
V.2. LES DIFFÉRENTS MODULES DU PROTOTYPE.....	116
V.2.1. Le module de gestion des contraintes.....	116
V.2.1.1. Définition des contraintes .....	116

<i>V.2.2. Le module de gestion des types de modèles</i> .....	117
<i>V.2.2.1. Arbre de typage</i> .....	117
<i>V.2.3. Le module de gestion des changements d'ingénierie</i> .....	118
<i>V.2.3.1. Gestion des vues (Views)</i> .....	119
<i>V.2.3.2. Gestion des pièces (Parts)</i> .....	120
<i>V.2.3.3. Gestion des poids des paramètres (Weights)</i> .....	121
<i>V.2.3.4. Gestion des utilisateurs (Users)</i> .....	122
<i>V.2.3.5. Gestion des matériaux (Materials)</i> .....	122
<i>V.2.3.6. Gestion des archives (Archives)</i> .....	123
<i>V.2.3.7. Gestion des changements (Change checking)</i> .....	123
<i>V.2.4. Le module de négociation</i> .....	124
<b>V.3. DÉFINITION DES CONTRAINTES DU PROTOTYPE</b> .....	125
<b>V.4. MODÈLE CAO UTILISÉ DANS LE PROTOTYPE</b> .....	128
<i>V.4.1. Modèle de conception : Aile d'un avion miniature</i> .....	128
<i>V.4.2. Modèle d'analyse de structure</i> .....	129
<i>V.4.3. Bibliothèque de matériaux utilisés</i> .....	129
<b>V.5. CRÉATION DU SYSTÈME D'AGENTS</b> .....	130
<b>VI. LES RÉSULTATS DE L'EXÉCUTION DU PROTOTYPE</b> .....	132
<b>VI.I. EXEMPLE D'EXÉCUTION DU PROTOTYPE</b> .....	132
<i>VI.I.I. État initial du modèle</i> .....	132

---

<i>VI.1.2. Travail des agents logiciels .....</i>	<i>134</i>
<i>VI.1.3. Conclusion .....</i>	<i>139</i>
VI.2. INTERPRÉTATION DU CONTENU DU TABLEAU NOIR .....	139
VI.3. NOTIFICATION DES UTILISATEURS .....	140
VI.4. PERFORMANCE DU PROTOTYPE .....	140
<b>CONCLUSION.....</b>	<b>142</b>
<b>BIBLIOGRAPHIES.....</b>	<b>145</b>
<b>ANNEXES .....</b>	<b>157</b>

## LISTE DES FIGURES ET DES TABLEAUX

FIGURE 1.1 : UTILISATION D'UN SYSTÈME CAO DANS L'INDUSTRIE [ZEID ET COLL., 1991].....	20
FIGURE 1.2 : PROCESSUS DE CONCEPTION ET DE FABRICATION [KUNWOO, 1999].....	22
FIGURE 1.3 : VUE FONCTIONNELLE D'UN SGGT [TANER, 1997] [CHEUNG ET COLL., 2003] .....	26
FIGURE 1.4 : UTILISATION DES TRADUCTEURS .....	28
FIGURE 1.5 : UTILISATION D'UNE NORME D'ÉCHANGE.....	28
FIGURE 1.6 : COMMUNICATION ENTRE LES DISCIPLINES INTERVENANTES DANS UN CHANGEMENT .....	34
FIGURE 1.7 : NÉGOCIATION D'UN CHANGEMENT D'INGÉNIERIE .....	34
FIGURE 2.1 : UNE ORGANISATION HIÉRARCHIQUE D'AGENTS .....	38
FIGURE 2.2 : LE MODÈLE DE RÉFÉRENCE DE LA GESTION DU CHANGEMENT DANS L'INGÉNIERIE [SHAH ET COLL., 1995].....	49
FIGURE 2.3: GESTION DU CHANGEMENT DANS LE SYSTÈME ACE [CHEN ET COLL., 2002] .....	51
FIGURE 2.4 : ARCHITECTURE GLOBALE DU SYSTÈME WEB-BASED ECM [HUANG ET COLL., 2001] .....	53
FIGURE 2.5: PLATEFORME DE COMMUNICATION SUPPORTANT UNE GESTION DE CHANGEMENT INTELLIGENTE.....	55
FIGURE 2.6 : LE MODÈLE INTERMÉDIAIRE (IM) DANS UNE REPRÉSENTATION PAR CARACTÉRISTIQUES.....	56
FIGURE 2.7 : LA REPRÉSENTATION D'UN MODÈLE SELON DIFFÉRENTS CONTEXTES [MARTINO ET COLL., 1998] .	58
FIGURE 2.8 : ARCHITECTURE DISTRIBUÉE DU SYSTÈME [MARTINO ET COLL., 1998] .....	58
FIGURE 2.9 : L'ARCHITECTURE DU MODÈLE D'INFORMATION PROPOSÉ : ARCHITECTURE DISTRIBUÉE DU SYSTÈME [MOKHTAR ET COLL., 1998].....	60

---

FIGURE 2.10 : PROPAGATION DU CHANGEMENT ENTRE COMPOSANTS.....	62
FIGURE 2.11 : GRAPHE HIÉRARCHIQUE DES CONTRAINTES .....	64
FIGURE 2.12 : ARCHITECTURE PACT [CUTKOSKY ET COLL., 1993] [SHEN, 2002].....	70
FIGURE 2.13 : ARCHITECTURE GÉNÉRALE DE DIDE [SHEN ET COLL., 1996] [SHEN ET COLL., 2000].....	72
FIGURE 2.14 : ARCHITECTURE DE SIFA [BROWN 1996].....	73
FIGURE 2.15: ARCHITECTURE SHARE .....	74
FIGURE 2.16 : PROCESSUS DE SÉCURITÉ TYPIQUE ENTRE LES DOMAINES; CLIENT, DÉPÔT ET LE DISPOSITIF .....	78
FIGURE 2.17: LA STRUCTURE AGENT DE PLACID .....	80
FIGURE 2.18: ARCHITECTURE DU CABOCAD [ROSENMAN ET COLL., 2001].....	83
FIGURE 3.1 : L'APPROCHE CLIENT/SERVEUR .....	89
FIGURE 3.2 : L'APPROCHE AGENT .....	89
FIGURE 3.3 : PRINCIPES DE LA MOBILITÉ .....	91
FIGURE 3.4 : MOBILITÉ FORTE .....	92
FIGURE 3.5 : ÉTAPES DU PROTOCOLE CONTRACT NET.....	95
FIGURE 4.1 : ARCHITECTURE PRÉLIMINAIRE DU SYSTÈME PROPOSÉ .....	100
FIGURE 4.2 : ARCHITECTURE DU SYSTÈME PROPOSÉ (A).....	101
FIGURE 4.2 : ARCHITECTURE DU SYSTÈME PROPOSÉ (B).....	104
FIGURE 4.3 : TABLEAU NOIR.....	105
FIGURE 4.4 : AILE D'UN AVION MINIATURE .....	107
FIGURE 4.5 : INTÉGRATION DU SYSTÈME PROPOSÉ AVEC LE SYSTÈME À ENTITÉS PARLANTES .....	114

---

FIGURE 5.1 : LES DIFFÉRENTS MODULES DU PROTOTYPE DÉVELOPPÉ .....	115
FIGURE 5.2 : MODULE DE CRÉATION DES CONTRAINTES .....	116
FIGURE 5.3 : ARBRE DE TYPAGE DES MODÈLES CAO .....	117
FIGURE 5.4 : MODULE DE GESTION DES CHANGEMENTS D'INGÉNIERIE .....	118
FIGURE 5.5 : MODULE DE GESTION DES VUES .....	119
FIGURE 5.6 : MODULE DE GESTION DES PIÈCES CAO.....	120
FIGURE 5.7 : MODULE DE GESTION DES POIDS DES PARAMÈTRES.....	121
FIGURE 5.8 : MODULE DE GESTION DES UTILISATEURS .....	122
FIGURE 5.9 : MODULE DE GESTION DES MATÉRIAUX .....	122
FIGURE 5.10 : MODULE DE GESTION DES ARCHIVES.....	123
FIGURE 5.11 : RÉSULTAT DE L'ANALYSE PAR ÉLÉMENTS FINIS .....	127
FIGURE 5.12 : GRAPHE DE LA DISTRIBUTION DES CONTRAINTES.....	127
FIGURE 5.13 : LE MODÈLE CAO UTILISÉ DANS LE PROTOTYPE .....	128
FIGURE 5.14 : L'ANALYSE PAR ÉLÉMENTS FINIS DU MODÈLE CAO UTILISÉ.....	129
FIGURE 5.15 : BIBLIOTHÈQUE DE GRADES D'ALUMINIUM .....	130
FIGURE 6.1 : LA PIÈCE « UPPER_SKIN_PANNEL » .....	132
FIGURE 6.2 : LA PIÈCE « LOWER_SKIN_PANNEL » .....	133
FIGURE 6.3 : LA PIÈCE « LEADING_SPAR ».....	133
FIGURE 6.4 : LA PIÈCE «TRAILING_SPAR » .....	134
FIGURE 6.5 : LE SYSTÈME D'AGENTS CRÉÉ APRÈS LE LANCEMENT DU PROTOTYPE .....	135



## Introduction

De nos jours, l'ingénierie des produits est une tâche de plus en plus réalisée de manière distribuée. Les travaux à différentes étapes du processus de réalisation de produits se font souvent dans différentes filiales et chez différents contractants. En effet, les compagnies sous-traitent aujourd'hui une partie importante des tâches dans des entreprises spécialisées afin de diminuer le coût final de production. Ces tâches peuvent appartenir à des domaines d'expertises différents : électriques, mécaniques, hydrauliques, etc., qui dépendent les uns des autres. Les concepteurs chez l'entreprise ou chez les contractants sont parfois amenés à effectuer des changements d'ingénierie assez importants sur les produits. Ces changements s'avèrent être de plus en plus faciles quand ils sont demandés dans les premières étapes de conception. Certes, faire des changements dans les dernières étapes peut affecter sérieusement le coût de production. Pour bien gérer un changement, il est primordial que les différentes disciplines intervenant dans le développement du produit se mettent d'accord sur le changement demandé pour éviter qu'il n'y ait des conflits et par conséquent des erreurs fatales dans le système de conception et de production. Il est donc important de faire collaborer toutes ces disciplines afin de se mettre d'accord sur une solution de changement. La gestion du changement dans les entreprises génère beaucoup de documents et nécessite beaucoup de signatures ce qui rend le processus de gestion du changement encore plus compliqué. Des outils de collaboration efficaces et intelligents sont donc de plus en plus nécessaires pour réaliser une coopération globale entre les différentes disciplines et par conséquent une gestion de changement plus efficace.

Le travail présenté s'inscrit dans le cadre de cette problématique. Il s'agit de la conception et de la réalisation d'un outil de collaboration intelligent pour la gestion du changement d'ingénierie et l'assistance à la prise de décision vis-à-vis des changements demandés dans le cadre de la conception assistée par ordinateur (CAO), dans un contexte d'ingénierie simultanée en utilisant les agents logiciels.

Dans le contexte de la conception mécanique assistée par ordinateur, il arrive souvent que des ingénieurs d'une discipline donnée demandant un changement d'ingénierie sur un produit soient confrontés à des problèmes générés dans d'autres disciplines à cause de ce changement. Les différents ingénieurs des différentes disciplines décident alors de collaborer pour trouver la solution adéquate. Cependant, cette collaboration demande un temps énorme qui pourrait être très coûteux en termes d'argent. Nous proposons donc d'automatiser partiellement ce processus de collaboration et faciliter ainsi la prise de décision pour les ingénieurs.

Pour implémenter le concept proposé, nous avons choisi la technologie « Agent », car elle constitue un outil efficace qui pourrait aider à intégrer différentes données hétérogènes. En plus, ils modélisent très bien la problématique étudiée.

Les objectifs principaux du projet dans sa globalité sont :

- Interpréter les changements demandés par les concepteurs.
- Proposer une architecture qui permet de gérer automatiquement les demandes de changements d'ingénierie et assister le concepteur à prendre des décisions quand les changements s'avèrent impossibles à effectuer.
- Maximiser la communication entre les différentes disciplines intervenant dans le développement d'un produit.
- Établir un lien entre le système de gestion des changements et le système CAO utilisé.
- Proposer un exemple d'application.
- Étudier les possibilités d'intégration avec le système à entités parlantes développé à l'ETS [SOUFI et coll., 2006].

Le document que nous présentons se décompose en six chapitres, le premier chapitre présente les différents systèmes existants dans une entreprise de conception mécanique, le lien qui existe entre eux et leur apport à la gestion des changements d'ingénierie. Dans le deuxième chapitre nous mettons l'accent sur les logiciels SGDT (Système de gestion de données techniques) en spécifiant leurs rôles dans le processus de gestion de changements. Nous présentons par la suite les travaux déjà fait dans ce domaine afin de situer la position de notre travail. La revue de littérature concernant la solution proposée est décrite dans le chapitre

trois. La présentation de l'architecture de notre système ainsi que ces principes de fonctionnement feront l'objet du chapitre quatre. Dans le chapitre cinq nous décrivons le prototype développé. Et enfin, nous présentons les résultats dans le chapitre six.

# CHAPITRE I

## I. Problématique couverte par la thèse

### I.1. Mise en contexte : Présentation des différents outils utilisés dans le processus de conception d'un produit

#### I.1.1. Systèmes CAO (Conception assistée par ordinateur) /FAO (Fabrication assistée par ordinateur)

##### I.1.1.1. Systèmes CAO

La CAO ou conception assistée par ordinateur vise à assister dans son travail tout concepteur dans plusieurs domaines (mécanique, électronique, informatique, etc.). Nous allons nous intéresser à la conception mécanique assistée par ordinateur. La CAO permet la modélisation géométrique 2D avec les fonctions de dessin et de schématique par exemple, et la modélisation 3D permet de traiter de la géométrie dans l'espace. Le taux d'utilisation des outils CAO dans les entreprises est important. Presque tous les domaines de l'industrie font appel aux logiciels CAO pour concevoir des produits à base de géométrie allant du simple au complexe et notamment dans le domaine de la mécanique.

La conception en ingénierie a bien évolué depuis l'ère de l'industrialisation jusqu'à l'ère de l'intégration. Au 18<sup>ième</sup> siècle, le développement des machines assistant ou remplaçant l'homme a vu le jour, surtout pour les tâches répétitives dans les chaînes de fabrication. À cette époque, le monde a connu les premières chaînes de montage automatisées (Ford). Au 20<sup>ième</sup> siècle, l'ordinateur assistant ou remplaçant l'homme dans les calculs les plus complexes est apparu. Selon la capacité des ordinateurs, les logiciels commençaient à apparaître sur le marché, les ingénieurs de l'époque étaient donc poussés à redéfinir des tâches suivant les

fonctions réalisées par ces logiciels. En 1963 Ivan Sutherland [SUTHERLAND 1963] a entamé les premiers travaux sur les systèmes de dessin assisté par ordinateur (DAO) qui ont évolué petit à petit vers la CAO et ce, en incluant des fonctions de calculs qui exploitent les maquettes virtuelles. Vers la fin du 20<sup>ième</sup> siècle et le début du 21<sup>ième</sup> siècle, l'Internet est apparu ce qui a permis le partage de l'information. Par conséquent, des standards universels d'échanges d'information ont commencé à se développer (normes STEP, IGES). Cette époque a été caractérisée aussi par l'explosion des fonctions dans les systèmes CAO.

Le marché des systèmes CAO évolue d'une manière continue et rapide, l'émergence des micro-ordinateurs et des stations de travail a fait en sorte que le prix des logiciels CAO diminue considérablement. Ceci dit, même les petites et moyennes entreprises peuvent se permettre aussi d'avoir un système CAO. La figure 1.1 [ZEID et coll., 1991] montre comment les systèmes CAO/FAO sont généralement utilisés dans l'industrie. Elle montre les composants de bases ainsi que les différentes fonctions que n'importe quel système de conception mécanique est supposé offrir. Les principaux modules présentés dans l'organigramme de la figure 1.1 sont le module géométrique et graphique, le module de conception, le module de fabrication et un module de programmation. Il y a plusieurs types de modèles disponibles pour la conception dans un système CAO. Dans chaque technique, une variété d'entités géométriques (module géométrique) peut être utilisée. Le module graphique contient les fonctions de transformation géométriques, les dessins de conception et la documentation, les couleurs et les couches (layers). Le module de conception inclut les applications de calcul de masse, d'analyse par éléments finis et de vérification des interférences. Le module de programmation est utilisé à des fins de personnalisation. Une fois que le document de conception est fini, les dessins et la documentation sont stockés dans une base de données. Le modèle est transmis par la suite aux applications FAO tel que le processus de planification, de vérification d'inspection et d'assemblage. Tous les logiciels CAO sont conformes à cette architecture. En fait, c'est l'interface utilisateur qui fait la différence entre un logiciel CAO et un autre. De ce fait, apprendre à manipuler un logiciel CAO permet d'apprendre facilement n'importe quel autre logiciel puisque les noyaux du système et les fonctions de bases changent très peu.

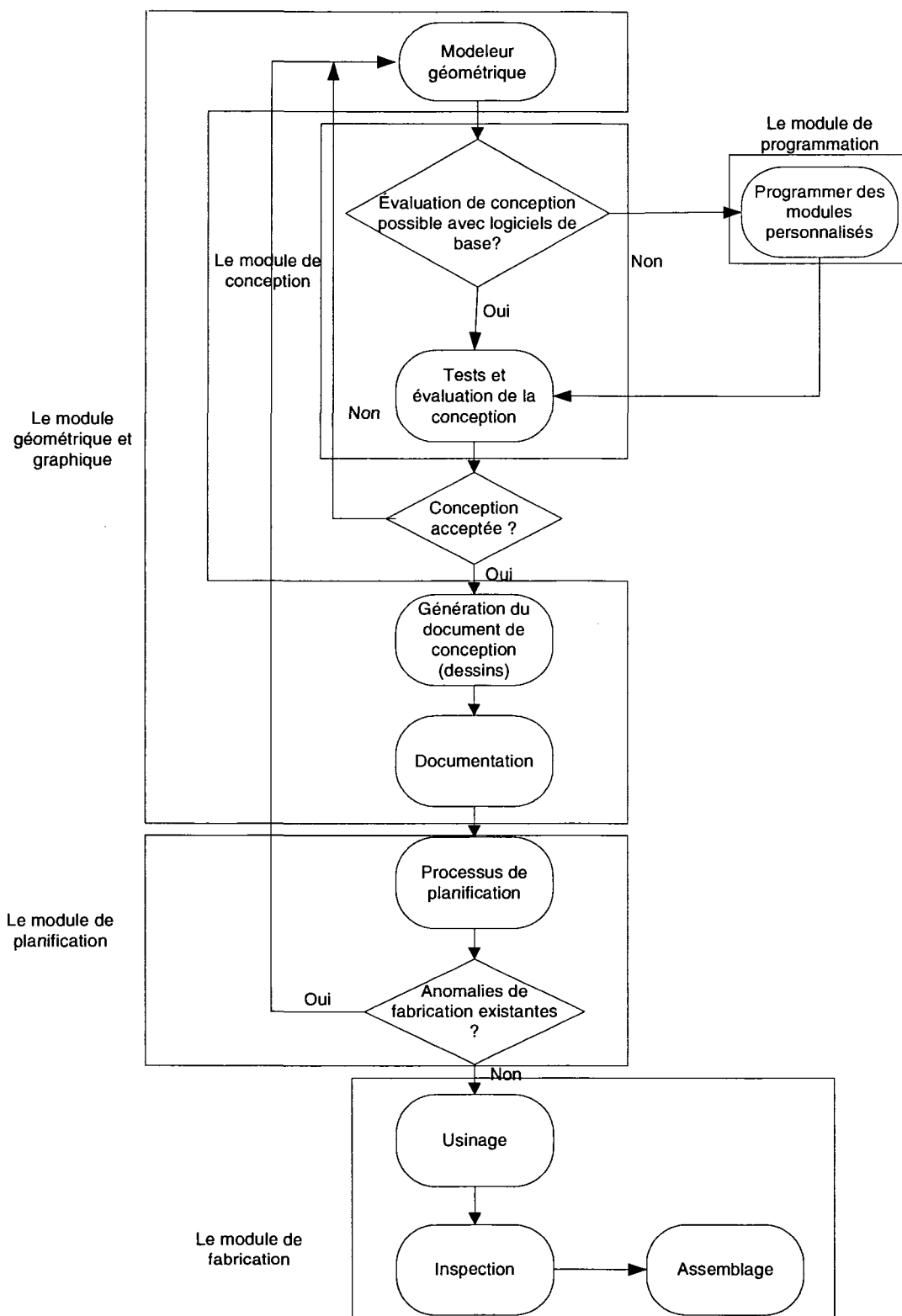


Figure 1.1 : Utilisation d'un système CAO dans l'industrie [ZEID et coll., 1991]

En conclusion, nous pouvons dire que les systèmes de CAO permettent d'assister les concepteurs dans la gestion de la complexité des produits dans un contexte où la pression du marché incite à produire plus rapidement et à moindre coût.

### **I.1.1.2. Systèmes FAO**

La FAO est une technologie qui a comme but de planifier, gérer et contrôler la fabrication. L'un des secteurs les plus matures de la FAO est le contrôle numérique (CN). La FAO utilise des techniques programmées pour contrôler une machine d'usinage qui coupe, poinçonne, plie ou tout simplement transforme la matière première en pièces finies. Pour usiner une pièce, l'ordinateur peut effectivement générer un nombre considérable d'instructions basées sur la géométrie de la pièce récupérée de la base de données CAO, en plus de quelques informations fournies par l'ingénieur de fabrication.

La FAO fournit également la possibilité de programmer des robots capables de prendre en charge quelques opérations d'usinage tel le soudage, l'assemblage ou encore le transport d'équipements.

### **I.1.1.3. Intégration des systèmes CAO et FAO**

Une fois que la conception d'une pièce est terminée dans le système CAO, le système FAO la prend en charge pour la fabriquer. Le système FAO contient plusieurs logiciels contribuant tous au processus de fabrication de la pièce. Premièrement, nous avons le PCAO (Processus de conception assistée par ordinateur) qui sert à planifier la phase de fabrication; ensuite il y a le logiciel CN qui sert à programmer les outils de fabrication afin de lancer la production, après il y a le logiciel d'inspection qui va inspecter la pièce dans la phase d'inspection; puis nous avons le logiciel pour programmer les robots pour qu'ils puissent s'occuper de l'opération d'assemblage d'une manière autonome.

Malheureusement, les systèmes CAO et FAO ne sont pas intégrés d'une manière efficace. Une intégration complète des deux systèmes augmenterait considérablement la production et permettra aux industriels de faire face à la concurrence ardue du marché. La première chose à laquelle il faut sérieusement réfléchir est l'automatisation du processus de planification

assisté par ordinateur (PPAO) qui se trouve être le pont entre la conception et la fabrication (figure 1.2).

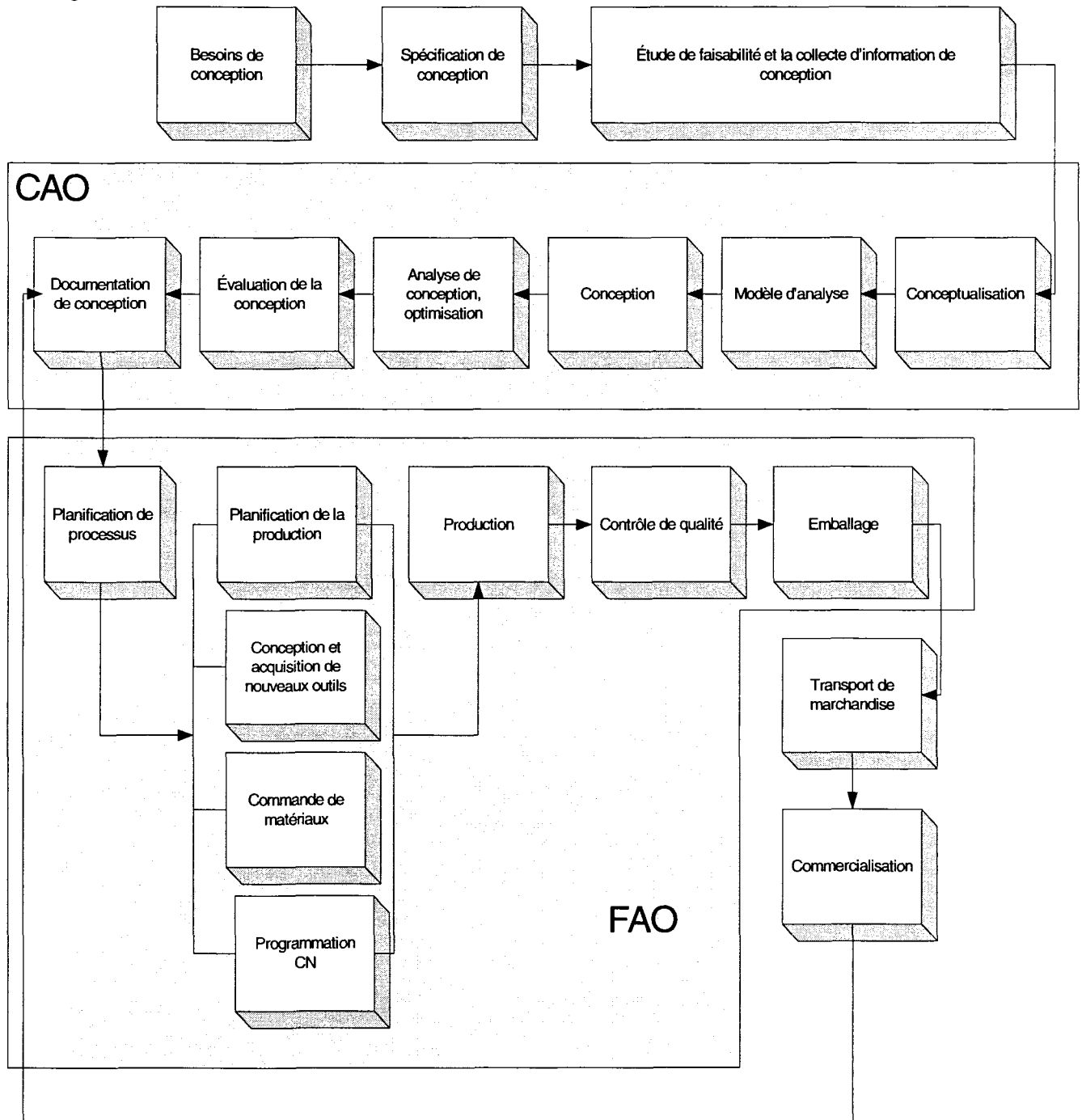


Figure 1.2 : Processus de conception et de fabrication [KUNWOO, 1999]

Les travaux de recherche qui se font actuellement sur l'interface entre la CAO et la FAO ont



pour but d'automatiser la communication entre les ingénieurs de conception et les ingénieurs de fabrication.

## **I.1.2. Systèmes PGI (Progiciels de gestion intégrée)**

### **I.1.2.1. Les PGIs**

Les PGIs sont des progiciels qui permettent de gérer et d'optimiser l'ensemble des ressources d'une entreprise, ils intègrent tous les systèmes déjà existants dans l'entreprise notamment la planification, la fabrication, les finances, le marketing etc. Les premiers progiciels sont apparus vers les années 70, ils ont offert une nouvelle vision de la gestion de l'information entre les différents départements d'une firme.

Ces progiciels présentent un genre d'épine dorsale pour l'organisation. Ils permettent en effet de faire circuler l'information entre les différents sous-systèmes de l'entreprise. Kalakota et Robinson [KALATOKA et coll., 1999] soumettent que la popularité des systèmes PGI provient du fait qu'ils ont résolu les défis posés par ce que l'on appelle « Les applications non connectées ou non branchées ». Hollande [GIBSON et coll., 1999] a affirmé que la raison majeure qui a poussé les industriels vers les PGIs c'est d'adapter les systèmes patrimoniaux aux systèmes contemporains. Les PGIs ont été également conçus pour des entreprises multi-sites, qui avaient besoin justement de contrôler leurs ressources et de s'adapter à des procédures de gestion et à des processus divers à travers l'organisation entière. Afin de réaliser l'intégration de toutes les unités d'une organisation, les systèmes PGI se fondent sur de grandes bases de données relationnelles centrales. Cette architecture représente un retour au modèle centralisé des années 60 et des années 70, où l'accès aux ressources informatiques et aux données était très centralisé. Par conséquent, l'utilisation de ces progiciels est une partie inhérente d'un phénomène général de centralisation dans les grandes entreprises.

### **I.1.2.2. Difficultés d'implémentation des PGIs**

Malgré leurs avantages, beaucoup de systèmes PGI ont échoué [STRATMAN et coll., 1999] (Stratman et Roth) à cause des difficultés d'exécution auxquelles ils font face en raison de la résistance des employés. Al-Mashari et Zairi [ZAIRI et coll., 1995] affirment que

l'exécution efficace des PGIs exige d'établir essentiellement des stratégies de gestion de changement et de favoriser l'infusion du système dans le milieu de travail. Avant l'implémentation d'un PGI il faut impérativement bien analyser tous les processus d'affaires existant dans l'entreprise, car si les processus sont mal compris, l'implémentation des PGIs ne servira qu'à faire les mauvaises choses assez rapidement.

### **I.1.2.3. Les PGIs et les systèmes CAO**

L'intégration entre les systèmes de CAO/FAO et les systèmes PGI peut être très réussie pourvu que l'échange des données entre les deux systèmes soit efficace. Afin de créer une communication entre les systèmes CAO et PGI, une organisation peut [SOLIMAN et coll., 2001] :

- Introduire le BOM (Bill of material) et toute autre information appropriée dans le système PGI pour effectuer la planification des besoins de fabrication et pour faire les commandes aux fournisseurs;
- Réaliser une véritable intégration entre les deux systèmes de sorte que l'information que le CAO/FAO produit soit automatiquement transmise au système PGI sans beaucoup d'effort.

La réalisation d'une véritable intégration peut rencontrer plusieurs obstacles entre autres :

- Le même type d'architecture de base de données est exigé dans toutes les plateformes existantes dans l'organisation.
- Le réseau existant doit supporter le grand échange des données entre les deux systèmes CAO et PGI.
- La cohérence des données doit être assurée entre les deux systèmes.

Ceci dit, plusieurs organisations ont effectivement commencé à améliorer cette intégration et ont pu par conséquent gagner en terme de coût de production et réduire considérablement le temps de mise en marché.

### **I.1.3. Logiciels de Systèmes de gestion de données techniques (SGDT)**

#### **I.1.3.1. Les SGDTs**

Les systèmes de gestion de données fournissent un moyen sécurisé pour modéliser, stocker, fédérer et gérer des données relatives aux produits et services offerts par une entreprise. Ils permettent à chaque acteur au sein d'une organisation distribuée de disposer d'une vue sur les informations relatives aux produits, constituants et assemblages, afin d'assurer un premier niveau de cohérence pour des produits complexes. Les organismes qui s'intéressent à l'ingénierie mécanique étaient les premiers à découvrir le besoin pour un système de gestion de données techniques (SGDT) mais, une fois développé ce type de logiciel a été utilisé en dehors de ces domaines d'application originaux.

Dans le domaine de la mécanique, les logiciels de SGDT fournissent des structures pour modéliser les relations, parfois très complexes, qui peuvent exister entre les différents constituants d'un assemblage, les personnes, les groupes de personnes, les schémas, ainsi que tous les documents liés par des liens technologiques. Les logiciels de SGDT permettent aussi de stocker les différents fichiers reliés au système CAO dans un ordinateur sécurisé, l'accès à ces données étant bien entendu contrôlé.

En outre, avec les systèmes SGDT, certaines tâches concourantes peuvent être facilement accomplies. L'ingénieur de fabrication est par exemple avisé du changement du concepteur et a accès au document de conception avant qu'il ne soit validé. Les travaux de fabrication sont étroitement liés aux travaux de conception; à chaque fois qu'il y a un changement dans la conception, le personnel de la fabrication est avisé, généralement par courriel, de la nouvelle version des documents reliés à la conception et peut donc faire les changements nécessaires.

Comme illustré sur la figure 1.3 [TANER, 1997] [CHEUNG et coll., 2003], quand les utilisateurs du système de CAO agissent sur des fichiers de données, la mise à jour des méta-données est faite automatiquement via le système SGDT (modification du nom du fichier par exemple). Ceci illustre la liaison étroite et l'intégration réussie des systèmes SGDT et des systèmes CAO.

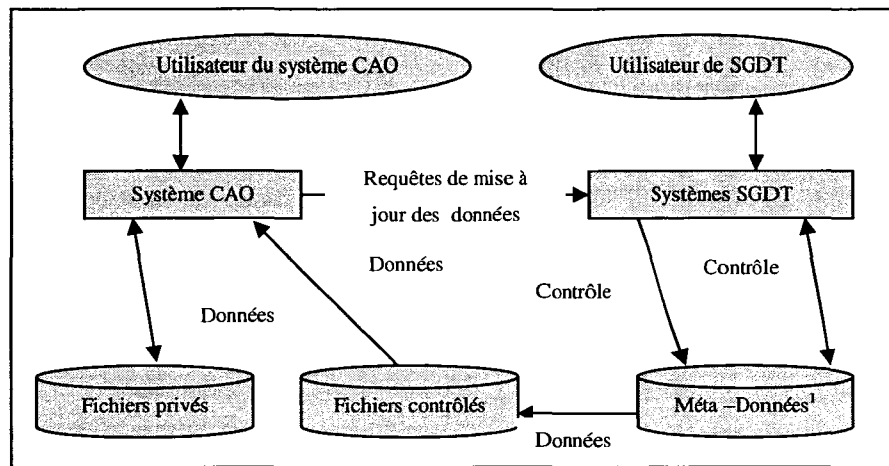


Figure 1.3 : Vue fonctionnelle d'un SGDT [TANER, 1997] [CHEUNG et coll., 2003]

Les systèmes SGDT peuvent être considérés comme des outils qui contribuent de très près à l'amélioration de la qualité et de la fiabilité d'un produit, accélèrent le temps de mise en marché, et participent à la réduction des coûts de développement du produit. Le problème avec ces outils c'est qu'ils ne prévoient pas de stratégie efficace pour la gestion des changements d'ingénierie.

### I.1.3.1. L'implantation du SGDT dans l'organisation

#### a) Intégration SGDT/CAO

Les systèmes SGDT sont bien intégrés aux systèmes CAO. L'avantage des systèmes SGDT c'est qu'ils sont facilement personnalisables en utilisant le langage VB. Les entreprises peuvent donc rajouter les fonctions et les attributs dont ils ont besoin assez facilement. Parfois les industriels ont besoin de fonctions pas nécessairement réalisables dans les systèmes SGDT, dans ce cas ils se dirigent vers d'autres logiciels externes.

#### b) Intégration SGDT/PGI

Les SGDTs de base sont axés essentiellement sur le stockage des données, la gestion des documents, le contrôle des révisions et des accès, la gestion des flux de données et la communication avec les autres systèmes au sein de l'organisation tels les PGIs [Product Lifecycle Management Information Center]. En fait, ces systèmes n'offrent aucune interface

directe avec le PGI ou le GRC (Gestion des relations consommateur) ou encore avec les systèmes de planification de la fabrication. Ce type de SGDT est plus approprié pour des petites et moyennes entreprises (PME) qui ont des chaînes d'approvisionnement assez simples.

D'un autre côté, il existe des SGDTs assez sophistiqués tels ENOVIA [ENOVIA] ou Métaphase [METAPHASE] qui assurent une communication assez développée avec le PGI d'une organisation. Le vrai problème qui existe est la définition et la standardisation des données échangées. Les SGDTs et les PGI ont besoin d'échanger leurs BOM. Ceci s'avère assez difficile puisque chacun des deux systèmes a sa propre définition du BOM puisqu'il a une vue différente sur le produit. Cependant, la norme AP 224 (STEP standard) [CHEUNG et coll., 2003] peut devenir le standard d'échange entre les deux systèmes. Cette norme réduira considérablement l'effort des fournisseurs de SGDT et PGI de partager l'information du BOM. Les données échangées vont être conforme au standard et donc compréhensible par les deux systèmes.

Et si nous nous posions la question suivante : Est-ce qu'on a réellement besoin d'un SGDT et d'un PGI dans une organisation? C'est la question que se posent la plupart des industriels et ils aimeraient bien que la réponse soit NON. Ceci signifierait qu'on voudrait avoir un système qui gère « le processus d'ingénierie »; de l'état « création » à l'état « libération » d'un produit incluant le contrôle des révisions et l'intégration avec le système CAO et gérer par la suite « le processus de fabrication »; planification de la production, coût, gestion des ventes, gestion des commandes, etc. Malheureusement, ceci est loin d'être réalisable, du moins pour l'instant puisque les développeurs des SGDTs ne connaissent pas vraiment les PGI et vice versa.

## **I.2. Échange de données entre des systèmes hétérogènes**

### **I.2.1. Échange de données entre des systèmes CAO différents**

Plusieurs types de systèmes CAO existent actuellement sur le marché, malheureusement ces systèmes ne gèrent pas l'information de la même manière et donc ne sont pas capables de s'échanger leurs données directement. Pour les industriels, l'échange des données techniques

est très important. Ceci leur permet d'échanger des documents avec des consommateurs, des fournisseurs ou des associés externes. Il existe deux solutions qui permettent de réaliser cet échange ; la première c'est le développement de logiciels qui permettent de relier deux systèmes spécifiques, c'est ce qu'on appelle les traducteurs directs (figure 1.4). Ceci dit, cette solution s'avère un peu coûteuse puisqu'il faudrait développer un traducteur pour chaque paire de logiciels CAO. La deuxième solution est l'utilisation des formats neutres d'échange de documents tel IGES, STEP, DXF (figure 1.5). Cette solution est la plus réaliste, elle exige le développement de deux traducteurs. Le premier traduit l'information à partir du système CAO au format STEP ou IGES, le deuxième convertit l'information à partir du fichier STEP ou IGES à un format compris par le système CAO.

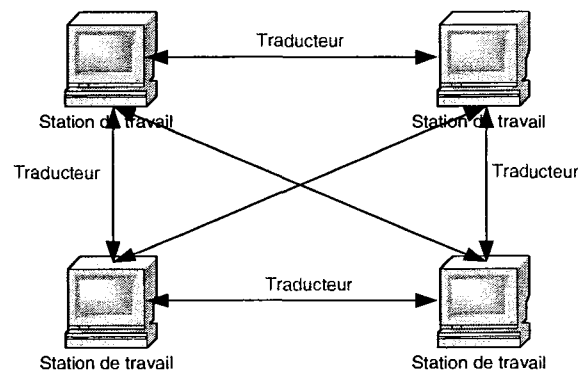


Figure 1.4 : Utilisation des traducteurs

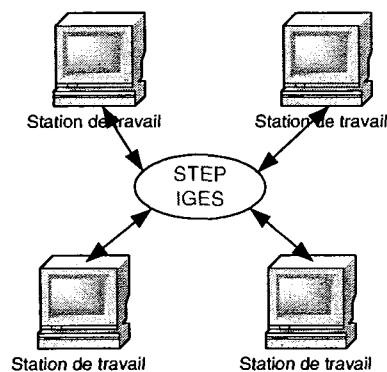


Figure 1.5 : Utilisation d'une norme d'échange

### I.2.2. Échange de données entre des systèmes SGDT différents

Les systèmes SGDT gèrent toute sorte d'information liée aux produits CAO. Les industriels

jugent qu'il est de plus en plus nécessaire de partager ces informations entre les différents intervenants dans le cycle de développement d'un produit, et ce, dans un environnement d'ingénierie simultanée. La plupart des fournisseurs de SGDTs proposent le format XML pour partager l'information entre différents systèmes SGDT. Ce format non standardisé nécessite tout de même une personnalisation pour être transmis et compris par les autres systèmes. L'OMG<sup>1</sup> ; le consortium de standardisation des systèmes orienté objet; propose donc d'utiliser un standard d'échange basé sur la norme STEP.

Le « schéma STEP PDM [SHEN-CHOU et coll., 2002] » contient la plupart des fonctionnalités pour des pièces et des documents existants dans un SGDT (identification, visionnement, structure, approbation, autorisation, classification, etc.). Le standard STEP PDM (AP232) offre aux industriels la possibilité de normaliser les fonctionnalités des SGDTs, en utilisant une nomenclature commune.

### **I.2.3. Échange de données entre des systèmes PGI différents**

Dans l'industrie, les compagnies ont besoin d'échanger des données avec des fournisseurs ou des sous-contractants. Les différents intervenants dans un échange de données peuvent ne pas avoir le même système PGI ou encore ne pas en avoir du tout (les petites compagnies). Les industriels ont donc opté pour XML pour normaliser l'échange de données entre eux.

## **I.3. Présentation générale des normes d'échange de données d'un produit**

### **I.3.1. STEP**

STEP est un standard international d'échange de données entre différents systèmes d'ingénierie. Le standard STEP est une norme de l'ISO (ISO 10303), qui a été développée pour traiter tous les produits manufacturés, tous les métiers et tous les stades du cycle de vie d'un produit et pour échanger des données en définissant un format neutre.

Le standard STEP se présente sous la forme d'un cadre de travail se composant lui-même de plusieurs parties :

1 : Object Management Group

Les méthodes de description qui fournissent des standards pour la définition conceptuelle des modèles de données à échanger.

- Les méthodes de mise en œuvre fournissant des standards pour la réalisation des échanges soit par fichier, soit par des accès directs à des bases de données.
- Les ressources intégrées : Ce sont des standards qui constituent les « briques » élémentaires de données de STEP. Ces briques élémentaires sont destinées à être réutilisées par les protocoles d'application.
- Les protocoles d'application : Ce sont des standards qui définissent un modèle d'échange pour un domaine d'activité donné (ex : AP209 pour les échanges de données de simulation, AP214 pour les échanges de données pour le domaine automobile, etc.)

Le langage utilisé pour la description de l'information sur les modèles est le langage EXPRESS (ISO 10303-11). EXPRESS permet de décrire les données d'un modèle sous forme d'entités. Il représente aussi les attributs et les relations entre les différentes entités.

L'inconvénient du standard STEP c'est qu'il extrait beaucoup d'information à partir des modèles CAO alors que les industriels ne veulent pas vraiment exposer trop leurs modèles à l'extérieur de l'entreprise.

### **I.3.2. IGES**

IGES est l'abréviation de « Initial Graphics Exchange Specification ». C'est l'une des plus anciennes normes d'échanges de données entre différents systèmes CAO. Elle est normalisée ANSI. Elle a été développée en 1980 à l'ICAM (US Air Force Integrated Computer Aided Manufacturing). Ce standard est encore très utilisé surtout dans l'industrie automobile et aérospatiale.

IGES présente beaucoup de limitations. Tout d'abord, il n'utilise pas de langage formel. Ensuite il ne distingue pas clairement entre les spécifications logiques du standard (l'interprétation des données du point de vue du système CAO), les exigences de certaines applications (comment utiliser les données extraites en respectant les exigences des



applications au sein du système CAO) et les spécifications physiques pour stocker les données dans le fichier d'échange. IGES ne contient pas de module de test pour vérifier la conformité des différents systèmes CAO au standard. En plus, IGES permet aux industriels de choisir la partie du standard qu'ils désirent implémenter, ce qui peut créer un problème de communication entre les processeurs IGES des systèmes CAO qui veulent faire un échange [MCMAHON et coll., 1998].

### **I.3.3. DXF**

Autodesk a développé sa propre norme d'échange de données ; DXF (Data eXchange Format) pour le logiciel AutoCad. Ce format d'échange de données est très simple, cependant il n'est utilisé qu'avec des systèmes CAO rudimentaires.

### **I.3.4. XML**

XML (eXtensible Markup Language) est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises.

Contrairement à HTML, qui est à considérer comme un langage défini et figé (avec un nombre de balises limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte.

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir.

Le format de données 3D XML est un format d'échange XML de données 3D des produits Dassault Systèmes, éditeur de logiciels CAO (CATIA SolidWorks) [3D XML].

### **I.3.4. Conclusion**

Contrairement à STEP, et en plus des inconvénients cités plus haut, IGES traduit seulement la géométrie et ne traite pas les solides ou encore les propriétés des surfaces. De ce fait, IGES n'est pas vraiment pratique pour beaucoup d'applications impliquant d'autres types de données que la géométrie. La norme DXF et d'autres normes propriétaires ne peuvent être utilisées qu'avec les systèmes CAO associés, qui sont en fait des systèmes assez rudimentaires. STEP reste donc la norme la plus complète qui satisfait le plus les besoins des industriels.

### **I.4. Ingénierie simultanée**

Confrontées quotidiennement à l'émergence de nouveaux marchés et à la croissante de la compétition, les entreprises soucieuses d'accroître leur productivité et leur compétitivité ne peuvent plus suivre le cycle classique séquentiel de développement d'un produit. Ces entreprises doivent produire à moindre coût et de plus en plus rapidement. C'est dans ce but que le concept d'ingénierie simultanée a été développé [PRASAD, 1996].

L'ingénierie simultanée est une approche qui intègre simultanément les différentes phases de développement d'un produit [CROSS, 2000], notamment l'identification des besoins du client, les spécifications, la conception et la fabrication du produit, incluant le service après-vente, l'entretien, et même le recyclage. En travaillant dans un environnement d'équipes multidisciplinaires, il est possible de développer rapidement des produits de qualité, à des coûts compétitifs.

Ceci dit, pour l'accélération du cycle de conception, l'amélioration du processus de production, l'augmentation de la qualité des produits et l'affrontement de la concurrence sur le marché, les entreprises devraient se soucier également de la gestion de la communication entre les différentes disciplines intervenant dans le processus de conception d'un produit dans un cadre d'ingénierie concurrente.

## **I.5. Problématique : Gestion du changement et assistance à la prise de décision dans le processus de conception et de modification d'un produit**

La gestion du changement dans le processus de conception a toujours posé un grand problème pour les industriels. D'après ce que nous avons vu plus haut, l'échange d'information entre les industriels et leurs sous-contractants (généralement de CAO à CAO ou de SGDT à SGDT ou d'un PGI à un PGI) peut se faire d'une manière plus au moins facile grâce à la norme STEP ou XML. Ce qui n'est pas encore évident, c'est l'échange d'information entre les outils intervenants dans le processus de conception d'un produit au sein de l'entreprise (CAO, PGI et SGDT), autrement dit, la propagation de l'information du système CAO vers les SGDTs, PGI ou autres systèmes externes. Ceci rend la gestion des modifications d'ingénierie entreprises par les ingénieurs de conception encore plus complexe.

Grâce à l'intégration des logiciels CAO et des logiciels SGDT, l'échange d'information entre ces deux systèmes est presque automatique. Dans le système SGDT on trouve l'information sur la création d'un document CAO et sur tous les documents qui y sont reliés. Le SGDT contrôle aussi l'accès aux documents et aux différentes révisions. Cependant, on ne trouve nulle part des informations sur la géométrie ou encore les caractéristiques des pièces contenues dans le document CAO.

Pour ce qui est de l'intégration entre les systèmes CAO, SGDT et PGI, certains développeurs ont réussi à transmettre des informations de fabrication des systèmes CAO et SGDT vers les systèmes PGI. Les informations concernant les dimensions et les caractéristiques d'un produit ne sont disponibles que dans le fichier CAO associé et ne sont pas accessibles ailleurs.

En plus de l'intégration des différents outils intervenant dans le cycle de vie d'un produit, les industriels se soucient également de la gestion de la communication entre les différentes équipes multidisciplinaires travaillant sur un produit, chose qui influence considérablement le processus de gestion du changement dans sa globalité. Chaque discipline devrait décider de l'approbation ou le rejet de n'importe quelle demande de changement demandée et notifier les autres disciplines concernées. Jusqu'à date, la communication entre les différentes disciplines se fait à l'aide d'échange de messages (figure 1.6). L'approbation d'un changement

demandé pour un produit ne peut ainsi être faite que si des intervenants humains représentant les différentes disciplines négocient et se mettent d'accord sur le changement ce qui peut prendre beaucoup de temps (figure 1.7). Les industriels ont donc grand besoin d'un système qui peut assister l'humain à prendre une décision vis-à-vis des changements demandés et qui permet par conséquent de diminuer le temps de traitement d'une demande de changement en vue d'accélérer la production.

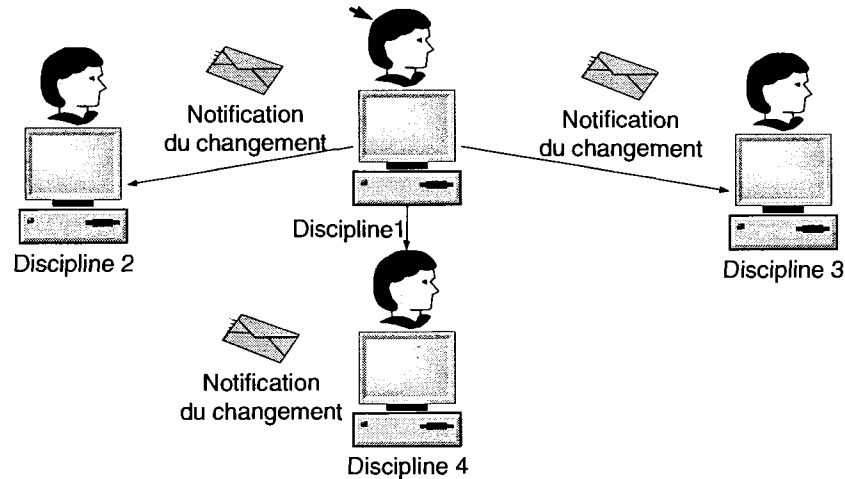


Figure 1.6 : Communication entre les disciplines intervenantes dans un changement

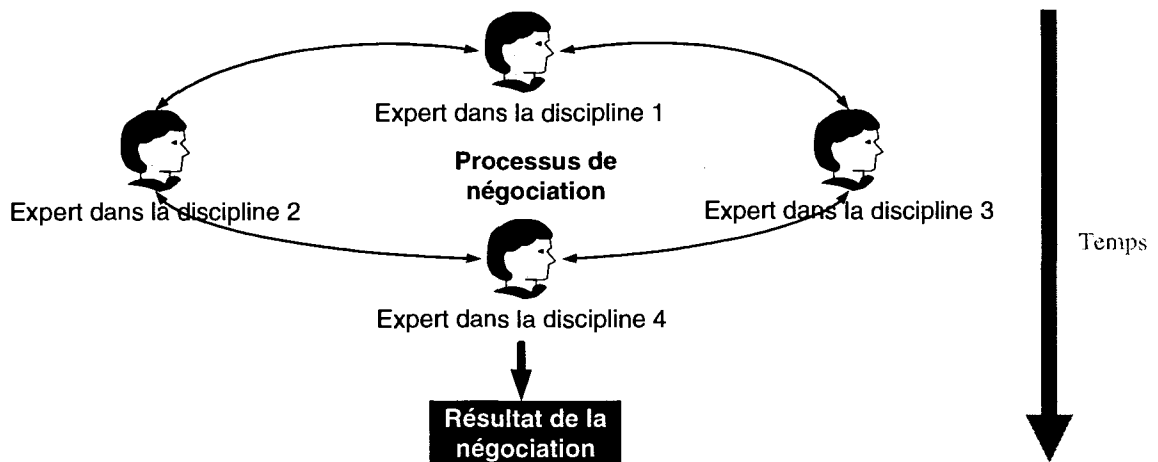


Figure 1.7 : Négociation d'un changement d'ingénierie

Le projet de doctorat propose de réaliser la propagation du changement effectué dans un

produit CAO vers les systèmes concernés par le changement, sans pour autant transmettre tout le fichier du produit ou encore sans envoyer des messages ou écrire des notes pour aviser les personnes concernées. Le projet de recherche CRIAQ s'intéresse à cette problématique. En effet, l'une des principales préoccupations du CRIAQ est la réalisation de la propagation interne d'un changement au sein d'un système CAO, ce qui a déjà été fait par des étudiants de l'ETS [SOUFI et coll., 2006], ainsi que la propagation externe d'un changement, autrement dit, dès qu'une modification est faite sur un modèle CAO, les nouvelles valeurs des paramètres modifiés sont transmises à d'autres systèmes se trouvant à l'extérieur de la sphère du système CAO utilisé pour un traitement quelconque. La propagation externe n'est qu'une partie du travail que nous proposons de réaliser dans notre projet de doctorat.

Le volet principal de ce travail est la proposition d'une solution pour la gestion du changement ou la gestion de la communication entre toutes les disciplines concernées par les changements d'ingénierie demandés. En effet, le système proposé permet de vérifier si le changement demandé pour un produit ne crée aucune incohérence avec les contraintes des différentes disciplines. Si jamais c'était le cas, notre système assiste les décideurs à prendre une décision vis-à-vis du changement demandé en leur proposant des alternatives. Cette partie de notre projet est en fait étroitement liée à la première partie puisque notre système devrait recevoir les changements effectués sur le modèle CAO (propagation externe) pour pouvoir les approuver ou les rejeter.

## CHAPITRE II

### II. État de l'art

#### II.1. Ingénierie simultanée et environnement multidisciplinaire

L'ingénierie concourante ou simultanée permet de franchir une nouvelle étape dans la gestion de projets de développement de systèmes complexes. L'ingénierie simultanée est fortement utilisée dans le domaine aérospatial, elle a permis à plusieurs équipes de développement impliquant des milliers d'ingénieurs représentant plusieurs disciplines et domaines différents (aérodynamique, propulsion, matériaux, structure, stabilité, contrôle, etc.) de mener à bien leurs tâches. L'ensemble des équipes travaille parfois sur les mêmes bases de données à partir de plusieurs terminaux répartis sur les différents continents. L'institut d'ingénierie simultanée (IIS) [ICARESPACE] a affirmé que les résultats obtenus par l'utilisation de l'ingénierie simultanée sont très satisfaisants. Toutefois, l'ingénierie concourante présente des limites très vite ressenties dans les gros projets. Il existe toujours quelques bases de données relatives à certaines disciplines qui restent isolées et non connectées à d'autres bases de données durant le processus de développement.

##### II.1.1. L'ingénierie simultanée dans le domaine de la CAO/FAO

Dans l'industrie de la CAO/FAO, le cycle de développement d'un produit est assez long, et les modifications à faire au cours du processus sont assez fréquentes. En plus de cela, le processus de fabrication s'avère difficile dans la plupart des entreprises de conception mécanique. Tout cela pousse les organisations à penser à l'ingénierie simultanée pour optimiser le travail sur leurs produits, améliorer leurs performances, assurer une communication continue entre les différentes étapes d'un PDP (Processus de développement de produit) et par conséquent, faire face à une compétition ardue sur le marché de la CAO. L'ingénierie simultanée est également

utilisée dans la réingénierie du PDP, autrement dit, elle permet de décomposer et combiner un PDP existant et replanifier ses activités [LIU et coll., 2004].

Les outils de l'ingénierie concourante s'avèrent également être efficaces pour l'intégration de la CAO et de la FAO. En effet, ces outils offrent une certaine flexibilité dans les systèmes PPAO (Processus de planification assisté par ordinateur) et permettent de prendre en compte les problèmes de fabrication dès les premières étapes de la conception. En fait, la tâche la plus difficile dans le projet d'intégration CAO/FAO est la modélisation et la représentation des pièces. Les caractéristiques ou « features » ont été largement utilisées à cet effet au cours des deux dernières décennies. Le problème le plus souvent rencontré dans l'ingénierie de production est justement la difficulté de gérer l'interaction entre ces caractéristiques (features) [MILL et coll., 1993]. C'est pour cela qu'il est vital pour l'intégration de la CAO et la FAO de bien gérer ces interactions. La CPF [DERELI et coll., 2004] (Conception pour la fabrication) est utilisée à cet effet, il permet en effet de prendre en considération tous les aspects de conception et de fabrication dans les premières étapes de la conception. La CPF est considérée comme un outil implémentant l'ingénierie simultanée.

### **II.1.2. L'implémentation de l'ingénierie simultanée**

Plusieurs modèles ont été développés pour implémenter l'ingénierie simultanée. Le CETEM [TANJA et coll., 2002], Modèle d'efficacité d'ingénierie (Engineering Effectiveness Model), a été développé par la fondation nationale de recherche en Allemagne. Ce modèle se compose de trois niveaux; les entrées qui décrivent l'équipe travaillant sur un projet (les membres de l'équipe, la taille de l'équipe, les départements impliqués, les compagnies impliquées, etc.) et la tâche de l'équipe. Le système décrit le degré du parallélisme, la nature des interventions, les itérations au sein du projet. Enfin, les sorties montrent le degré de l'accomplissement des tâches et la qualité du travail fait pour le projet. Le modèle NRPDAP [YAN et coll., 2002] (Network of product-process design activity pairs), le réseau des paires d'activité de conception Processus-Produit, est développé par l'institut de recherche en automatique en Chine. Il représente le processus de conception comme un réseau de paires « Produit - Processus », il détermine le degré de simultanéité entre le produit et les activités qui lui sont reliées dans le

processus de conception, et enfin représente et estime la durée de l'accomplissement des tâches de chaque paire.

Shi-Jie Chen et Li Lin [CHEN et coll., 2002] affirment que sans une réelle coordination entre des équipes multifonctionnelles ([SIEGEL, 1991] [SMITH, 1997]) un projet ne pourrait pas aboutir. Ils proposent un modèle qui a comme objectifs de :

- Identifier toutes les tâches d'un projet; les tâches indépendantes, dépendantes et interdépendantes en utilisant les matrices binaires DSM ([STEWART, 1981a] [STEWART, 1981b]).
- Décomposer les grands groupes de tâches interdépendantes en petits groupes faciles à gérer.
- Assurer une certaine cohérence entre les différentes tâches du projet.

Il existe d'autres applications à base d'agents logiciels (III.2.2) qui ont été développées comme outils d'ingénierie concurrente. Dans le laboratoire d'intelligence artificielle [D'AMBROSIO et coll., 1996] à l'université de Michigan, des chercheurs ont développé une méthodologie à base d'agents pour la prise de décision au sein d'une organisation.

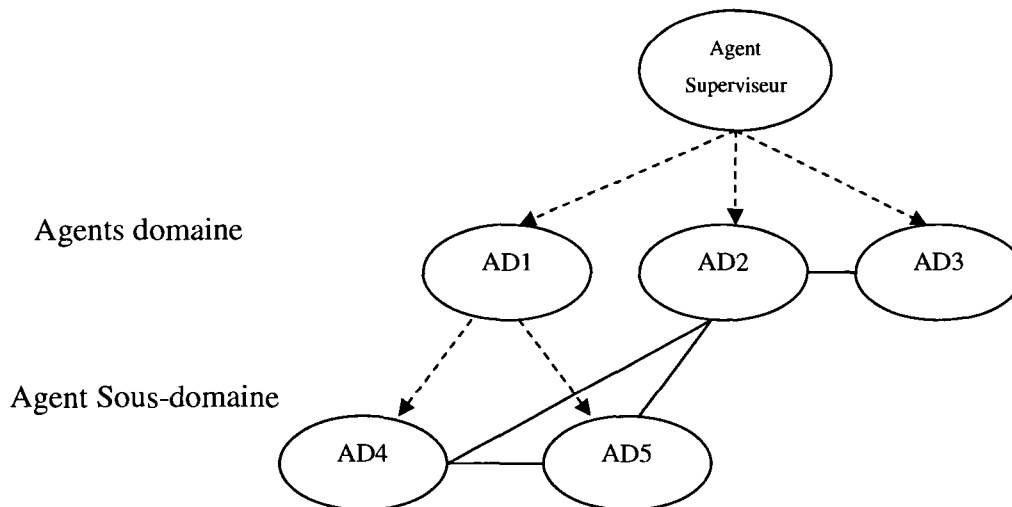


Figure 2.1 : une organisation hiérarchique d'agents



Leur méthodologie se nomme l'ingénierie concourante hiérarchique et fournit un appui aux relations « Superviseur / Subordonné » et « point-à-point » dans le cadre d'un projet.

Un problème typique de l'ingénierie concourante hiérarchique est présenté à la figure 2.1. L'agent superviseur délègue la résolution d'un problème de conception à trois agents subordonnés AD1, AD2, AD3 et leurs transmet également ses préférences et les contraintes qu'ils devraient respecter. L'agent subordonné AD1 délègue la résolution du problème à AD4 et AD5 avec des préférences et des contraintes. La résolution du problème se fait finalement entre les quatre agents AD4, AD5, AD2, AD3 sous la supervision de l'agent superviseur et d'AD1. Les lignes continues représentent le réseau de résolution du problème, les lignes avec pointillés représentent les relations entre les différents agents.

I<sup>2</sup>QFD [KAO et coll., 2002] est un système de Tableau noir à base d'agents qui a été développé pour faciliter la coordination au sein d'un projet de conception. Ce système a été défini avec IDEF0 (diagramme d'activité contenant plusieurs niveaux de décomposition d'un processus en terme de fonctions, d'entrées et de sorties) et des réseaux de Pétri de haut niveau. Basé sur une logique temporelle, I<sup>2</sup>QFD permet de manifester la dynamique d'un processus de conception et de générer une planification pour le projet assurant la simultanéité prévue.

Les grandes organisations dans l'industrie de l'aéronautique utilisent déjà des outils performants afin d'implémenter le concept d'ingénierie simultanée dans leurs entreprises. Cependant, les petites et moyennes entreprises (PME), ont grand besoin de ce genre d'outils [ALBIN et coll., 1996] [USHER, 1996]. Pour assurer leur continuité sur un marché très concurrentiel, les PME se dirigent vers l'implémentation des solutions d'ingénierie simultanée [PAWAR et coll., 1997]. Le projet de CEPRA (Concurrent engineering in practice) [WEBER et coll., 1999] vise à développer un système électronique de consultation (consulting) afin de soutenir ces PME et les aider à choisir l'outil adéquat qui va leur permettre d'implémenter l'ingénierie simultanée d'une manière efficace. Le système comporte trois modules: un module d'évaluation qui évalue les pratiques courantes au sein de l'entreprise et identifie les problèmes qui peuvent être causés par ces pratiques, un module d'identification de solutions qui va améliorer les pratiques de l'entreprise et proposer des solutions aux problèmes et enfin un module de sélection d'outils qui suggère des méthodes et

des outils qui peuvent aider l'implémentation de l'ingénierie simultanée d'une manière efficace.

Les systèmes de gestion de données techniques sont largement utilisés dans l'industrie pour assurer une certaine simultanéité dans le travail des concepteurs. Toutefois, ils ne sont pas considérés comme des outils d'ingénierie simultanée efficaces. Toutefois, ils offrent des fonctionnalités très avancées permettant une certaine coordination entre les différentes équipes au sein d'un projet de conception. Dans la section qui suit, nous allons présenter les SGDT existants sur le marché. Nous allons expliquer leurs avantages et leurs limites ainsi que leurs apports à l'ingénierie concourante.

## **II.2. Systèmes de gestion de données techniques (SGDT)**

Il existe plusieurs logiciels SGDT sur le marché, chaque produit a sa spécificité par rapport aux autres. La concurrence entre ces différents produits est ardue puisque les industriels se demandent constamment si le logiciel qu'ils ont est toujours bon et comment ils peuvent améliorer les solutions SGCV (Systèmes de gestion de cycle de vie) dont ils disposent sans investir dans un nouveau produit. Ils se demandent également comment leurs SGCV pourraient suivre l'évolution des autres systèmes utilisés dans l'organisation tels les PGI ou les systèmes CAO.

Nous présentons dans les sections suivantes les produits SGDT les plus connus sur le marché selon une étude d'AMR Research [BURKETT, 2003].

### **II.2.1. SGDT indépendants**

Les SGDT indépendants sont des SGDT vendus indépendamment des systèmes CAO. Une fois achetés ils peuvent être intégrés à n'importe quel système CAO.

**Agile [AGILE]** : Agile SGCV est considéré comme un bon SGDT qui a donné ses preuves au sein de plus de 1200 organisations dans plusieurs industries; automobile, aérospatiale etc. Parmi ces 1200 entreprises on cite : Colubris Networks, Novoste, OnStor, Proxim, Sonic Innovations, Visx, Alcatel, Boeing, Dell Inc., Flextronics International, GE Medical,

Haemonetics, Hitachi, Johnson & Johnson, Leapfrog, Lockheed Martin, Magna Steyr, Siemens, QUALCOMM et ZF.

**Eigner** [AGILE] : Eigner est un groupe allemand acheté par agile. Eigner SGCV ressemblera donc beaucoup à Agile SGCV.

**MatrixOne** [MATRIXONE] : Est un produit américain qui a également donné ses preuves dans plusieurs entreprises, entre autres, adidas-Salomon, Agilent, Autoliv, Boeing, Celestica, Eaton, General Electric, Honda, Honeywell, Johnson Controls, Nokia, Philips, Porsche, Procter and Gamble, Shanghai Bell, Toshiba, and Yazaki trust.

## II.2.2. SGDT liés aux systèmes CAO

Ce type de SGDT est complètement lié à un système CAO donné, Nous citons :

**PTC** [PTC] : La solution PTC est utilisée avec le système CAO PRO/Engineer. PTC propose le système PDS (Product Development System) comme SGCV. Le PDS repose sur trois principes clés; créer, collaborer et contrôler :

- Créer des informations numériques détaillées.
- Collaborer pour prendre des mesures déterminantes et impliquer les équipes projet, les clients, les fournisseurs et les partenaires.
- Contrôler les processus, tels que la gestion des modifications et la revue de conception.

PDS est un produit certifié CMII. Les clients de PTC dans l'aérospatiale et la défense sont : Boeing, Airbus, Lockheed Martin, NASA, Raytheon, General Dynamics, BAE SYSTEMS, Thales, Rolls Royce, Smiths Aerospace. Les clients dans le domaine de l'automobile sont : ArvinMeritor, Cooper-Standard Automotive Group, DaimlerChrysler, Ferrari, Harley Davidson, Hyundai, Peterbilt, Porsche, Siemens VDO Automotive AG, Toyota, TRW Automotive, Volkswagen/Audi.

**EDS** [UGS] : Grâce à l'alliance entre EDS et UGS [EDS], le SGCV UGS a été considérablement amélioré. UGS (Teamcenter PLM, métaphase) est utilisé avec les systèmes

CAO Solid Edge et Unigraphics. EDS ou l'UGS amélioré est utilisé dans plusieurs domaines, entre autres; l'aérospatiale (Eclipse Aviation Corporation), l'automobile (Adam Opel).

**IBM/DS [IBM PLM Solutions]** : IBM propose le produit Smarteam comme SGDT/SGCV. Smarteam est utilisé avec le logiciel CAO CATIA. Il est utilisé par Bombardier, Pratt & Whitney et autres industriels, il est intégré à ENOVIA VPLM.

### **II.2.3. SGDT intégrés dans des PGI**

Ces produits sont des systèmes PGI qui contiennent des SGDT.

**Baan [CRM]** : Baan a fait son entrée sur le marché du SGCV avec Oracle en 2003. Il a été acheté par SSA Global dans la même année. Baan (ou SSA) propose un système PGI qui intègre en lui-même un système SGDT/SGCV. Baan a des clients dans le monde entier surtout au Japon et en Chine.

**Oracle [ORACLE]** : Oracle a ajouté un module SGCV dans le système PGI Oracle. Ce module a été développé avec plusieurs partenaires, notamment, CH2M HILL, Honda Motorcycle, Pella Corporation et Technip-Coflexip. AMR Research [AMR Research] affirme qu'Oracle SGCV pourrait être très compétitif sur le marché du SGCV.

**SAP [COLIN, 2001]** : My SAP SGCV est un module du système PGI MySAP. MySAP SGCV intègre tous les participants au processus de développement : concepteurs, fournisseurs, fabricants, et clients.

### **II.2.3. Comparaison entre les différents SGDT**

Nous proposons dans cette section une comparaison entre les différents SGDT cités plus haut selon plusieurs critères :

a) Lien du logiciel SGCV avec le client :

- Le GBC (CNM) (Gestion des besoins du consommateur) est un module au sein du SGCV qui vise à prendre en compte les besoins des clients. Nous présentons dans ce qui suit les fonctionnalités les plus importantes du GBC et les logiciels SGDT qui les implémentent :
  - Product configuration (PC) : Offre la possibilité de configurer les options du produit et les mettre à jour. (Oracle, Baan, Matrix One, PTC)
  - Concept BOMs (CBOM) : Définit un BOM normalisé incluant tout les BOMs existants dans le système. (EDS et MatrixOne)
  - Requirements Managements (RM) : Couvre les révisions continues entre les contraintes et le document de conception. (EDS et IBM)
  - WMT (Web Market Test) : Permet de rassembler les rétroactions des consommateurs sur un produit donné.

b) L'amélioration de la conception collaborative et la gestion de l'interopérabilité entre les systèmes CAO :

- SGDT indépendant : Agile a gagné avec l'acquisition d'Eigner un outil d'intégration entre les systèmes CAO. MatrixOne vient de finir son produit « Integration Exchange Framework » qui offre d'autres possibilités d'intégration.
- SGDT lié aux systèmes CAO : Le produit « PRO/E Wildfire [Parametric Technology Corporation, 2003] » améliore le partage des modèles de données entre « Windchill PDMLink [PLM Community] » et « Windchill ProjectLink ». L'outil « Teamcenter Community [BURKETT, 2003] » d'EDS fournit un bon moyen d'échange des vues 3D des produits via le WEB.
- SGDT intégrés dans des PGI : L'outil « Oracle CADView-3D [ORACLE 2003] » de Oracle permet aux utilisateurs d'accéder à toutes les données liées à un modèle à partir de la représentation du modèle lui-même et ce, via « Oracle E-Business Suite [Quest software] ». SAP et Baan le permettent aussi.

c) Gestion des données de produits :

La gestion des données de produits est la fonction la plus mature des SGCV. Les

industriels exigent différentes fonctions des SGCV dont ils disposent concernant la gestion des données. Nous citons dans quelques points ces fonctions :

- L'interopérabilité avec le système CAO utilisé est une fonctionnalité qui est largement développée par Dassault Systems, EDS et PTC.
- Chez certains industriels, il existe des environnements de développement multi-CAO qui échangent non seulement des données géométriques, mais aussi des données électriques, hydrauliques ou autre. Eigner est le SGCV qui pourrait les convenir.
- La mise à jour des informations contenues dans la liste des fournisseurs à chaque fois qu'il y a un changement dans les documents de conception, est une compétence d'Agile [BURKETT et coll., 2003a].
- Le produit conçu à la commande et la création ou l'adaptation d'un produit aux options spécifiques d'un client (Engineer to order : ETO [BURKETT, 2003]) est une fonctionnalité offerte par : EDS, Eigner, Baan et SAP. EDS se trouve être le plus fort dans ce créneau [BURKETT et coll., 2003a].

d) Gestion des ressources lors du développement d'un produit

- **Les Applications PPM** (Product portfolio management) gèrent les portefeuilles de projets. Jusqu'à maintenant, SAP est le leader dans ce domaine. L'acquisition de « Novare » et de « ProductFactory » et « MS2 » [BOURKE, 2003] [BURKETT, 2003] par PTC et Agile respectivement est un vrai début vers une stratégie de gestion de portefeuille pour ces deux entreprises.
- **Direct Materials sourcing** (DMS) [BURKETT, 2003] : Les applications DMS permettent de mettre à jour continuellement la liste des fournisseurs en y mettant les fournisseurs qui proposent des offres intéressantes et moins chères. Les vendeurs de SGCV offrent des méthodes différentes qui permettent l'envoi des RFQ (Requests for quote) aux fournisseurs pour obtenir des propositions de prix. Les deux SGCV qui offrent la possibilité de faire une analyse de prix sont Agile et EDS. EDS offre en plus la possibilité de faire des enchères.

Nous présentons dans le tableau 2.1 un résumé de la comparaison faite plus haut.

Vendeur	Gestion de portefeuille	CNM	Design collaboratif	Gestion des données des produits	DMS
Agile					
Agile (Eigner)					
EDS					
IBM/Dassault					
MatrixOne					
PTC					
SAP					
Oracle					
Baan					

	Fonctionnalité très forte
	Fonctionnalité forte
	Fonctionnalité bonne

Tableau 2.1 : Comparaison entre les différents SGCV[BURKETT et coll., 2003a]

## II.2.4. Conclusion

Malgré la grande évolution que connaissent les logiciels de SGDT, il existe quelques fonctionnalités importantes qui ne sont pas assez développées voire même inexistantes. En effet, la gestion du changement à l'extérieur du logiciel CAO n'est pas prise en charge par les SGDT. EDS et Smarteam offrent maintenant la possibilité de vérifier les contraintes sur les

produits, cependant ces outils ne font qu'assister les utilisateurs à faire eux même la vérification des contraintes du cahier des charges. En plus, les SGDT n'offrent aucune intelligence qui permette d'assister à la prise de décisions. En effet, les décisions prises dans des organisations implémentant l'ingénierie simultanée, nécessitent l'intervention d'un très grand nombre de relations entre des données contenues dans un grand nombre d'éléments de planification. Dans les sections suivantes, nous allons voir s'il existe des outils autres que les SGCV qui permettraient de faire la gestion des changements d'ingénierie et d'assister à la prise de décision.

### **II.3. Mécanismes de gestion des changements d'ingénierie dans l'industrie**

Les changements d'ingénierie ou « Engineering Change (EC) » permettent de demander et d'effectuer des changements sur des produits. Ces produits peuvent être des documents, des pièces achetées ou fabriquées, des processus et même des commandes [DEC et coll., 1998]. Ces changements d'ingénierie affectent de 70% à 80% du coût final du produit [MCINTOSH, 1995]. Pour diminuer le coût des produits, il est très important de réussir la gestion de ces changements, et de s'assurer que chaque changement demandé est transmis à la bonne place au bon moment. En fait, les changements d'ingénierie dépendent fortement des processus d'ingénierie et de la technologie de l'information utilisée au sein d'une organisation. Dans les EDM (Engineering data management) et les SGDT, il existe déjà des modules de gestion du changement qui vont assister les concepteurs à entreprendre eux même la gestion des changements demandés. Cependant, ils sont loin de satisfaire les besoins de l'entreprise surtout dans un cadre d'ingénierie simultanée.

Dans un environnement distribué, il y a une multitude de modifications et de changements qui devraient être fait à travers le processus global de conception afin de faciliter la fabrication, éliminer les conflits, et satisfaire les contraintes sur les produits. Dans un tel environnement, il pourrait arriver que plusieurs entreprises ou filiales décident de s'impliquer toutes ensemble dans la conception d'un produit. Dans ce cas, il faudrait veiller à ce que l'échange de données entre ces différentes firmes soit bien réussi en s'assurant que :



La communication entre les compagnies impliquées dans la conception soit étroite et débute dans les premières étapes de conception.

- La communication entre les différentes disciplines au sein de l'organisation soit également étroite et débute dans les premières étapes de conception.
- La communication entre les différentes disciplines au sein de l'organisation soit également étroite et débute dans les premières étapes de conception.
- La réponse aux changements d'ingénierie soit rapide.
- Des logiciels de gestion des données soient disponibles.

Les industriels pourraient décider par exemple, de coopérer avec des fournisseurs ou des partenaires pour plusieurs raisons. Cela pourrait être plus rapide et moins coûteux de développer une tâche chez un partenaire plutôt que d'essayer de développer les outils nécessaires à sa fabrication au sein de l'organisation. En plus, chaque partenaire apporte ses propres compétences pour mener à bien la conception du produit, ce qui permet d'assurer une bonne qualité au produit.

La coopération dans un environnement multidisciplinaire peut réduire légèrement les changements d'ingénierie. Une bonne gestion du changement pourrait avoir plus d'impact sur les changements d'ingénierie en les réduisant considérablement.

Il existe trois types de changements d'ingénierie :

- Les changements d'ingénierie durant la conception : Leurs impacts est minime.
- Les changements d'ingénierie durant la production : Les changements durant la production peuvent causer plusieurs problèmes. Dans certaines disciplines, un produit n'entre en production que s'il est approuvé par plusieurs organisations voire même des agences gouvernementales. Les changements demandés après l'approbation nécessitent donc des procédures de gestion très spéciales.
- Les changements d'ingénierie durant la reconstruction d'un produit : Dans le cas de la création d'autres versions d'un produit

Plusieurs stratégies sont utilisées pour minimiser l'impact de la demande de changement. Tout d'abord, il est préférable d'éviter les changements en passant plus de temps à vérifier les produits avant de les approuver [CLARK et coll., 1991]. Ensuite, il vaut mieux effectuer les changements dans les premières étapes du processus de conception [TERWIESCH et coll., 1999] [HUANG et coll., 1999]. Enfin, il est important d'essayer de faciliter le processus du changement quand il s'avère indispensable. Ces techniques n'ont malheureusement pas été développées pour un environnement multidisciplinaire. Elles pourraient convenir à une entreprise qui travaille seule sur un projet.

Devant l'obligation d'implémenter l'approche ECM (Engineering Change Management), certaines organisations se sont contentées d'améliorer leurs procédures de gestion du changement [HARHALAKIS, 1986]. D'autres ont décidé d'utiliser des systèmes qui pourraient les aider à mieux gérer les changements d'ingénierie. Nous présentons dans la section suivante quelques travaux réalisés dans le domaine de la gestion du changement.

### **II.3.1. Travaux déjà réalisés dans le domaine de la gestion du changement**

#### **A) La gestion des changements dans l'ACE (Allied Concurrent Engineering)**

L'ACE est un processus de collaboration distribué dans lequel différentes entreprises ou disciplines peuvent coopérer afin de concevoir un produit et développer les processus qui lui sont reliés. En fait l'ACE combine les deux notions de collaboration virtuelle [SHAH et coll., 1995] et ingénierie concourante. L'approche ACE partage efficacement les données et les processus d'un produit pendant tout son cycle de développement. Le projet ACE est divisé en deux phases. La première phase développe la méthodologie de gestion des changements. La deuxième phase développe le système de gestion des changements. Yuh-Min Chen et compagnies [CHEN et coll., 2002] ont étudié le modèle de référence de l'ingénierie du changement (EC) (figure 2.2) avant de développer leur propre modèle.

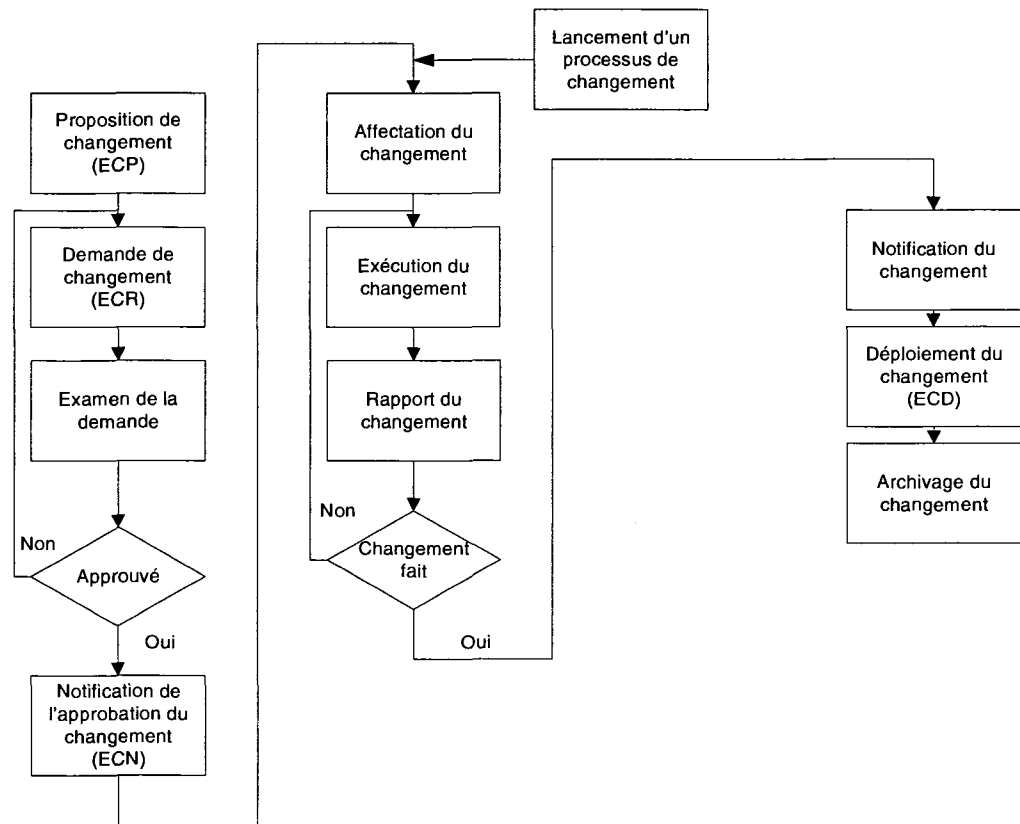


Figure 2.2 : Le modèle de référence de la gestion du changement dans l'ingénierie [SHAH et coll., 1995]

L'ECR (Engineering change request) est un document qui explique en détail les changements d'ingénierie demandés avec les justifications. C'est la première action à entreprendre une fois que la décision de faire un changement est prise. Une fois que l'ECR est approuvé, l'ECN (Engineering change notification) est ensuite rédigé, il est envoyé aux personnes concernées par le changement afin de les informer que le ECR est approuvé, et leurs expliquer les détails du changement demandé. Une fois que les changements sont faits, les résultats sont envoyés aux personnes appropriées. Les résultats sont également envoyés à certaines applications telles les PGIs ou MRPs (Material requirements planning) pour une utilisation ultérieure. Cette étape est référée par l'ECD (Engineering change deployment). Dans la dernière étape du processus, toute l'information sur les résultats ainsi que les impacts des changements est

enregistrée. Le modèle de l'ACE utilise essentiellement le modèle de référence (figure 2.2).

Le processus de changement dans l'approche ACE commence par une proposition de changement. Après c'est la consultation et l'approbation de cette proposition. Un projet de gestion du changement est lancé pour gérer le changement demandé, il est considéré comme une tâche du processus ACE. Le projet de gestion du changement d'ingénierie peut être défini comme un sous-projet du projet ACE. Toutes les activités qui pourraient être influencées sont identifiées. En plus, l'ECM contrôle ces activités et transmet les informations qui les concernent lors de l'exécution du processus de changement. Les étapes de l'exécution, le déploiement et l'archivage, sont similaires à celles déjà définies dans le processus de référence à la figure 2.2. Dans la figure 2.3 nous représentons le processus de gestion du changement dans le système ACE.

L'ACE a donc comme objectif d'intégrer efficacement toutes les ressources d'une entreprise. Autrement dit, les changements sont propagés vers les autres systèmes utilisés dans l'entreprise. Le succès du processus d'intégration dépend fortement de l'intégration des différents systèmes d'information de l'organisation. L'implémentation de l'ACE est donc une solution d'intégration pour l'entreprise plutôt qu'un nouveau module au sein du système d'information déjà existant [GRUNINGER et coll., 1996] [BERNUS et coll., 1994]. L'ACE est « Orienté-Processus », « Centré-Produit », « Basé-Projet ». Nous expliquons dans ce qui suit ces trois concepts :

- **Orienté-Processus** : Le processus de gestion du changement est configuré selon le processus d'ingénierie défini pour un produit donné.
- **Basé-Projet** : Un projet est défini pour développer le produit; la planification au sein du projet détermine le temps de chaque activité et les ressources qu'il faudrait lui allouer.
- **Centré-Produit** : Les composants d'un produit sont définis dans la structure du produit. ACE assure une communication entre ces différents composants ainsi qu'avec les processus de conception reliés au produit.

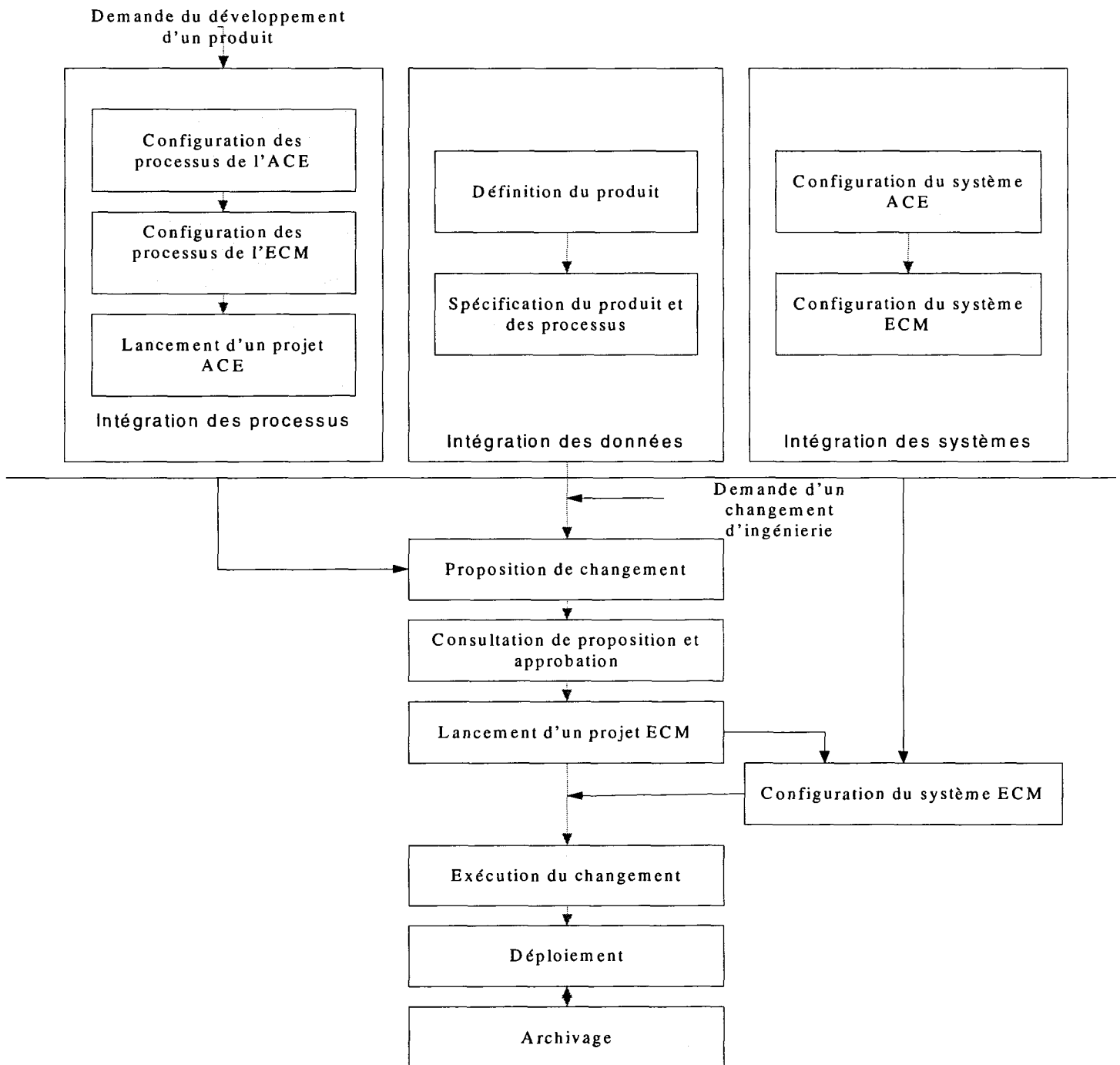


Figure 2.3: Gestion du changement dans le système ACE [CHEN et coll., 2002]

## B) WEB-Based ECM

Dans ce prototype, la technologie WEB est utilisée pour développer un système ECM. Cette

approche offre plusieurs avantages. Premièrement, la quantité de documents générés lors d'une demande de changement est réduite considérablement. Deuxièmement, le temps de traitement est diminué. Les données générées par le changement sont partagées et transmises aux parties concernées.

La figure 2.4 montre l'architecture globale de l'application WEB-Based ECM. Cette application n'intègre pas les données des produits et les processus gérés normalement par les SGDTs. Ceci permet évidemment d'alléger le système puisqu'il s'occupe seulement de la procédure de la gestion du changement. Cependant, cela peut gêner la procédure d'évaluation de l'impact du changement sur les opérations de l'entreprise. Selon les développeurs de ce prototype, un compromis serait d'inclure les informations pertinentes concernant les produits et les processus dans leurs systèmes et laisser la gestion approfondie de ces derniers aux systèmes spécialisés.

La base de données ECM est relationnelle et supporte le langage SQL. Le serveur web de l'ECM propose une page web composée de plusieurs formulaires. Chaque formulaire réalise quelques fonctionnalités qui supportent plusieurs activités du cycle de vie du changement d'un produit. Le serveur d'application collecte et distribue les données aux clients, il communique également avec la base de données pour la mettre à jour ou en extraire les données. Les clients ECM ne font pas parti du système avant de se connecter au serveur web.

Le formulaire «EC Log» montre un résumé des changements déjà entrés dans le système ECM. Le formulaire «EC Request» permet aux utilisateurs de proposer un changement. Le formulaire «EC Evaluation» montre l'analyse faite par le système des impacts du changement demandé sur les produits et les processus selon les données introduite dans le système par les développeurs. Le formulaire «EC Notice» est utilisé pour aviser les différentes disciplines de l'approbation du changement, il permet aussi de visualiser d'autres notifications déjà faites pour d'autres demandes de changements.

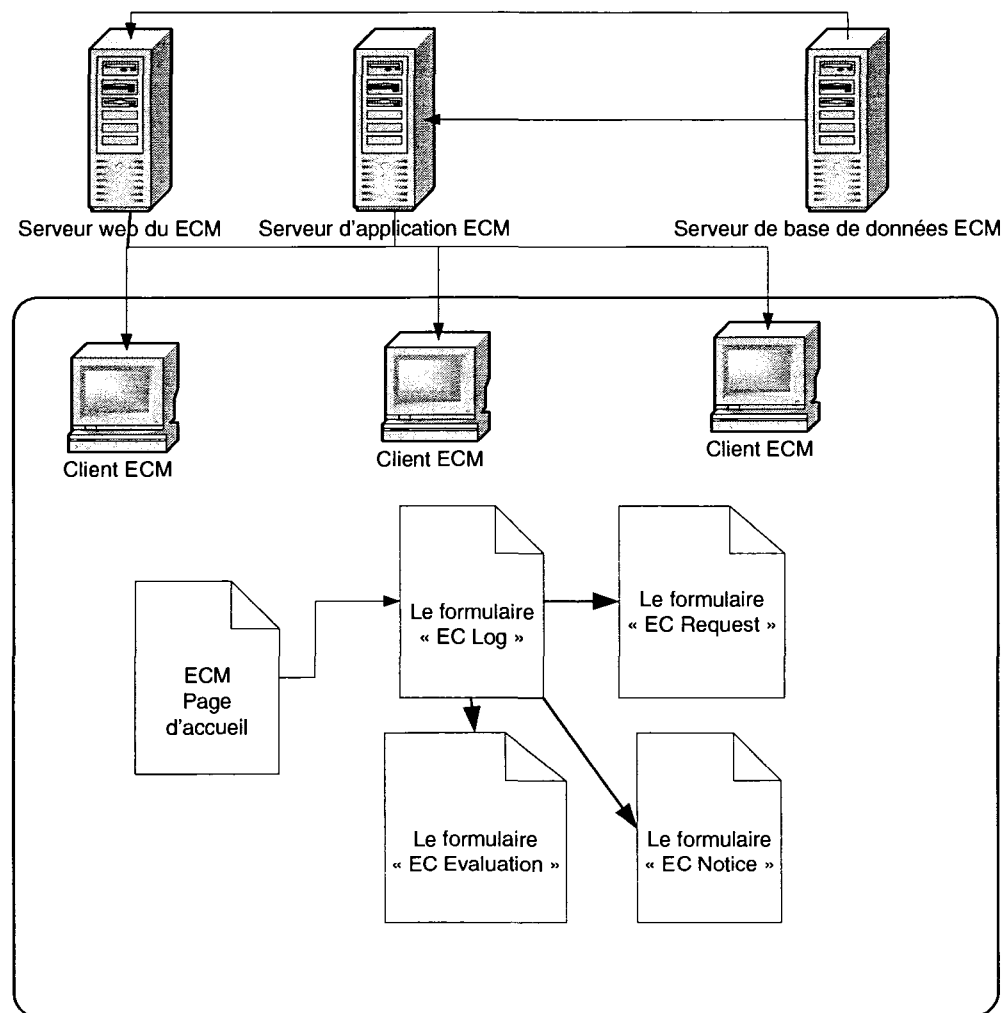


Figure 2.4 : Architecture globale du système WEB-Based ECM [HUANG et coll., 2001]

### C) Projet SIMNET

Le projet SIMNET [ROUIBEH et coll., 2003a] propose une approche pour la gestion du changement basée sur la notion de «paramètre». Les membres du projet ont effectué des entrevues avec plusieurs ingénieurs de «Siemens SGP Transport Systems» qui a subventionné le projet SIMNET et ils se sont rendu compte que les ingénieurs voient leurs travaux comme un ensemble de prise de décisions (intentions des concepteurs) et non comme des documents ou des processus. Ces décisions d'ingénierie déterminent des variables d'ingénierie. Les objets les plus élémentaires de ces variables d'ingénierie sont les « paramètres ». Les paramètres

représentent en fait, une grandeur spécifique à une situation d'ingénierie. Ils peuvent référer à des dimensions, des forces ou des mouvements. Durant une conception collaborative, plusieurs intervenants peuvent changer ou modifier la valeur d'un paramètre. La relation entre les paramètres détermine les interactions entre les différents décideurs lors de la conception collaborative, notamment les fournisseurs et les partenaires d'une compagnie.

Un paramètre est défini par :

- Sa valeur : Les valeurs des paramètres et les relations entre eux reflètent les spécifications de la conception.
- Son degré de maturité : Donne le degré de certitude du paramètre. Valeur estimée ou valeur exacte.
- Son statut : L'assignation des valeurs aux paramètres passe par plusieurs étapes. (Par exemple: « prédéfini, en travail, en approbation, libéré »)

L'approche basée sur les paramètres introduit la notion de « rôle ». Les rôles agissent directement sur les paramètres. Il y a cinq catégories de rôles [ROUIBEH et coll., 2003b] :

- Le coordinateur : Est la personne techniquement responsable du paramètre et qui surveille son évolution.
- Les collaborateurs : Ce sont des ingénieurs travaillant tous sur le même paramètre, mais chacun d'un point de vue différent.
- Les commentateurs : Tous les gens qui utilisent le paramètre, mais ne le conçoivent pas, tel un planificateur de production qui devrait décider si une pièce avec certaines dimensions peut être usinée ou pas.
- Les participants : Les personnes qui désirent être informées du développement de certains paramètres sans pour autant travailler sur le paramètre. S'ils prévoient travailler sur ce dernier, ils sont au moins au courant des changements,
- Le superviseur : Il s'occupe de l'opération « libération » d'un paramètre.

La capture des relations entre les paramètres donne un réseau de paramètres. Ce réseau



supporte la gestion de changement à travers cinq étapes importantes :

- Définir le changement désiré
- Identifier les paramètres à changer
- Contrôler le changement en le propageant vers les autres paramètres liés aux paramètres à changer en s'assurant que le changement a bien été effectué
- Envoyer un rapport du changement aux personnes intéressées.
- Enregistrer le changement.

Une fois qu'une demande de changement est formulée, le coordinateur crée une liste de demandes de changement – si elle n'a pas déjà été créée - et inclut le paramètre en question dans la liste. Les collaborateurs sont ensuite informés. Les coordinateurs et les collaborateurs se mettent d'accord sur une valeur pour le paramètre. Une fois que le changement est approuvé, le système identifie les autres paramètres concernés et décide si le changement doit affecter chaque paramètre identifié. Enfin, les changements approuvés sont réalisés et enregistrés pour une éventuelle réutilisation (figure 2.5).

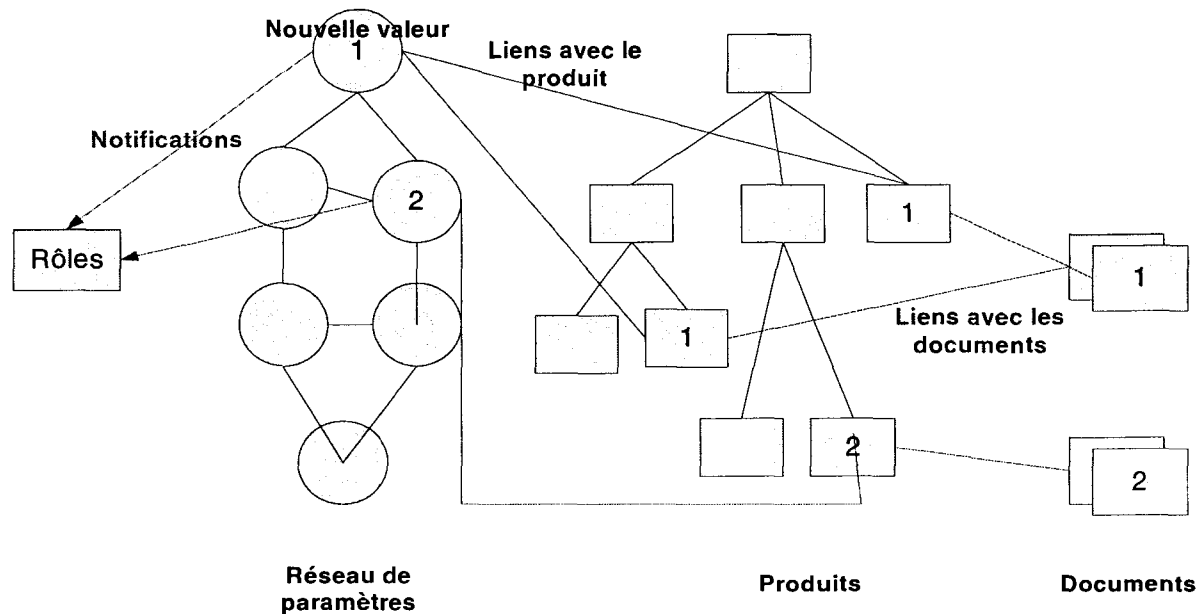


Figure 2.5: Plateforme de communication supportant une gestion de changement intelligente

### D) Les systèmes de conception basés sur les caractéristiques (feature-based)

La représentation des produits en utilisant les caractéristiques se base sur deux concepts : La reconnaissance des caractéristiques et la conception par caractéristiques. Cette approche facilite grandement le partage de l'information sur les produits entre différentes disciplines. En effet, le produit partagé offre plusieurs vues aux différentes applications qui l'entourent. Ces vues sont des interprétations différentes du produit selon le contexte dans lequel il est visualisé. Pour assurer cette flexibilité, un modèle de données dynamique a été défini; il se nomme le modèle intermédiaire (IM) (figure 2.6) [MARTINO et coll., 1998].

Dans un contexte donné, l'expert peut faire une demande de changement. La demande est envoyée via l'IM au concepteur qui va exécuter le changement après une négociation avec les autres disciplines concernées et selon une certaine priorité. Finalement, les modifications acceptées sont réalisées directement dans l'IM et propagées aux différentes vues qui dépendent de la vue dans laquelle la modification a été faite.

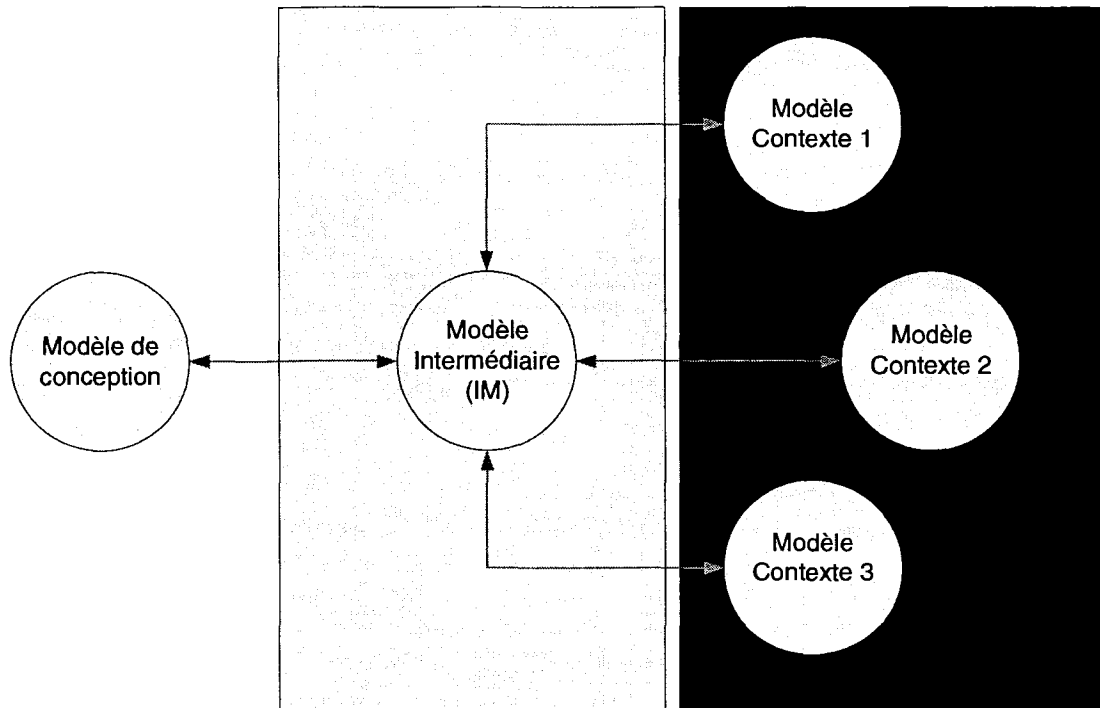


Figure 2.6 : Le modèle intermédiaire (IM) dans une représentation par caractéristiques

Le modèle intermédiaire IM est un modèle à multiples points de vue, il permet la

représentation de plusieurs ensembles de caractéristiques dans une même structure. Puisque les caractéristiques ne sont pas toujours disjointes et peuvent interférer complètement ou partiellement avec chaque entité dans le document de conception, on lui associe l'information concernant le contexte et la caractéristique auxquelles elle appartient. De cette manière, il est possible de partager le modèle entre plusieurs contextes.

Lors d'une modification dans un document de conception, il est possible qu'il y ait des conflits entre les différents contextes du modèle. La négociation de la modification est gérée par le concepteur qui s'occupe de propager la demande de changement aux différentes disciplines. S'il n'y a pas de conflits, la demande de modification est acceptée et le changement est fait. Dans la figure 2.7 nous présentons un objet décrit selon deux points de vue différents. Dans un contexte de conception, il est décrit comme une combinaison d'un bloc avec deux nervures. Par contre dans un contexte d'application, il est décrit comme un bloc avec une fente. Les caractéristiques des deux représentations partagent les deux surfaces  $f_1$  et  $f_2$ . Si nous modifions la largeur de la fente  $w_3$  la modification va influencer les caractéristiques des deux nervures dans le contexte de conception. Dans le modèle intermédiaire,  $w_3$  est exprimée comme la distance entre les deux surfaces parallèles  $f_1$  et  $f_2$ . Ces deux surfaces sont des entités partagées appartenant aux nervures dans la vue de conception et à la fente dans la vue d'application. Ainsi, lorsque la modification de  $w_3$  est propagée aux nervures, les valeurs de  $w_1$  et  $w_2$  sont déduites de la valeur de  $w_3$ . Les contraintes sur les nervures doivent être considérées pour décider si la modification de  $w_3$  doit être acceptée ou rejetée. La communication entre les différentes vues et le modèle intermédiaire est assurée par CORBA (figure 2.8).

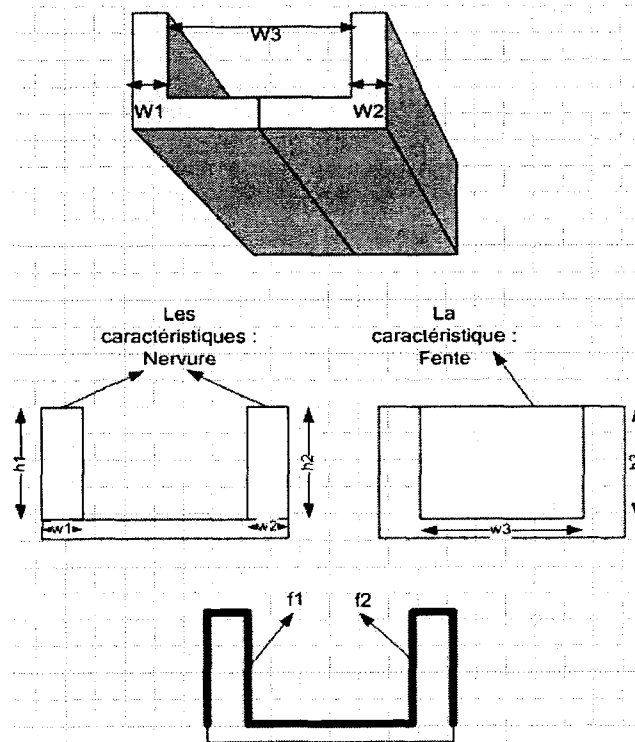


Figure 2.7 : La représentation d'un modèle selon différents contextes [MARTINO et coll., 1998]

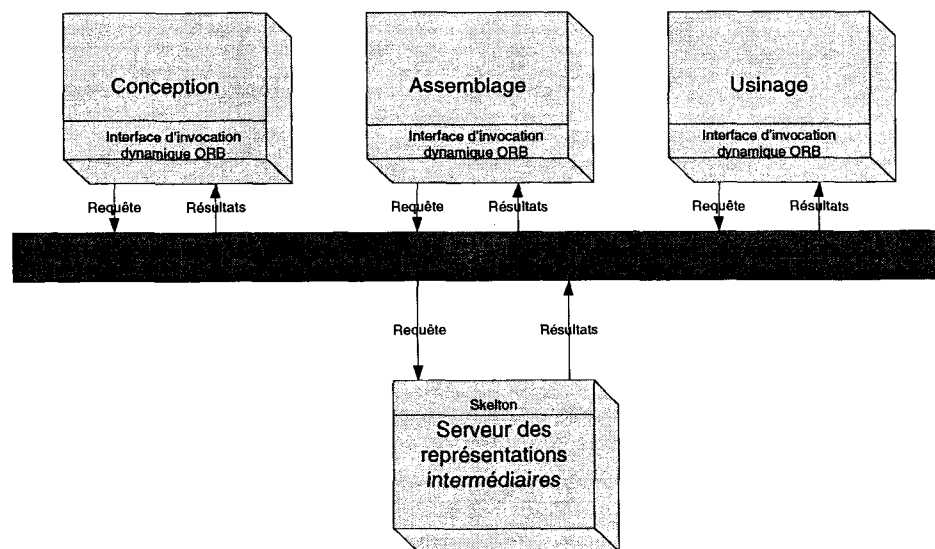


Figure 2.8 : Architecture distribuée du système [MARTINO et coll., 1998]

### **E) Modèle d'information pour la gestion des changements d'ingénierie dans un environnement collaboratif Information**

La conception dans le domaine de la construction nécessite un travail collaboratif entre les différentes disciplines impliquées [MOKHTAR et coll., 1998]. Les concepteurs doivent parfois prendre des décisions vis-à-vis des documents de conception même s'ils sont parfois géographiquement séparés. Pour maintenir une certaine cohérence au sein du système il faut absolument que les informations concernant les changements faits, soient coordonnées. Dans un projet de construction, les données qui sont utilisées comme entrée d'une certaine discipline constituent probablement les données de sortie d'une autre. Donc, il peut arriver qu'un concepteur modifie les données de sortie d'une discipline en oubliant que c'était déjà les entrées d'une autre ce qui va certainement créer des incompatibilités au sein du système. L'étape du processus de conception qui s'avère être la plus critique est l'étape de conception détaillée. C'est là où les différents documents de conception sont générés. Donc, une petite erreur dans ces documents peut générer des pertes considérables lors de la production; telles une inconsistance dans les données de conception ou une disparité entre les pièces reliées ou encore le dysfonctionnement de certains composants. Ces erreurs peuvent causer des pertes exorbitantes, des ruptures de contrats, des retards de livraison, etc. Selon l'auteur de l'article, ces problèmes peuvent affecter non seulement l'industrie de la construction, mais aussi l'économie nationale puisque les investisseurs vont probablement cesser d'investir dans la construction ou que les prix des services et des produits de construction vont augmenter d'une manière exponentielle. Cet article [MOKHTAR et coll., 1998] présente un modèle d'information développé et implémenté dans l'école de construction de l'université de Concordia afin d'améliorer la coordination entre les différents intervenants dans un processus de conception. Ce travail peut inspirer les chercheurs dans le domaine de la conception mécanique vu qu'ils sont confrontés au même problème. L'objectif de ce travail est l'utilisation des technologies de l'information au service de la conception dans le domaine de construction afin de renforcer le processus de coordination entre les différentes expertises intervenantes dans un projet de construction.

L'approche décrite dans le présent article propose un modèle d'information qui a comme

caractéristiques :

1. La capacité de gérer les changements de conception
  - La propagation du changement vers les disciplines touchées.
  - L'enregistrement de tous changements faits sur les documents de conception
  - La planification des changements futurs
2. Les informations liées à la conception sont stockées et échangées sous forme de données de conception plutôt que sous forme de schémas ou « Drawings ».
3. Une seule source de données contenant tous les composants nécessaires pour un projet de construction

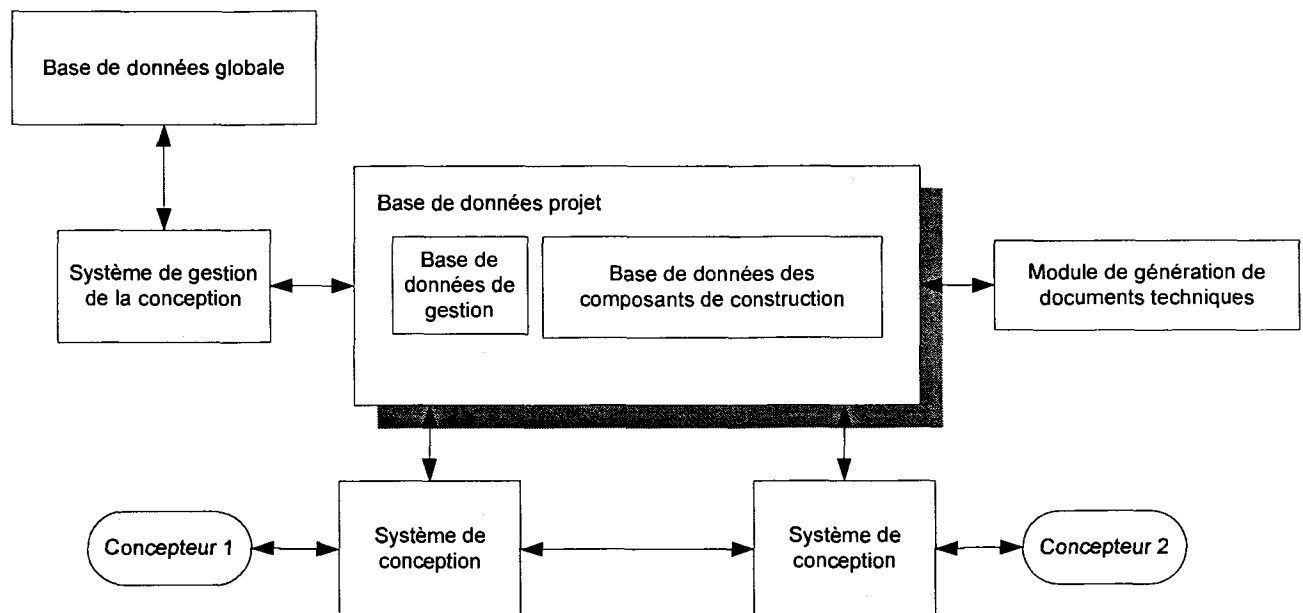


Figure 2.9 : L'architecture du modèle d'information proposé : Architecture distribuée du système [MOKHTAR et coll., 1998]

Le cœur du modèle proposé est la base de données projet, elle est composée de deux parties (figure 2.9); la base de données de gestion et la base de données des composants de construction. La base de données des composants contient toutes les données nécessaires à la description de n'importe quel composant de construction au sein d'un projet. Ces données permettent de produire les documents techniques pour la construction. La base de données de

gestion contient à son tour toute l'information permettant de gérer les différentes fonctions du modèle d'information. Afin de générer les données incorporées dans la base de données des composants, les différents concepteurs sont connectés à la base de données projet via le module système de conception. Le gestionnaire de conception est le responsable de la gestion des documents techniques reliés à la production, il est également connecté au module central (Base de données projet) via le système de gestion de conception. Une fois que la base de données des composants de construction est remplie, les documents techniques de construction sont générés à partir du module de génération des documents techniques. Lors de l'étape de conception détaillée, les concepteurs traduisent les idées figurant dans les documents de conception préliminaire en composants de construction élémentaires (mur, colonne, porte...). Les concepteurs de différentes disciplines interagissent avec les données de conception dans la base de données via le module de manipulation des données se trouvant à l'intérieur du système de conception afin de visualiser, modifier ou rajouter des composants. Pour assurer la gestion des changements d'ingénierie, l'auteur propose d'allouer la tâche de propagation des changements aux composants de conception. Chaque composant est en fait capable de trouver toutes les disciplines qui peuvent être touchées par le changement de l'un de ses attributs et leur envoi un message spécifiant le changement fait et comment cela peut les affecter. La base de données projet devient alors une base de composants actifs qui aident les concepteurs à collaborer entre eux. Pour que les composants actifs s'occupent de la propagation, il faut qu'ils soient équipés d'une base de connaissances qui va les aider à identifier les disciplines affectées par le changement. Cette base de connaissance est composée de règles de « pre-construction » et de règles dynamiques. Les règles de « pre-construction » sont définies dans la phase de conception détaillée alors que les règles dynamiques sont définies lors de la préparation des dessins de conception ou encore lors de l'utilisation du modèle. Quand un concepteur effectue un changement sur un ou des attributs d'un composant de construction, ce dernier vérifie toutes les règles dans la base de connaissance, si l'une de ces règles s'applique sur le changement effectué il envoie des messages aux personnes concernées. Tous les changements effectués sur les modèles et les messages envoyés sont enregistrés dans la base de données de gestion. Ils peuvent être utilisés lors de la création des documents techniques de conception ou aider à analyser le processus de production en évaluant par exemple les performances des concepteurs. Les concepteurs ont aussi la

possibilité avec ce modèle de réaliser des plans de changement à l'avance, il faudrait par contre avoir assez de connaissances pour déterminer les relations entre les composants. Il s'agit d'identifier des chemins de changements (change path) possibles (figure 2.10). Pour valider le modèle présenté, l'auteur propose d'utiliser une étude de cas qui va démontrer qu'une incompatibilité dans les données de conception peut mener à un dépassement de coût équivalent à 2 000 000 \$

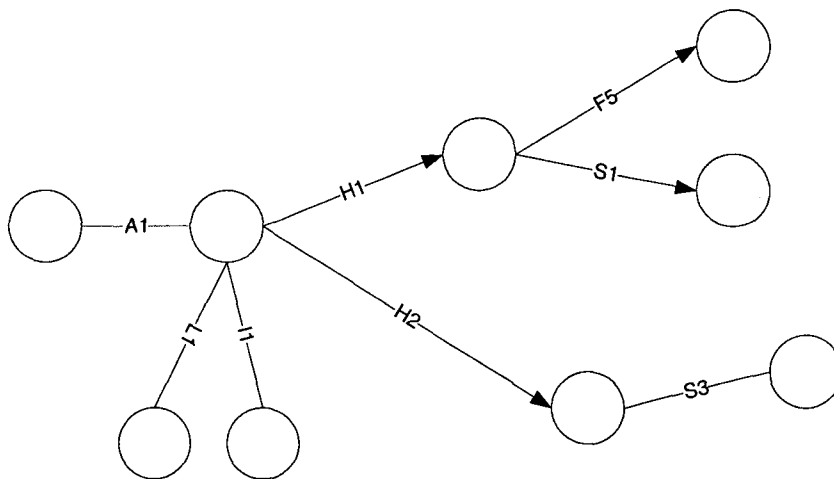


Figure 2.10 : Propagation du changement entre composants

L'auteur a exploré dans cet article plusieurs domaines de recherches lors de sa recherche bibliographique; la gestion des données de conception (Krishnamurthy and law, Platt), la collaboration à base d'agent logiciel (DICE, IBDE, SEED...), la création d'un modèle de base de données pouvant contenir toutes les données de conception dans le domaine de construction, les normes d'échange de données. L'auteur assume que le peu de travaux de recherche qu'il a trouvé n'ont fait qu'explorer le domaine de la collaboration entre différentes disciplines sans trouver un système qui fait cette collaboration d'une manière efficace. Il affirme également qu'aucun standard sur l'organisation des données de conception dans une seule base de données structurée n'a pu être adopté.

Le travail présenté par l'auteur est très intéressant puisqu'il propose de résoudre l'un des problèmes les plus controversés qu'est la gestion des changements d'ingénierie. Il propose de créer un système de construction à base de composants actifs qui serait capable de



communiquer entre eux pour coordonner tout changement fait par les concepteurs. Les liens entre ces composants sont définis par les concepteurs.

Pour valider ce projet, l'auteur devrait penser au nombre énorme de liens à définir quand il s'agit d'un vrai et gros projet de construction. Ceci peut effectivement causer un vrai handicap pour le projet, car les concepteurs vont perdre un temps énorme juste pour la définition des liens. Donc, une évaluation du coût de la création des liens donnera une bonne idée sur la performance du système. En plus, l'auteur n'a pas traité le problème de détection de boucle, c'est-à-dire qu'en partant d'un composant, les notifications de changement vont se propager vers d'autres composants et vont peut-être revenir à la fin au même composant de départ. Il faudrait donc prévoir un module de détection et de résolution de boucles. Le système proposé par l'auteur est un nouveau concept de système de construction, il ne pourrait pas être intégré aux systèmes déjà existants. Se procurer ce genre de système peut être effectivement très coûteux pour les industriels puisqu'ils doivent probablement se passer de leur système de conception actuel.

#### **F) La gestion de la complexité dans un environnement CAO**

La conception des produits mécaniques nécessite l'utilisation de plusieurs technologies et l'intégration de plusieurs métiers qui ont chacun leur propre base de connaissance et leur propres outils. Ces métiers ne communiquent malheureusement pas entre eux d'une manière efficace. En plus, un produit est généralement modélisé plusieurs fois sous différents points de vue ce qui mène à des problèmes de cohérence assez sérieux. Lors d'une étude réalisée dans le domaine de l'aéronautique, l'auteur de [EUSTACHE et coll., 2001] a constaté que les problèmes d'incohérence sont souvent causés par les modifications mal propagées. Cet article propose une approche qui permet de mieux gérer la cohérence au sein des produits CAO. Autrement dit, il propose de partager la connaissance des différents métiers et de prendre en considération les interactions entre eux afin de faciliter la gestion des changements d'ingénierie.

L'auteur propose de créer un système à base d'entités et de contraintes, chaque entité est reliée à une référence physique et chaque contrainte est appliquée à une ou plusieurs entités, les

contraintes donnent une idée sur l'intention de conception. Les données touchées par les contraintes peuvent être conceptuelles au début de la conception (immatérielles et non concrétisées) ou concrètes à la fin (face, sommet, rayon...). Ces différents niveaux de détail constituent une hiérarchie. Cette hiérarchie inhérente permet de mieux comprendre niveau par niveau le document de conception en sa globalité. Les contraintes entre les entités (qui peuvent être concrètes ou conceptuelles) sont par conséquent modélisées dans cet article par un graphe hiérarchique non orienté. Plus les détails sont définis plus les contraintes et les entités sont décomposées.

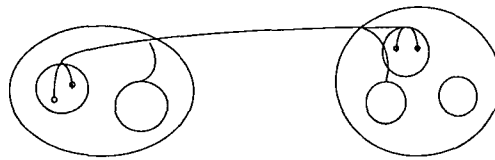


Figure 2.11 : Graphe hiérarchique des contraintes

La modélisation des contraintes sous forme de graphe hiérarchique (figure 2.11) va beaucoup aider les concepteurs à faire la validation des spécifications techniques et la propagation des changements. Les contraintes et les entités ont des attributs (identificateur, nom, référence, type, statut, ...). Les différents types des contraintes sont :

- Dépendante : Cette contrainte peut relier des entités de métiers différents, ce type aide à la propagation de la modification, il informe l'utilisateur en cas de modification pour qu'il procède à la propagation.
- Relationnelle : Contrainte mathématique
- Booléenne : Combinaison de plusieurs contraintes à l'aide des opérateurs logiques
- Règle : C'est une contrainte de condition elle peut être composée de contraintes relationnelles et booléennes

La propagation des changements et l'évaluation des contraintes dépendent de la validation des contraintes et des entités. Quand une contrainte relationnelle ou booléenne n'est plus valide, la contrainte de dépendance informe les entités affectées pour qu'une réévaluation et une

notification soient faites. La contrainte de règle doit alors revérifier ses conditions et réexécuter ses actions. Si une contrainte ou une entité reste invalide même après la réévaluation, toutes les contraintes et les entités « parents » doivent être marquées comme invalide. Une entité ou une contrainte ne peut être validée que si toutes les contraintes et entités « enfants » sont validées. Une interface a été développée au sein du système CAO pour créer, supprimer et modifier des entités. Une entité est modifiée quand sa référence physique (point, surface, solide...) est modifiée ou qu'une contrainte est rajoutée dessus.

Dans l'industrie mécanique, les produits intégrant le travail collaboratif sont de plus en plus en demande. Plusieurs études proposent des modèles sémantiques à un niveau élevé; l'approche «Top-down» et l'approche «Bottom-up». Le premier type d'approche vise à modéliser le produit en exprimant ses spécifications via la réalisation de certaines fonctions. Malheureusement, il n'a pas pu gérer le problème de cohérence quand des données d'un niveau de granularité assez bas sont modifiées. Le deuxième type consiste à ajouter de l'information à un niveau sémantique plus haut en prenant en considération les différents types de métiers intervenant dans la conception, un exemple serait la conception par caractéristique. D'autres approches «Bottom-up» proposent de relier une base de connaissances au système CAO et assurer la cohérence au sein des produits en vérifiant les contraintes et les règles. L'auteur expose d'autres types de recherches qui proposent de maintenir plusieurs vues sur le produit. Cependant, de la manière dont ils veulent le réaliser, ils ne vont qu'augmenter encore plus la complexité de la gestion de la cohérence. L'auteur affirme que les contraintes définies dans le prototype développé sont simples et que d'autres types de contraintes plus complexes pourraient être définies. Il atteste aussi que son système ne traite pas les conflits qui peuvent exister entre les contraintes.

La recherche présentée dans cet article est très intéressante. Le système proposé permet de faire la vérification des spécifications ainsi que la propagation des changements d'ingénierie. L'auteur propose d'utiliser la logique floue dans un développement futur pour pouvoir faire des compromis et valider des contraintes même si elles ne sont pas tout à fait vérifiées du moment qu'elles soient acceptables. Néanmoins, ce système ne fait pas de l'assistance à la prise de décision, autrement dit quand les contraintes ne sont pas vérifiées, il ne propose pas

d'autres alternatives à l'utilisateur. En plus, le nombre d'entités qu'il faudrait définir est énorme surtout dans les gros projets mécaniques.

### **G) Gestion des changements d'ingénierie pour les modèles multi-vues**

La technique de conception par paramètre associe les informations d'ingénierie à la géométrie et à la topologie en utilisant des contraintes. Elle permet aux concepteurs de faire des modifications sur des pièces en changeant la valeur des paramètres. Actuellement, elle est utilisée pour générer des solides 3D paramétriques à base de caractéristiques, et ce, à partir de «Sketch» 2D (Pro engineer). Certains chercheurs utilisent les systèmes de conception paramétrique 3D afin de gérer les relations géométriques entre les différentes vues d'un produit, nous parlons ici des vues au sens projections et non les points de vue métier. Avec les fonctions paramétriques fournies dans un système 3D, un nouveau modèle 3D peut être généré à partir des changements effectués et des projections peuvent être faites par la suite vers les autres vues. Néanmoins, cette méthode est une solution indirecte pour traiter le problème de gestion des changements au sein des modèles multi-vues et qui en plus de cela dispose de deux inconvénients. Le premier est que la capacité paramétrique du système pour des modèles 3D est limitée pour certains systèmes CAO. Le deuxième est que cette méthode n'est pas convenable pour les données multi-vues non reliées à des modèles 3D paramétriques. Le présent article présente une nouvelle approche à base de connaissances pour le développement d'un système de résolution de contraintes pour les modèles multi-vues. Une fois qu'un paramètre est modifié, les entités reliées au paramètre dans la vue courante et dans les autres vues (projections géométriques) sont mises à jour.

Un problème de contraintes géométriques est défini par un ensemble d'entités géométriques et un ensemble de relations ou de contraintes géométriques. Les contraintes sont appliquées sur une entité géométrique pour fixer ses degrés de liberté (un point 2D a besoin de 2 coordonnées pour être déterminé). Les systèmes de conception paramétrique à base de connaissance déjà existants se focalisent sur les modèles géométriques 2D qui ne contiennent aucune information concernant les différentes vues sur un modèle. En général, pour résoudre un problème multi-vues (propagation de modification), il faut combiner les informations de la méthode à une seule vue avec les données de projection [ZHONGHUI et coll., 2001]. Les

types de contraintes utilisées dans le modèle multi-vues sont les mêmes utilisées dans le modèle à une seule vue de Wang; Les contraintes explicites et implicites. Les contraintes explicites réfèrent aux dimensions telles que définies par le concepteur, par contre les contraintes implicites réfèrent aux relations géométriques et topologiques tels que les tangentes ou le parallélisme puisqu'ils sont obtenus par auto-détection des positions géométriques initiales dans le modèle. Les règles dans les problèmes multi-vues se composent de règles de méta-niveau et de règles de raisonnement. Les règles de méta-niveau (méta-level) sont des règles générales qui ne considèrent pas les contraintes détaillées, elles fournissent une connaissance générale pour l'inférence et aident à la bonne sélection des règles de raisonnement. Les règles de raisonnement sont des règles indépendantes qui s'appliquent dans des cas spécifiques. Les deux règles méta-niveau utilisées dans le modèle à une seule vue sont valables pour le modèle à multi-vues, soit la règle CMM (the criterion of minimal modification) et la règle CCW (the criterion of constraint weight).

Les trois vues abordées dans cet article sont; la vue de face (FV), la vue de côté (SV) et la vue d'en haut (TV). Un modèle multi-vues réfère à un modèle géométrique 2D, il utilise le principe de ligne de projection pour propager les modifications faites sur un modèle 2D dans la vue source (FV par exemple) vers les vues cibles (SV, TV) selon la prémisse suivante :« Si une entité dans l'une des trois vues (FV, SV, TV) est modifiée, les entités affectées par les lignes de projections de cette dernière dans les autres vues doivent être mise à jour pour qu'elles soient cohérentes avec les nouvelles lignes de projection de l'entité modifiée ». Cette prémisse constitue une règle de type méta-niveau. Un projecteur est un algorithme qui génère les lignes de projections d'une entité et cherche les entités dans les vues cibles qui coïncident avec ces lignes et qui peuvent être affectées par une modification. Les règles de raisonnement sont composées de toutes les règles de raisonnement du modèle à une seule vue ainsi que les règles de projection.

Toutes les contraintes d'un modèle géométrique constituent une base de faits. Cette base est utilisée pour inférer d'autres contraintes géométriques. Selon la théorie des graphes, les contraintes et les entités constituent un graphe où les entités sont les nœuds et les contraintes sont les arcs. Puisque les modifications faites sont généralement des modifications de

dimension, le graphe de dimensions joue un très grand rôle dans la résolution des contraintes. Le CCW est utilisé pour formuler les arbres de dimensions dans le modèle. La procédure d'inférence de contraintes pour le modèle multi-vues présenté dans le présent article est résumée ci-dessous :

- 1- Produire tous les arbres de dimensions d'un modèle en utilisant le CCW. Sélectionner l'arbre dans lequel le nœud racine a le poids le plus élevé (les poids sont affectés selon l'importance de l'entité). La vue de l'arbre courant est considérée comme la vue courante.
- 2- Obtenir l'entité représentée par le nœud racine de l'arbre choisi, si l'arbre choisi est le premier et si le nœud racine est un point, considérer le comme un point fixe sinon choisir un point de l'entité et le considérer comme un point fixe.
- 3- Obtenir un nœud connecté au nœud courant et inférer les contraintes vers le nouveau nœud, localiser les entités (dans les autres vues) qui sont sur les lignes de projection de la nouvelle entité inférée. Propager les contraintes vers ces entités. Répéter l'étape 3 jusqu'à ce que tous les nœuds soient visités.
- 4- Sélectionner le prochain arbre de dimension et répéter 2 et 3. Quand tous les arbres de dimensions sont visités, l'inférence est terminée.

Plusieurs efforts ont été déployés pour améliorer la conception par paramètres. Il existe trois approches pour résoudre le problème de propagation des contraintes géométriques; l'approche numérique, l'approche constructive et l'approche à base de connaissances. Dans l'approche numérique, les contraintes sont traduites sous forme de contraintes numériques où les coordonnées des points sont variables. L'approche constructive est basée sur le fait que la plupart des configurations dans les dessins d'ingénierie sont solvables en utilisant des règles de constructions se basant sur les dimensions du dessin, si l'un des paramètres du dessin est modifié le dessin est régénéré. Dans l'approche à base de connaissance, les contraintes sont représentées comme des prédicats, la résolution des contraintes passe par deux phases, une phase de planification où les contraintes sont analysées et la procédure d'inférence est appliquée afin de créer un plan de construction pour la propagation des modifications, et une phase d'exécution où le plan de construction est exécuté afin de propager les modifications et

de générer le nouveau modèle. Toutes ces approches sont limitées aux modèles à une seule vue, l'auteur propose dans son travail de traiter les modèles multi-vues. L'approche présentée est limitée aux modèles non complexes (ligne, cercle, arcs). L'auteur affirme qu'il devrait définir plus de types d'entités et de contraintes pour traiter des problèmes multi-vues plus complexes.

Le travail réalisé par Zhonghui Wang et al. [ZHONGHUI et coll., 2001] s'avère très rigoureux et très fructueux. Cependant, l'application de ce genre de modèle aux systèmes CAO actuel semble être un projet très ambitieux. Premièrement, le système développé ne traite que des formes géométriques très simples. Deuxièmement, il se focalise essentiellement sur la géométrie, alors qu'un modèle CAO dispose de plusieurs vues dans différents métiers qui ne font pas forcément parti de la géométrie. En plus, le nombre d'arbres de dimensions à définir dans le cas d'un modèle CAO assez complexe est énorme. D'un autre côté, il faudrait absolument prévoir une analyse de performance pour voir si le moteur d'inférence développé pourrait supporter un nombre très grand de contraintes (dans le cas d'un produit très complexe) et s'assurer que l'inférence est correctement faite.

### **II.3.2 Conclusion**

Tous les outils et les concepts que nous avons présentés dans cette section aident à informer les bonnes personnes au bon moment du changement fait sur un document de conception. Ils permettent aussi de propager la modification vers d'autres disciplines. Cependant, ils n'assistent pas vraiment les décideurs à prendre des décisions vis-à-vis du changement. En plus, il existe peu de travaux traitant de la négociation du changement entre les différentes disciplines au sein d'une organisation. Dans la section suivante, nous allons présenter les travaux faits sur la gestion du changement en utilisant la technologie « agent logiciel » pour voir si les systèmes de gestion de propagation externe et de négociation du changement proposé dans ces travaux réussissent à offrir une vraie assistance aux décideurs.

## II.4. L'intégration des agents logiciels aux systèmes CAO pour la gestion du changement

Il y a une multitude de projets qui intègrent la technologie agent à la conception assistée par ordinateur afin d'assurer la collaboration et la coordination lors de la conception, et ce, dans un cadre d'ingénierie simultanée. Nous allons présenter quelques-uns de ces projets afin de montrer leur apport au domaine de la conception mécanique coopérative.

### II.4.1. PACT ou PACE (Palo Alto Collaborative Testbed/Environment)

PACT est un projet supervisé par un groupe de recherche de l'université Stanford, Lockheed, Hewlett-Packard et Enterprise Integration Technologie. L'objectif de ce projet est d'explorer une nouvelle méthodologie pour résoudre des problèmes d'ingénierie en se basant sur le partage de la connaissance et sur la collaboration. Nous présentons dans la figure 2.12 le schéma de l'architecture PACT.

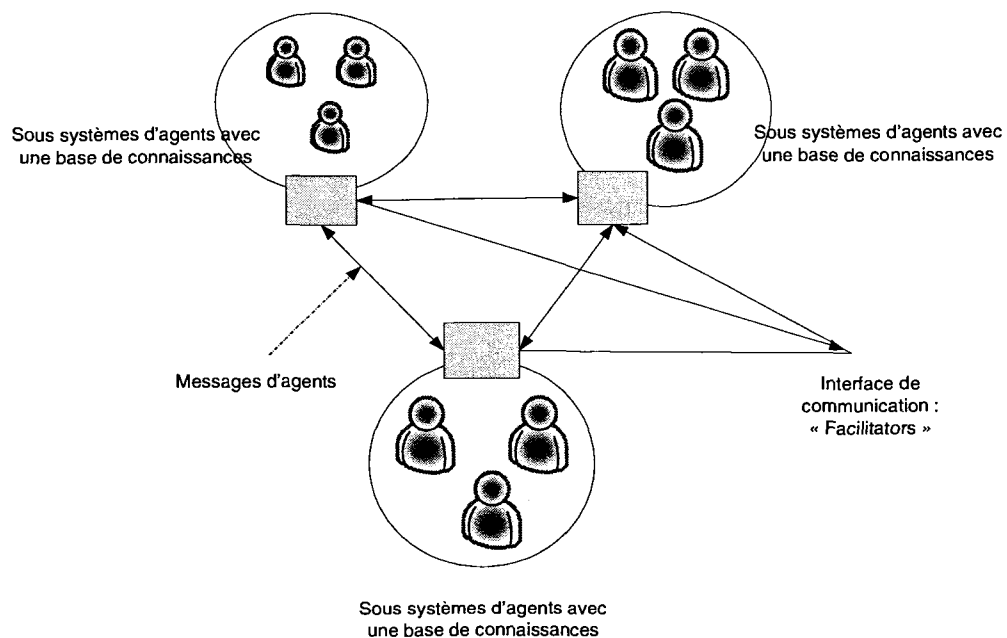


Figure 2.12 : Architecture PACT [CUTKOSKY et coll., 1993] [SHEN, 2002]



Les différents groupes d'agents communiquent entre eux via les interfaces de communications ou «facilitators» qui traduisent les différentes requêtes en un langage standard (ACL). L'architecture PACT a été conçue pour être intégrée dans un système ouvert qui supporte différents systèmes d'exploitation et formats de connaissances.

Le groupe de recherche a réalisé des expérimentations sur un robot manipulateur. Chaque sous-système dans l'architecture PACT modélise un aspect différent du robot des points de vue mécaniques, électronique, logiciel). Les différents sous-systèmes coopèrent entre eux via des langages et des services pour synchroniser les modifications effectuées sur le design du robot [SHEN et coll., 2000].

#### **II.4.2. DIDE (Distributed Intelligent Design Environment)**

DIDE est une architecture basée aussi sur la technologie agent; elle est utilisée pour développer des systèmes de CAO distribués. L'objectif essentiel de ce projet est d'intégrer les outils d'ingénierie tels que les systèmes de CAO et de FAO (Fabrication assistée par ordinateur), les bases de données et les systèmes à base de connaissances dans un système complètement ouvert et dédié à l'ingénierie de conception tout en permettant une certaine intelligence lors de la fabrication [SHEN et coll., 1995] .

Les agents dans ce système échangent les informations entre eux via Internet et n'utilisent pas de médiateurs ou de «facilitators» comme dans l'architecture PACT (figure 2.13). Tous les agents sont autonomes, indépendants et communiquent entre eux directement via le protocole [OSACA, 1996] avec cinq types de messages : *Request*, *Inform*, *Notice*, *Announce* et *Bid* [SHEN et coll., 1998].

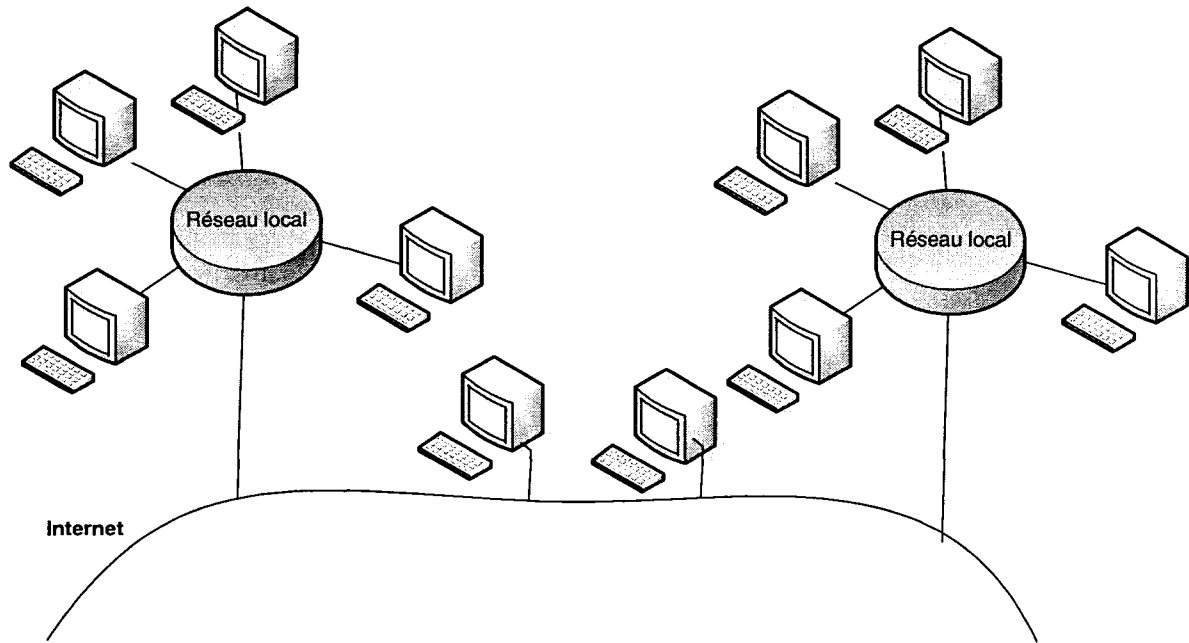


Figure 2.13 : Architecture générale de DIDE [SHEN et coll., 1996] [SHEN et coll., 2000]

Dans l'architecture DIDE, chaque groupe est contrôlé par un « gestionnaire d'agents ». Chaque concepteur au sein du groupe contrôle un certain nombre d'agents et utilise une interface qui est vue comme un agent dans le système. Chaque groupe de concepteurs peut modifier un modèle localement et les résultats sont gardés sous plusieurs versions et c'est le « gestionnaire de produits » qui s'occupe de la synchronisation de toutes les versions existantes pour ne garder à la fin qu'une seule version finale.

Tous les agents dans ce système sont connectés au réseau. Quand un agent veut échanger de l'information avec un autre agent, il envoie son message en « émission » et c'est l'agent qui arrive à interpréter le message qui devient le destinataire de celui-ci. Une fois qu'il reçoit le message, il commence automatiquement le travail puisque les tâches qui lui sont affectées par les autres agents sont prioritaires par rapport à son travail courant.

### II.4.3. SiFA (Single Function agent)

Cette architecture vise à explorer les différentes formes d'interactions, de communication et de résolution de conflits. Dans SiFA [BROWN 1996], le processus de conception est divisé en plusieurs tâches et chaque fonctionnalité est assignée à un agent. Chaque agent dans SiFA a

une seule fonction qui s'applique sur une cible dans un certain point de vue.

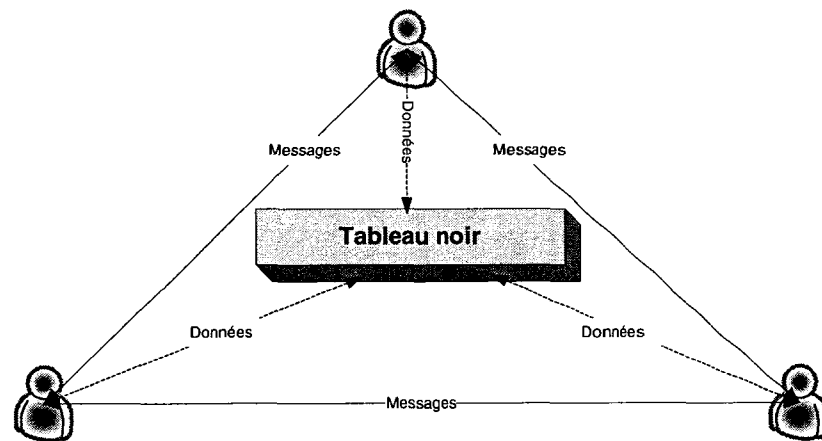


Figure 2.14 : Architecture de SIFA [BROWN 1996]

Dans ce système on distingue plusieurs types d'agents. On présente dans ce qui suit, quelques-uns de ces types :

- Le « sélecteur » : Qui se contente de choisir une valeur parmi d'autres.
- L'estimateur : Qui donne une valeur approximative à partir des valeurs des attributs d'un modèle.
- L'évaluateur : Qui utilise les données du modèle ou du produit pour voir pour quelles limites le « design » rencontre encore les objectifs fixés initialement.
- Le «suggesteur» : Qui suggère des solutions alternatives pour arriver aux buts fixés au départ.

Dans le système SiFA, l'exécution des tâches est contrôlée par un agenda. Il y a une certaine planification qui donne l'ordre d'exécution des agents. Les informations concernant l'état courant du « design » sont stockées dans un « tableau noir » (la définition des systèmes à tableau noir sera donnée un peu plus loin), ce qui les rend accessibles à tous les agents et permet à ces derniers de mettre à jour l'état du « design » en cas de modifications par les concepteurs. Les agents communiquent aussi entre eux directement et c'est ce qui distingue le

système SiFA des systèmes « tableau noir » classiques (voir la figure 2.14).

#### II.4.4. SHARE

SHARE est une architecture distribuée qui permet aux agents de communiquer entre eux pour résoudre les problèmes d'ingénierie, les objectifs de ce projet sont :

- Supporter la communication et la collaboration entre les ingénieurs concepteurs.
- Permettre aux différents outils utilisés par les différents concepteurs de partager les données.

SHARE permet la capture et l'organisation des messages multimédias concernant un produit; il incorpore plusieurs outils comme NOTEMAIL qui permet de créer, visualiser et partager des documents multimédias. Tous les documents sont stockés dans une machine. Les utilisateurs ont des pointeurs sur les différents documents [SHEN et coll., 2000].

Les agents dans ce système communiquent via Internet (figure 2.15), chaque agent représentant l'une des entités suivantes :

- Un concepteur et ses outils CAO personnels.
- Une base de données ou autres informations de services.
- Un service qui supporte le processus d'ingénierie.

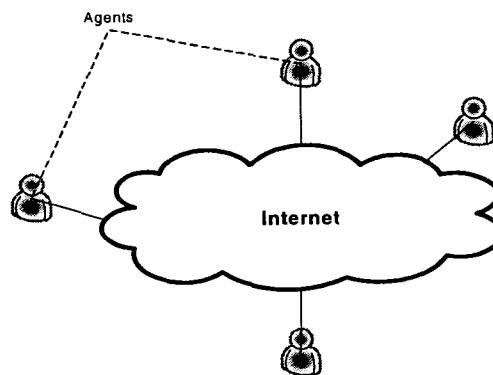


Figure 2.15: Architecture SHARE

#### **II.4.5. Processus distribué de planification et d'allocation de ressources basé sur les agents logiciels**

Les processus de planification de la fabrication et d'allocation de ressources, sont considérés comme étant deux tâches différentes. La planification de la fabrication, détermine comment le produit va être fabriqué. Ce processus organise les tâches de fabrication d'une manière séquentielle pour atteindre certains buts, à savoir, un coût minimal ou un temps de fabrication minimal et pour satisfaire certaines contraintes. D'un autre côté, la réservation des ressources pour la fabrication est le processus d'assigner des ressources à travers le temps à chaque tâche du plan de fabrication. Elle détermine c'est quoi le temps le plus approprié pour exécuter chaque opération en prenant en considération les limitations du système de fabrication en terme de ressources. Le problème de planification de fabrication et d'allocation des ressources est un problème NP-complexe, ce n'est pas du tout évident de lui trouver une solution optimale.

Les agents intelligents ont été utilisés pour développer des systèmes PPAO (Processus de planification assisté par ordinateur). L'approche agent a été reconnue comme étant un moyen efficace qui assure une certaine adaptation et un dynamisme au processus de planification.

IDCPPS [CHAN et coll., 2001] est un exemple de système de planification de processus distribué et coopératif basé sur les agents logiciels. Les tâches de planification sont divisées en trois niveaux; La planification initiale, la prise de décision et la planification détaillée. La planification initiale vise à étudier la faisabilité de la fabrication et génère des alternatives possibles. L'étape de prise de décision, est déclenchée quand l'ordre de fabrication est émis, le résultat de cette étape est une liste alternative presque optimale de plans considérant la disponibilité des ressources pour la fabrication. Le dernier niveau, détermine la sélection finale de machines et d'outils pour la fabrication. Un agent logiciel est assigné à chaque niveau, on trouve aussi dans le système un agent tâche, un agent ressource et un agent de coordination. Malheureusement, aucun prototype n'a été développé pour valider ce concept.

Lim et Zhang [LIM et coll., 2000] ont introduit le système APPSS qui est conçu de plusieurs agents logiciels. Il est utilisé essentiellement pour la reconfiguration dynamique et

l'optimisation des ressources pour la fabrication en temps réel.

Kornienko et al. [KORNIENKO, 2003] considèrent le processus de planification comme un problème de satisfaction de contraintes. Leur système génère un plan distribué optimal satisfaisant toutes les contraintes. Dans ce système, un agent logiciel joue plusieurs rôles, il peut exécuter son programme principal ou d'autres programmes en cas d'urgence. L'agent logiciel décide s'il doit exécuter lui-même un plan d'urgence ou faire appel à un autre agent qui peut gérer la situation.

Parmi les différentes approches d'encapsulation d'agents logiciels définies dans cet article [SHEN et coll., 2006] Weiming Shen et al. définissent deux approches distinctes :

**La décomposition fonctionnelle** : Dans cette approche les agents logiciels sont utilisés pour encapsuler des modules de fonctions, telles la planification, l'allocation de ressources, la gestion du transport du produit ou encore la distribution du produit.

**La décomposition physique** : Dans cette approche, les agents logiciels présentent des entités physiques tel, les travailleurs, les machines, les outils, les produits, les pièces, etc.

Néanmoins, la décomposition fonctionnelle est l'approche la plus utilisée pour intégrer les systèmes dans une organisation (système CAO, base de données, PGI, etc.).

Les protocoles de négociation sont utilisés dans la plupart des systèmes d'allocation de ressources à base d'agents logiciels. Le protocole Contract\_Net et ses versions modifiées est notamment le protocole le plus utilisé. Il existe aussi d'autres protocoles utilisés tel le mécanisme de vote.

#### **II.4.6. Les contraintes de sécurité pour les dispositifs distribués sur le réseau**

La décentralisation des organisations a, de nos jours, un impact considérable sur les systèmes de gestion traditionnels de ces organisations. Les services et les ressources peuvent être distribués un peu partout. L'industrie commence à exiger des systèmes de contrôle assurant une certaine portabilité, interopérabilité, configurabilité. De tels systèmes peuvent offrir

plusieurs avantages [YUEFEI et coll., 2005]:

1- Les tâches et les composants de contrôles peuvent être échangés entre les différents vendeurs.

2- Différents dispositifs peuvent être gérés par des opérateurs se trouvant à l'extérieur de l'organisation.

3- Les différents dispositifs peuvent être reconfigurés à distance pour répondre à des événements qui peuvent survenir à l'improviste.

L'IEC61499 (International Electrotechnical Commissions's function Block) [Function Blocks for Industrial –Process Measurement and Control Systems, 2000] a donné une série d'approches pour la conception de processus industriels distribués et de systèmes de contrôle. Il spécifie une série de modèles de références et couvre le cycle de vie d'un système de contrôle en incluant les phases de planification, de conception, d'implémentation, de validation, d'opération et de maintenance.

Selon L'IEC61499, un système de contrôle distribué est défini comme étant une collection de dispositifs interconnectés et communiquant entre eux. Une fonction exécutée par un système de contrôle est modélisée comme une application et peut être distribuée à travers d'autres dispositifs. L'IEC61499 assure la :

**Portabilité :** La capacité d'échanger des composants ou des codes de contrôle entre différents fournisseurs.

**Configurabilité :** Les dispositifs chez différents vendeurs peuvent être configurés à distance selon les besoins.

**Interopérabilité :** Les dispositifs chez les différents vendeurs peuvent coopérer entre eux.

Il utilise le standard XML pour encoder les commandes de contrôle et les réponses à ces commandes. L'IEC61499 offre donc une architecture tout à fait convenable à un

environnement de contrôle collaboratif, asynchrone et distribué.

Dans le modèle présenté par Weiming Shen et al. dans [SHEN, 2003], il y a trois domaines logiques (figure 2.16) :

Le domaine client : Ce domaine est protégé par un Agent Sécurité AS. Cet agent implémente des politiques d'authentification, de confidentialité et d'intégrité. Le client est une plateforme d'opération qui offre des interactions utilisateurs.

Le domaine dépôt : Ce domaine contient des composants de contrôle exprimé en termes de fonctions IEC61499. Ces composants constituent des fonctions de contrôle différentes qui peuvent appartenir à différents fournisseurs. Ces dépôts peuvent être complètement distribués et appartenant à plusieurs organisations. Chaque dépôt est également contrôlé par un AS.

Le domaine dispositif : Chaque dispositif contient une passerelle sécurisée (PS) et exécute les composants de contrôle.

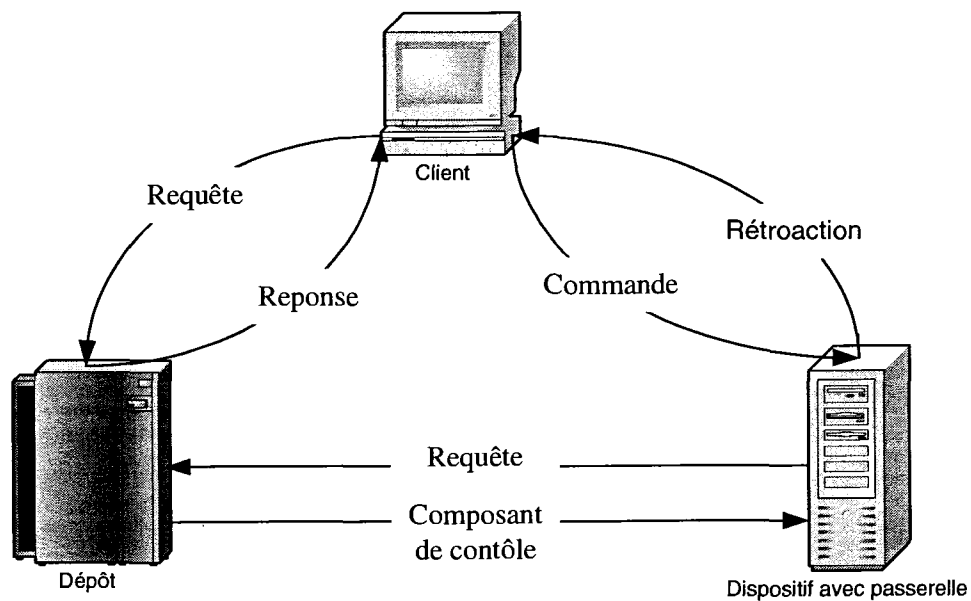


Figure 2.16 : Processus de sécurité typique entre les domaines; client, dépôt et le dispositif

Dans le modèle proposé, chaque dispositif contient une série d'agents qui assure sa sécurité.



AS : Il vérifie l'authenticité et l'intégrité des données et décrypte par la suite le flux de données reçu qui peut contenir des commandes de contrôle en format XML. Pour le flux sortant, l'AS signe et encrypte les données.

L'agent XML-Binder : Il transforme les données XML reçues en objet java exécutable.

L'agent Admission : Il vérifie si les tâches demandées peuvent être exécutées selon sa politique d'admission.

L'agent queue : Il met la tâche demandée dans une queue pour attendre son tour.

L'agent Exécution : Il est responsable de l'exécution de la tâche.

#### **II.4.7. Des agents coopératifs en conception distribuée**

En général, les activités coopératives dans un espace virtuel nécessitent:

- La communication au sein d'un groupe,
- L'organisation et la cohésion des groupes et des activités,
- La distribution et la division de l'information, des applications et des ressources,
- La définition de l'espace distance entre les membres d'une équipe qui coopère, de l'espace temps pour les interactions (l'exécution séquentielle ou parallèle des tâches).

L'objectif de la plateforme PLACID [FOUGÈRES, 2004] (Platform for the innovating and distributed design) est justement d'assister la conception coopérative en satisfaisant les contraintes citées plus haut, les différentes applications d'assistance auxquelles font appel les acteurs du processus de conception sont appelées des micro-outils. Les micro-outils sont des applications légères, faciles à utiliser, qui peuvent être implémentées dans des environnements partagés assez facilement. Ces applications peuvent être connectées ensemble via une base de données. PLACID est basé sur les agents logiciels, il contient quatre types d'agents : un agent application (ou micro-outils), un agent médiateur, un agent système et un agent interface. Les

micro-outils supportés par la plateforme ne font pas parti nécessairement du processus de conception, ils peuvent être considérés comme des services ponctuels auxquels la conception pourrait faire appel.

L'architecture PLACID (figure 2.17) se compose de quatre couches :

Couche 1 : Gestion de l'interface utilisateur (présentation, interactions et contrôle du dialogue) dans un contexte collaboratif, donc multi-utilisateurs

Couche 2 : Gestion des outils d'aide à la co-conception (micro-outils de conception, micro-outils de coopération et éventuellement autres outils de conception)

Couche 3 : Gestion de l'organisation des co-acteurs et des travaux collaboratifs (couche d'agents outils spécialisés dans la coordination de groupe, la gestion des objets partagés, etc.)

Couche 4 : Système d'exploitation et la gestion de la communication à bas niveau

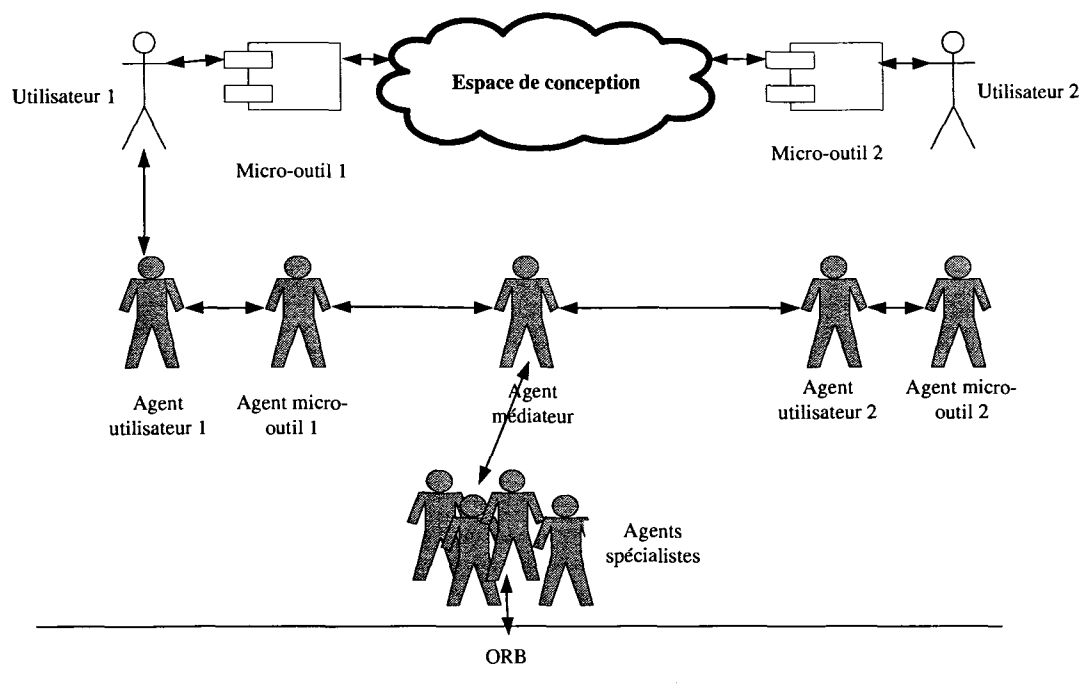


Figure 2.17: La structure Agent de PLACID

Pour tester sa plateforme, Alain-Jérôme Fougères a conçu deux types de micro-outils;

le premier correspond à un meeting-électronique, le deuxième permet d'effectuer une analyse fonctionnelle à un modèle de conception. L'utilisation de ces deux micro-outils se fait dans un environnement collaboratif.

#### **II.4.7. Un système CAO collaboratif basé sur les agents logiciels**

Xinzhong Zhu et al. [XINZHONG et coll., 2006] proposent à leurs tours un système CAO collaboratif basé sur les agents logiciels. Ils présentent un système multi-utilisateurs dans lequel les agents logiciels assurent le partage de l'information et des données. L'interface de l'agent «Concepteur » contient des outils qui supportent les opérations collaboratives ainsi que les opérations individuelles. Cette interface contient les modules suivants :

**L'espace de partage** : Ou le tableau blanc, c'est l'espace de communication entre les membres et c'est là où ils mettent aussi les objets à partager;

**Le pointeur** : Pointe sur le sujet sur lequel le concepteur doit se concentrer le plus.

**Le vidéo temps réel** : Transmet l'image temps réel du concepteur aux autres membres communicants avec lui.

**La voix temps réel** : Transmet la voix en « temps réel » du concepteur aux autres membres communicants avec lui.

**La boîte de dialogue temps réel** : C'est un module de « chat » au cas où la vidéoconférence ne fonctionne pas.

**L'espace des membres individuels** : Le concepteur édite via ce module des fichiers (texte, images, etc.) il peut aussi envoyer les fichiers qu'il désire partager avec les autres membres.

**Le service de fichiers** : Offre un système d'accès aux fichiers (FTP, etc.)

**La fonction de contrôle et de coordination** : C'est une interface de contrôle qui gère les membres du groupe communicant.

**Le fichier log** : C'est un module d'archivage qui enregistre toutes les opérations effectuées.

En plus des agents concepteurs, il existe d'autres agents dans le système qui vont assurer le contrôle des communications et le transfert des données entre les différents concepteurs.

#### **II.4.8. D'autres recherches**

Weidong [WEIDOG et coll., 2003] et son équipe proposent une architecture ouverte à base d'agents flexibles pour supporter le design collaboratif distribué ou centralisé. Dans cette architecture, les collaborations synchrones et asynchrones sont permises. Les mécanismes de génération de versions et de la gestion des accès aux modèles sont implémentés pour supporter la communication synchrone et asynchrone. La technologie DCOM est utilisée pour l'implémentation du prototype.

Le système CABoCAD [ROSENMAN et coll., 2001] est un environnement de conception ouvert dans le domaine de la construction. Ils adoptent l'approche orientée composant et utilisent les notions de « composants distribués et magasin d'assemblage » (figure 2.18). Chaque élément de construction est considéré comme un ensemble réutilisable encapsulant un modèle de données et des méthodes. Un mécanisme d'agent se trouve dans chaque composant afin d'en maintenir la cohérence. L'agent composant (CA) est l'élément de base du système. La gestion des CAs distribués se fait à partir d'une interface WEB. Les magasins d'assemblage contiennent plusieurs CAs et sont conçus par des ingénieurs de conception de disciplines différentes. Feijo et Bento [FEIJO et coll., 1998] proposent un nouvel environnement de programmation Xlog+ pour développer des systèmes CAO basés sur des architectures d'agents hybrides dans lesquelles le raisonnement est effectué à l'aide d'une logique de premier ordre. La logique du premier ordre est un ensemble de symboles désignant des variables, un ensemble de symboles désignant des fonctions, un ensemble de symboles désignant des prédicats et des connecteurs logiques et quantificateurs. Tous ces symboles permettent de formuler des énoncés qui formalisent mathématiquement le concept du raisonnement.

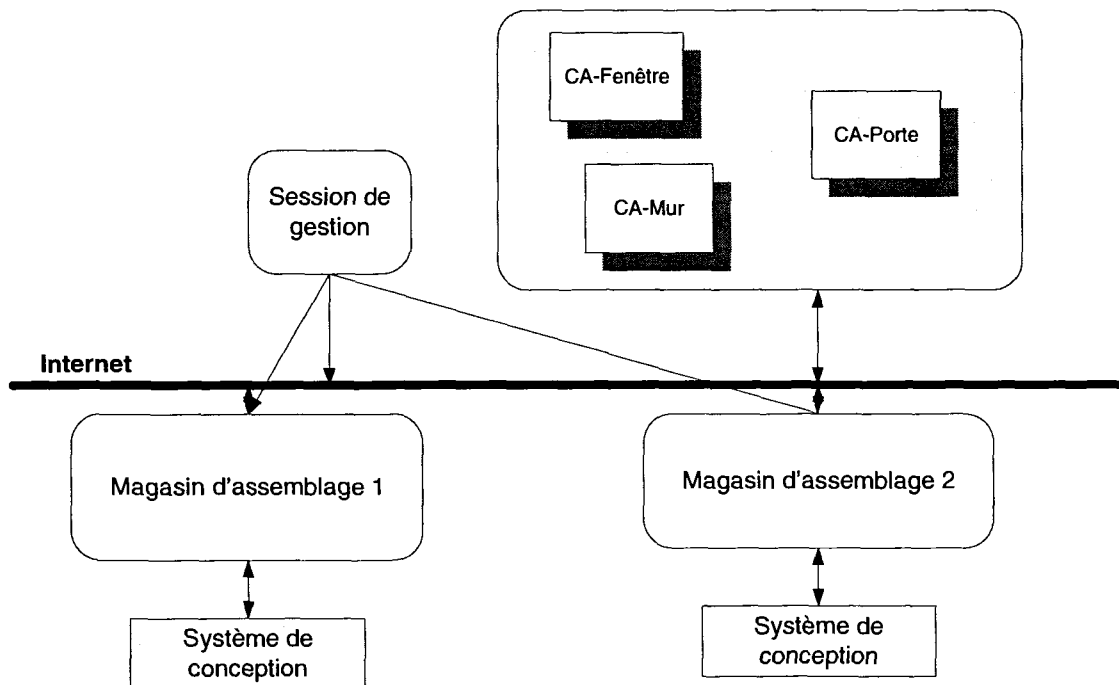


Figure 2.18: Architecture du CaboCAD [ROSENMAN et coll., 2001]

Zha utilise les réseaux de Petri [ZHA, 2002] pour résoudre les problèmes de la conception dans le cadre de l'ingénierie concurrente. Il utilise un modèle de réseau de Petri de haut niveau basé sur l'approche Orienté-Objet [ZHA et coll., 2000] [ANDRAS, 1995] [RUDAS et coll., 1997] comme outil d'intelligence artificielle afin de résoudre les problèmes d'interaction entre les agents de conception. Les éléments essentiels de son système sont : l'humain, des agents experts capables de prendre des décisions ou d'assister les décideurs à prendre des décisions.

Ilan Berker et al [BERKER et coll., 1996] proposent un modèle de résolution de conflit dans un environnement de conception collaborative en utilisant des agents SiFA. Chaque agent SiFA dans l'architecture proposée a sa propre expertise et exécute ses tâches selon son propre point de vue. C'est la différence entre les points de vue des agents qui engendre des conflits.

Des chercheurs de l'Université de Kyoto [YOSHIMURA et coll., 2001] ont suggéré une architecture à base d'agents mobiles qui supporte la négociation entre les différentes disciplines impliquées dans la conception, et ce, afin de trouver des solutions optimales. Le

Le système prend en considération les besoins du client et les solutions proposées par les concepteurs experts des différentes disciplines. L'architecture multi-agents est composée de plusieurs agents experts stationnaires qui vont trouver des solutions optimales, chacun selon son expertise; des agents mobiles vont transmettre les solutions d'un agent expert à un autre pour assurer la négociation et vont évaluer à la fin les solutions trouvées par les agents experts. Si la solution négociée est satisfaisante, l'agent mobile la propose aux experts (humain) sinon il relâche certaines contraintes pas très importantes et relance la négociation. Malheureusement, aucun prototype n'a été développé pour valider l'architecture proposée par les auteurs.

À l'université Stanford [FROST et coll., 1996], une équipe de chercheurs décrit une approche permettant de relier le processus de fabrication au processus de conception pour éviter que les erreurs de conception ne se répercutent sur la production. Les concepteurs doivent prendre en considération les contraintes de production dans les premières étapes de conception pour que les modifications se fassent plus rapidement et plus facilement. Le dialogue entre les deux processus est assuré par des agents logiciels communiquant via Internet. Les agents utilisent le langage KQML [FININ et coll., 1993] pour s'échanger des messages. David Naso et al. [NASO et coll., 2004] utilisent les agents logiciels pour diriger les pièces dans le système de production. Les agents s'occupent du « routage » d'une manière autonome en prenant les décisions eux-mêmes. Les chercheurs ont développé des algorithmes basés sur la logique floue pour implémenter le système d'agents.

#### **II.4.6. Position de notre travail de recherche**

Dans la section II.3, nous avons examiné les outils permettant de faire de la gestion du changement dans le domaine de la conception assistée par ordinateur dans un cadre d'ingénierie concurrente. Cependant, nous avons constaté que ces outils ne font qu'aider les concepteurs et les experts et leur préparer le terrain pour faire eux-mêmes la gestion du changement et la négociation en cas de conflits entre les métiers, donc il n'y a aucune automatisation dans le processus de gestion du changement.

Dans les différentes applications citées dans la section II.4, les systèmes multi-agents sont

utilisés pour assurer le partage des connaissances entre les différentes disciplines d'une organisation. Cependant, le problème de gestion du changement n'est pas cité d'une manière explicite. Il est clair que le partage des connaissances entre différentes disciplines exige une certaine cohérence au sein du système global, ce qui est certainement traité dans ces différents systèmes, mais les problèmes générés lors d'un changement d'ingénierie nécessitent une attention plus particulière. Il est donc essentiel d'examiner ces problèmes de très près et de développer des systèmes de gestion du changement qui nous permettent de résoudre ces problèmes et d'assurer une vraie cohérence au sein des systèmes de conception. En plus, la majorité des systèmes et des architectures proposées sont des solutions complètement intégrées au système CAO utilisé. Cela n'est pas vraiment pratique, puisqu'il faudra refaire tout le travail si l'on décide de changer de plateforme de conception.

Ceci dit, ce que nous proposons dans notre projet de doctorat, c'est un système multi-agents :

- 1- Indépendant du système CAO : Notre système est connecté au système CAO via une interface. Le coût du développement d'une nouvelle interface est négligeable devant le coût du développement d'un nouveau système de gestion des changements.
- 2- Chaque agent de notre système a une certaine expertise qui lui permet de valider ou de rejeter une demande de changement.
- 3- Les agents négocient afin de proposer une solution optimale qui s'approche le plus possible du changement demandé.
- 4- Propagation du changement du système CAO vers notre système (propagation externe)
- 5- Assistance à la prise de décision.

Lors du développement du prototype nous avons pris en considération certaines hypothèses qui ont certainement diminué la complexité des produits traités. Toutefois, le prototype développé valide le concept de l'intégration des agents à la conception. Il pourrait être amélioré par la suite en l'adaptant à la complexité

## **CHAPITRE III**

### **III. Proposition d'une solution d'aide à la gestion du changement à base d'agents logiciels : Revue de littérature**

#### **III.1. Introduction**

L'objectif de notre projet de recherche est la proposition d'un système capable d'automatiser le processus de gestion des changements d'ingénierie dans un système CAO, tout en assurant la cohérence des données issues des différentes disciplines touchées par le changement. Nous présentons dans cette partie les concepts généraux de la solution ainsi que l'architecture à base d'agent proposée. Ensuite, nous démontrons l'originalité de notre architecture, nous présentons un exemple d'application et nous étudions les possibilités d'intégrer notre solution avec d'autres travaux déjà existants. Enfin, nous définissons l'environnement de développement que nous avons choisi.

#### **III.2. Définition des concepts de la solution proposée**

##### **III.2.1. Environnement multi-vues et gestion du changement**

Les projets de conception à grande échelle impliquent différentes disciplines ou points de vue lors de la conception. Chaque discipline a son propre sujet de préoccupation et d'expertise. Lors d'un projet de conception, un produit a une représentation différente ou vue pour chaque discipline.

La vue est l'ensemble des informations qu'un concepteur pourrait extraire de la représentation d'un produit, et ce, selon son propre domaine d'expertise. Un modèle de conception est associé à chaque vue. Il y a avoir des informations communes entre les différentes vues d'un



produit.

Lorsqu'un concepteur, appartenant à une discipline donnée, décide de faire une modification dans un modèle, il faudrait que toutes les vues de ce modèle soient synchronisées après le changement afin d'assurer une certaine cohérence au sein du système. Nous implémenterons une telle fonctionnalité dans le système que nous proposons.

## **III.2.2. Les Agents logiciels**

### **III.2.2.1. Les agents intelligents**

Le domaine des agents intelligents est fortement lié à celui de l'Intelligence Artificielle (IA). En effet, les premiers chercheurs ayant employé le terme «agent intelligent», étaient des spécialistes de l'IA. On parle souvent d'«agents intelligents» par opposition aux «agents mobiles», car ils proviennent de deux domaines différents. Les agents mobiles sont dans la plupart des cas des objets (au sens de la programmation orientée-objet), alors que les agents intelligents ont été introduits en premier par les chercheurs en IA pour désigner des entités logicielles autonomes non forcément objets.

Ici, nous parlerons des agents intelligents au sens de l'IA. Le concept d'agent est à la frontière de nombreuses disciplines : IA (systèmes à base de connaissances, robotique, langage naturel), philosophie, psychologie, génie informatique (logiciel). Les définitions dépendent donc des applications. Nous n'allons donc pas donner une, mais plusieurs définitions.

Un agent intelligent a des perceptions, associées à un raisonnement intelligent, pour aboutir à des actions. Il doit exécuter l'action qui maximise l'efficacité de la réalisation de sa tâche, en fonction de ses perceptions et de ses connaissances du monde. Mais il y a aussi la notion d'autonomie, c'est-à-dire d'action sans l'intervention d'un tiers. Dans le domaine des agents intelligents, on distingue trois grands axes fondamentaux. Premièrement, il y a l'approche planificative qui cherche à répondre à la question «que doit-on faire ?», c'est à dire quelles actions poser et dans quel ordre. Comme un agent est une entité qui agit, on voit qu'il est intéressant de planifier ses actions, dans le but d'arriver à l'autonomie. Deuxièmement, il y a l'approche réactive; Dans cette approche, le comportement intelligent doit émerger de

l'interaction entre divers comportements plus simples. Et finalement, il y a l'approche hybride; elle envisage la possibilité de réunir ces deux approches afin d'obtenir une architecture hybride. On peut ainsi imaginer un agent composé de plusieurs couches comportementales. La couche inférieure serait purement réactive, c'est à dire prenant des décisions basées uniquement sur les informations sensorielles (l'information issue de l'environnement dans lequel l'agent évolue). La couche intermédiaire, ne considérant pas les données brutes, prend en compte des données environnementales pour prendre des décisions. Il existe plusieurs types d'Agents intelligents :

**Agents logiques** : Agents basés sur les connaissances disponibles concernant le monde et un raisonnement (logique) portant sur les actions possibles sur ce monde.

**Agents flous** : Ils basent leurs raisonnements sur la logique floue. La logique floue est une méthode de raisonnement dans l'incertitude. La théorie des ensembles flous permet de spécifier jusqu'à quel point un objet satisfait une description floue. Les ensembles flous prennent une approche différente des autres méthodes logiques; la valeur de la vérité étant un nombre compris entre 0 et 1, non pas juste vrai ou faux (0 ou 1).

**Agents neuronaux** : Ils utilisent la technique des réseaux de neurones (branche de l'intelligence artificielle).

**Agents génétiques** : Ils se basent sur les algorithmes génétiques (simulation de la sélection naturelle). La sélection naturelle, c'est la mort des faibles et la survie des forts.

### III.2.2.2. Les agents mobiles

Le concept d'agent mobile a vu le jour après un examen critique de la façon dont les ordinateurs communiquent depuis la fin des années 70. Face à la croissance exponentielle de ses utilisateurs, une nouvelle approche était nécessaire pour répondre à leurs besoins. Comparons l'approche client/serveur, et l'approche par agents mobiles :

**Approche Client/serveur** : En utilisant l'appel de procédure à distance (RPC) d'un client vers un serveur pour, par exemple, effacer un certain nombre de fichiers d'un certain âge, il faut un

appel pour dresser une liste des fichiers que l'on veut effacer et d'autres appels pour les effacer. Chaque appel engendre une requête et une réponse. Le nombre d'appels pour effacer les fichiers dépend du nombre de fichiers, car c'est l'utilisateur qui se charge de donner l'instruction au serveur de les effacer. On peut aisément constater que cette manière de faire demande du temps à l'utilisateur et charge le réseau (figure 3.1).

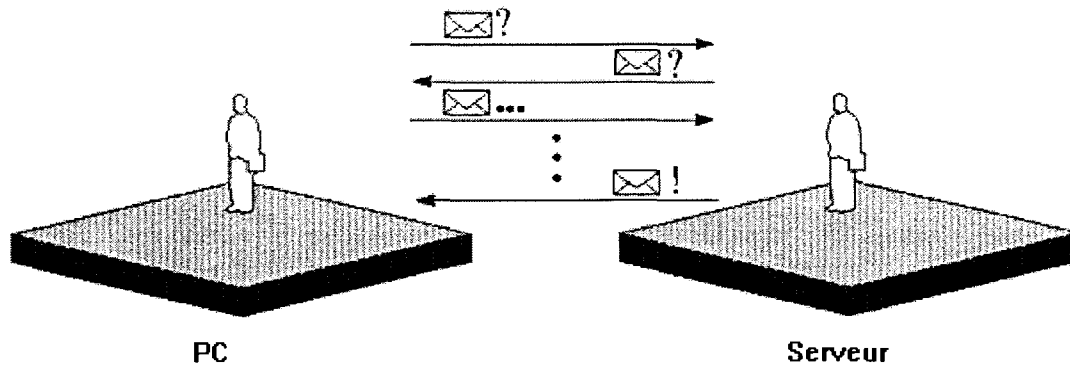


Figure 3.1 : l'approche client/serveur

**L'approche Agent** : Cette nouvelle approche correspond à une programmation à distance (RP). Dans ce cas, chaque message reçu par le serveur consiste en un programme qui s'exécutera sur le serveur. Ce programme est un agent mobile, donc une entité logicielle qui va agir au nom du client. D'une manière autonome, cet agent va chercher au niveau du serveur tous les fichiers d'un certain âge. Il pourra même être doté d'autres délégations telles que la décision du type ou de l'importance du fichier à effacer (figure 3.2).

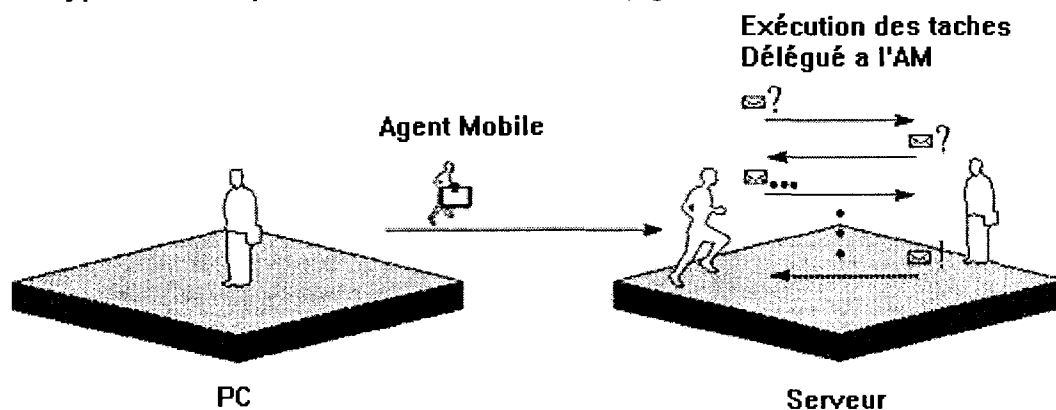


Figure 3.2 : L'approche Agent

Les avantages qu'apporte cette technologie sont tangibles quand on la compare au modèle

client/serveur. La technologie des agents mobiles apporte:

- Réduction de la charge des réseaux : Moindre consommation de ressources réseau, car elle déplace le traitement à l'endroit où se trouve l'information plutôt que de rapatrier l'information vers le traitement.
- Encapsulation des protocoles : Établissement de «canaux» basés sur les protocoles propriétaires.
- Exécution asynchrone et autonome : Les tâches étant embarquées dans les agents mobiles, elles deviennent indépendantes du processus les ayant créées et s'exécutent de façon autonome. Ainsi, il n'est plus nécessaire de maintenir ouverte la connexion entre les systèmes.
- Adaptation dynamique : Les agents mobiles ont la capacité de sonder leur environnement et de réagir aux changements de façon autonome.
- Gain de temps et personnalisation : Si un serveur offre deux services, par exemple, l'un qui liste les fichiers de l'utilisateur et l'autre qui efface les fichiers (par nom), l'utilisateur peut, par le biais d'un agent, ajouter une procédure pour effacer les fichiers d'un certain âge, ou tout simplement ceux qui répondent à un certain nombre de critères. On peut alors dire que le serveur offre des services personnalisés pour cet utilisateur, ce qui permet une extension des services offerts par le serveur.

Étant donné que par définition un agent mobile est une entité logicielle qui fonctionne d'une manière autonome et qui se déplace d'un endroit à un autre à travers un réseau, on peut mettre en évidence un certain nombre de termes caractérisant cette technologie (figure 3.3):

- Places : La technologie des agents modélise un réseau comme une collection de places offrant des services à tout agent qui y entre. Dans la figure ci-dessous, le PC représente une place où subsiste un client alors que le serveur représente une place où nous avons des agents chargés chacun d'un certain nombre de tâches.
- Agents : Les applications qui communiquent entre elles sont modélisées comme une collection d'agents. Chaque agent occupe une place, peut se déplacer d'une place à une autre, et ses propres procédures s'exécutent indépendamment et d'une manière

concourante aux autres agents.

- **Déplacement** : Les agents se déplacent d'un endroit à un autre aussi loin que nécessaire, et peuvent ainsi obtenir un service distant puis rejoindre leurs points de départ. Le voyage qu'entreprend un agent est initié par lui-même, avant ou en cours d'exécution, ainsi, soit il commence son exécution et la continue durant le voyage, soit il s'exécute une fois arrivé à destination.
- **Rencontre** : Deux agents peuvent se rencontrer s'ils se trouvent sur la même place. Cette rencontre permet à un agent d'invoquer un service qui peut être prodigué par un autre agent sur cette même place.
- **Connexions** : Deux agents sont en connexion lorsqu'ils communiquent alors qu'ils se trouvent à des places différentes.
- **L'Autorité** : Le système d'agents permet à un agent ou une place d'attribuer une autorité à un autre agent ou place. Pour contrôler l'accès à ses fichiers, un serveur doit être au courant de l'autorité de n'importe quel agent qui lui demande, par exemple, d'énumérer ou d'effacer des fichiers. On comprend aisément que ce besoin est important pour le réseau.
- **Permissions** : La permission détermine le genre d'opération que peut effectuer un agent à une place donnée, par exemple lire, exécuter, etc.

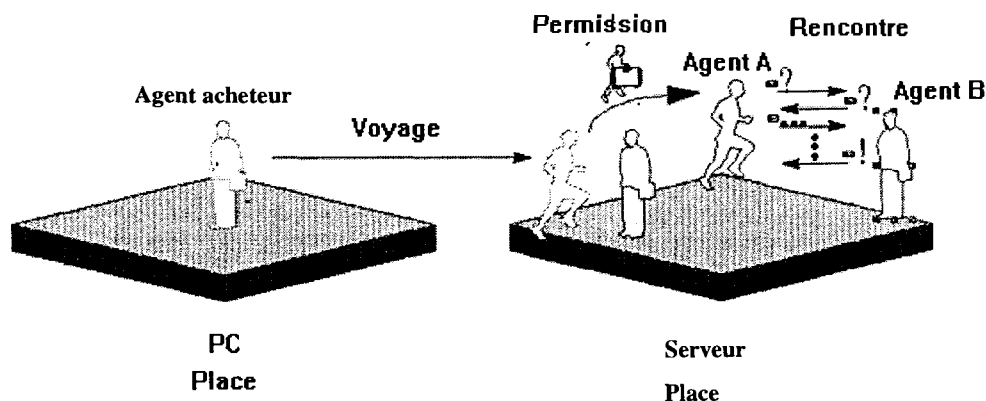


Figure 3.3 : Principes de la mobilité

Un agent mobile est capable d'effectuer les différentes parties des tâches qui lui sont

attribuées dans différents endroits. Il migre d'une machine à l'autre en rassemblant toute l'information dont il a besoin.

On peut distinguer trois types de mobilité :

- La mobilité forte (figure 3.4) : Quand l'agent migre tout en transportant son état d'exécution. L'état d'exécution de l'agent est l'ensemble des informations caractérisant le point de marche ou d'arrêt de son exécution. Après une forte migration, l'agent continue son exécution exactement à partir du point où il s'est arrêté avant sa migration.

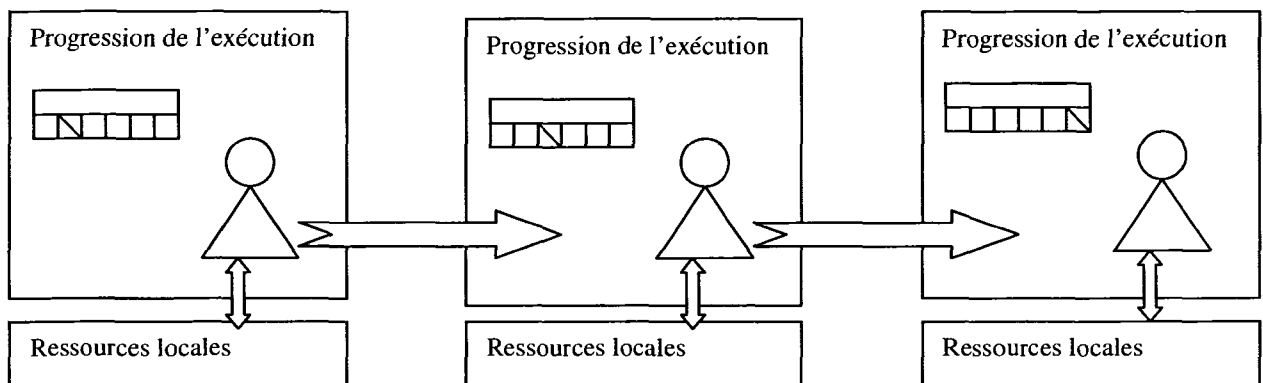


Figure 3.4 : Mobilité forte

- La mobilité faible : L'agent maintient son état de données en se déplaçant d'une machine à l'autre. Son état de données est en fait constitué des valeurs des variables sérialisables (qui peuvent être envoyées via le réseau) dans la machine source. Ces données sont envoyées via le réseau et récupérées par l'agent dans la machine de destination. C'est le programmeur qui décide des variables qui feront partie de l'état de données.
- La mobilité nulle : L'agent est stationnaire, donc il s'exécute uniquement sur la plateforme où il a été créé.

### III.2.3. Les techniques de négociations

La négociation est le processus par lequel un groupe d'agents arrive à une décision mutuelle acceptable sur un sujet donné. La négociation est requise quand les agents ont des objectifs personnels ou sont coopératifs et qu'il pourrait y avoir des conflits entre eux. Elle se fait à travers un ensemble de propositions, des choix, pour éventuellement atteindre une décision commune et satisfaisante. Dans les systèmes multi-agents, l'objectif est de construire des agents négociants en réduisant l'intervention humaine au maximum. La négociation automatique peut être vue comme une recherche dans un espace d'ententes. Les techniques de négociation peuvent être réalisées selon des modèles basés sur des théories de jeux, des approches heuristiques et des approches basées sur l'argumentation. Donc, selon le nombre d'acheteurs et de vendeurs, nous pouvons distinguer quatre cas de figures pour la négociation; un seul acheteur et un seul vendeur, enchère (plusieurs acheteurs et un seul vendeur), enchère inverse (un seul acheteur et plusieurs vendeurs), marché (plusieurs acheteurs et plusieurs vendeurs). Nous définissons dans ce qui suit les trois théories de négociation les plus connues.

#### III.2.3.1. Théorie de la vente aux enchères (VAE)

La théorie de VAE [TUROCY et coll., 2002] présente plusieurs protocoles de négociation. Cette partie résume les protocoles présentés dans le cadre de cette théorie. Dans une session de vente aux enchères il y a un vendeur, plusieurs acheteurs potentiels et un protocole de VAE. Le but du vendeur est d'obtenir le plus haut prix pour son produit, et celui des acheteurs est d'obtenir ce produit au plus bas prix. Nous présentons dans ce qui suit les protocoles VAE existants :

- Enchère anglaise : Dans ce type d'enchère, tous les agents ont connaissance du déroulement de la VAE. En général une mise de départ est fixée, et chacun des acheteurs propose une somme supérieure à celle enregistrée. Cette somme-là devient la nouvelle somme enregistrée, autrement dit les agents acheteurs devraient négocier selon la dernière somme proposée. L'enchère se termine lorsqu'aucun agent ne veut augmenter sa proposition ; elle est donc remportée par le dernier agent qui a fait une

proposition. La *stratégie dominante* de l'agent acheteur est donc de proposer une somme la plus petite possible, qui soit supérieure à celle enregistrée jusqu'à ce que la surenchère atteigne la valeur maximale qu'il peut offrir. Ce protocole de VAE est également appelé *l'enchère publique*.

- Enchère sur enveloppe scellée : Dans ce type d'enchère, aucun agent n'a connaissance de la mise des autres agents. Ainsi, il décide de sa mise en fonction de son estimation de la valeur de l'objet et de son estimation des offres des autres agents. On ne peut pas parler dans ce cas de *stratégie dominante*, puisque l'acheteur n'a pas de vision des mises des autres agents.
- Enchère hollandaise : Dans le protocole de l'enchère hollandaise, le vendeur décide d'un prix maximal, autrement dit il majore les sommes que peuvent proposer les agents acheteurs et décrémente progressivement ce prix jusqu'à ce qu'il atteigne une somme jugée acceptable par l'un des agents acheteurs qui remporte alors l'enchère. Le type de stratégie utilisée ici est donc le même que celui utilisé précédemment puisque les agents n'ont pas connaissance de l'estimation faite par leurs confrères.
- Enchère de Vickrey : Dans ce type d'enchère, aucun agent n'a connaissance de la mise des autres agents. L'agent qui remporte l'enchère est celui qui a proposé la somme la plus élevée; il remporte donc le produit, mais au prix de la seconde mise, c'est à dire celle se trouvant juste au-dessous de sa mise à lui. Ainsi, la stratégie utilisée force l'acheteur à ne pas faire une proposition supérieure à ce qu'il est capable d'offrir, car si jamais il est choisi il est obligé de payer le montant proposé en second et en même temps il ne doit pas proposer un montant trop faible au risque de perdre l'enchère qu'il avait la possibilité de gagner.

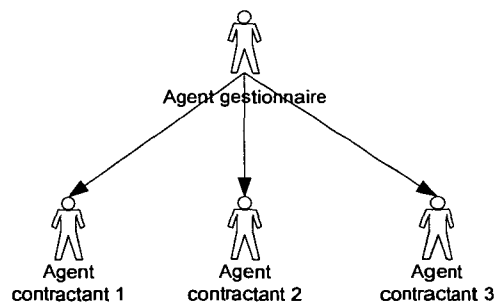
### III.2.3.2. Contract-Net

Un des protocoles les plus étudiés pour la négociation est le protocole du réseau contractuel **Contract-Net** [CHENG et coll., 1996]. C'est une des approches les plus utilisées dans les SMA (Système Multi-Agents). Les agents coordonnent leurs activités grâce à l'établissement de contrats pour atteindre des buts spécifiques. Un agent, agissant comme gestionnaire, décompose son contrat (une tâche ou un problème) en sous-contrats qui pourront être traités

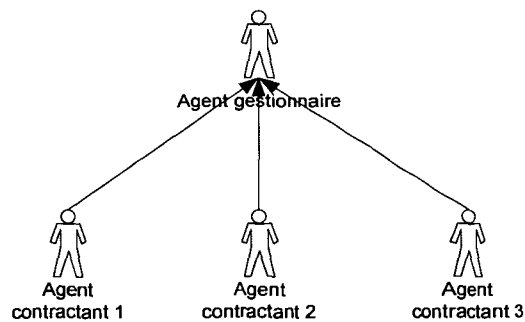


par des agents contractants potentiels. Le gestionnaire annonce chaque sous-contrat sur un réseau d'agents. Les agents reçoivent et évaluent l'annonce. Les agents qui ont les ressources appropriées, l'expertise ou l'information requise envoient au gestionnaire des soumissions qui indiquent leurs capacités à réaliser la tâche annoncée. Le gestionnaire évalue les soumissions et accorde les tâches aux agents les plus appropriés. Ces agents sont appelés des contractants. Enfin, gestionnaires et contractants échangent les informations nécessaires durant l'accomplissement des tâches. (Figure 3.5).

## 1) Annonce de tâche



## 2) Soumissions



## 3) Allocation de la tâche au gagnant

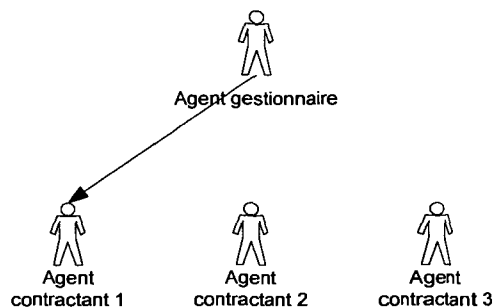


Figure 3.5 : Étapes du protocole Contract Net

### III.2.3.3. Formation et restructuration de coalition

Ce protocole fait apparaître la notion d'optimum de Pareto (un optimum de Pareto est une situation où il n'est pas possible d'améliorer la situation d'un individu sans détériorer celle d'au moins un autre). Cette notion permet d'obtenir, dans le cadre d'une négociation, un état stable où la solution choisie est optimale pour le groupe. Cet état stable est atteint lorsqu'il n'y a plus de possibilités d'améliorer la satisfaction d'un agent qu'au détriment de celle d'un autre agent.

Le protocole se déroule en trois phases distinctes [MARCIA et coll., 2004] :

- Initialisation de la négociation et transfert de tâches : Un agent initie la négociation, il devient l'initiateur en informant les autres agents et en recevant leur accord. Chaque agent doit attendre la fin d'une négociation avant d'en commencer une nouvelle. Après l'initiation, tous les agents se transmettent leurs tâches. Ils peuvent ainsi former des coalitions représentant chacune des tâches ainsi que leurs paramètres associés. Dans chaque coalition on trouve un groupe d'agents qui va s'occuper d'une tâche donnée. L'initiateur calcule alors tous les ensembles de coalitions possibles.
- Négociation : Lorsqu'un agent reçoit un groupe de coalitions, il l'ordonne, selon ses préférences, en groupes d'ensembles homogènes. Il envoie alors, s'il n'est pas le dernier, les groupes de coalitions à un autre agent par ordre de préférence. S'il est le dernier, il choisit le meilleur groupe acceptable parmi ceux présents.
- Transmission de la solution : Une fois l'optimum de Pareto obtenu, le groupe de coalitions choisi - ne pouvant être qu'un optimum de Pareto - est envoyé à tous les agents qui l'acceptent comme résultat de la négociation.

### III.2.3.4. Théorie de jeux

La théorie des jeux est utilisée dès qu'il y a une situation dans laquelle plusieurs personnes – les joueurs – ont à prendre une décision et qu'il existe plusieurs centres d'intérêts et plusieurs centres de décision. Cette théorie est connue dans l'étude des problèmes économiques, politiques, diplomatiques ou militaires. Cette théorie est utilisée dans les protocoles de négociation dans les systèmes multi-agents. Les trois concepts de cette approche et qui sont

utilisés à des fins de négociation, sont :

- La fonction d'utilité : Elle définit le coût des tâches à accomplir pour atteindre un but donné. L'utilité est représentée sous forme de matrices de décisions.
- L'espace des tâches : Dans ce contexte, les tâches sont les actions que les agents entreprennent et qui ont une certaine utilité.
- Le protocole de négociation : Dans la théorie des jeux, une fonction d'utilité est reliée à chaque agent. Par conséquent, chaque agent dans le système entreprend les actions qui vont maximiser sa fonction d'utilité. Le processus de négociation constitue un processus interactif d'offres et de contre-offres dans lequel chaque agent prend la décision qui maximise sa fonction d'utilité.

### **III.2.3.5. Les systèmes à tableau noir**

Métaphore d'un groupe de spécialistes devant un tableau qui attendent une opportunité pour fournir une expertise. La résolution du problème commence lorsque le problème et les données initiales sont sur le tableau. Les caractéristiques essentielles d'un tableau noir sont les suivantes :

- L'indépendance de l'expertise (les spécialistes ou sources de connaissance travaillent indépendamment des autres experts).
- La diversité des techniques de résolution de problème (la représentation interne de chacun des spécialistes est cachée).
- La flexibilité de la représentation de l'information (pas de restriction concernant le contenu éventuel du tableau).
- Le langage d'interaction commun.
- L'activation fondée sur les événements.
- Le besoin de contrôle (un agent externe aux spécialistes dirige la progression de la résolution du problème et gère l'activation des spécialistes).
- La génération de solutions incrémentales.

Werkman [WERKMAN, 1990] propose un système de tableau noir pour supporter la négociation entre agents.

### **III.2.3.6. La négociation basée sur des plans**

Ce protocole est basé sur des stratégies de coopération visant la résolution de conflits existants dans les plans d'exécution des agents. L'algorithme GPGP [DECKER et coll., 2000] (Generalized Partial Global Planning) a été utilisé dans ce sens; il invite les agents à s'échanger leurs plans pour y détecter les conflits, les résoudre et décider par la suite quel agent fera quoi. Le processus de négociation est défini par des états, des messages et des règles de conversation.

### **III.2.3.7. La négociation basée sur l'intelligence artificielle**

Pour que les négociations entre agents soient le plus proche possible de celles entre humains, certains chercheurs ont opté pour les techniques d'intelligence artificielle. Sycara [SYCARA, 1989] [SYCARA, 1991] a utilisé les CBR (Case-based reasoning) et la théorie de l'utilité. Dans cette approche, les actions présentes et futures des agents dépendent des actions passées et de leurs fonctions d'utilité qu'il faut à tout prix maximiser. Un agent peut modifier les croyances, le comportement et les intentions des autres agents en utilisant les techniques de persuasion [SIERRA et coll., 1997]. Sathi [SATHI et coll., 1989] propose une approche qui vise à utiliser la relaxation des contraintes pour trouver une solution qui satisfait tous les agents dans le système.

### **III.2.3. Typage**

Chaque modèle CAO a un type qui lui est associé. Les types de modèles peuvent être définis avant même que les modèles géométriques ne soient créés. Ceci permet de formuler les contraintes sans se lier aux modèles physiques directement. Nous avons définis les types de modèles en utilisant une arborescence XML. Pour qu'une contrainte soit vérifiée correctement pour un type donné, elle doit être vérifiée pour les types fils de ce dernier.

## **CHAPITRE IV**

### **IV. Proposition d'une solution d'aide à la gestion du changement à base d'agents logiciels : Étapes d'implémentation**

#### **IV.1 Définition des paramètres à changer dans le document de conception**

Le système de gestion de changement proposé est composé de trois grands modules qui pourraient être décomposés à leur tour en plusieurs petits modules (voir section IV.2.2). Le premier est divisé en deux parties; la première partie vise à créer les demandes de changements générées automatiquement selon les modifications que le concepteur veut faire sur les pièces; la deuxième partie se trouve à l'extérieur de CATIA et permet de gérer et d'archiver toutes les demandes de modifications faites par tous les concepteurs au sein de l'organisation. Le deuxième module est indépendant du premier module et de l'utilisateur; c'est le système multi-agents, il va vérifier si le changement proposé peut être adopté ou pas. Le troisième module permet de définir les contraintes sur les produits CAO. Ces contraintes vont être utilisées par les agents comme une base de connaissances. Lorsqu'un concepteur désire proposer un changement, il démarre le modèle sur lequel il envisage de travailler, fait ses modifications et démarre par la suite le logiciel de gestion des changements. Ce dernier détecte automatiquement le changement effectué et lance la vérification. (figure 4.1).

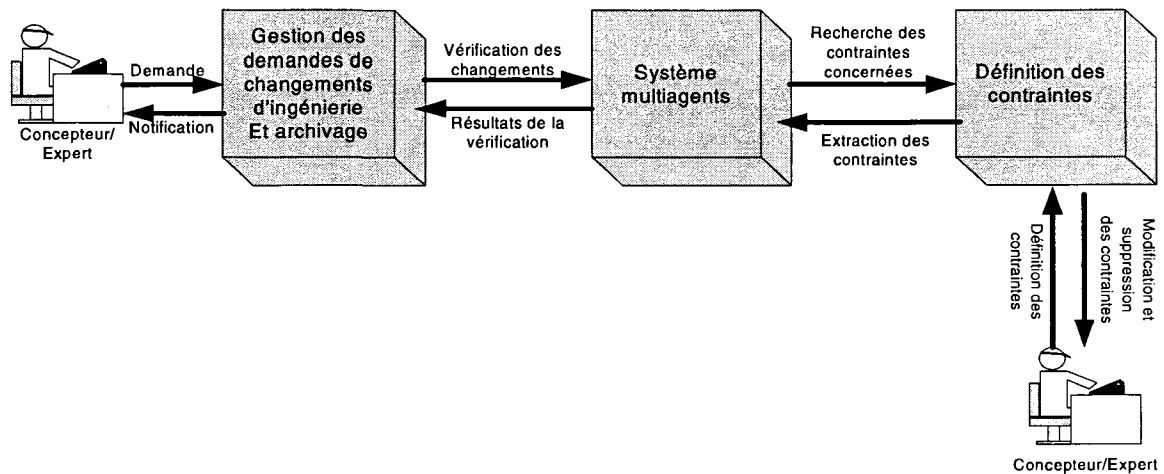


Figure 4.1 : Architecture préliminaire du système proposé

## IV.2 Architecture multi-agents proposée

### IV.2.1. Architecture multi-agents

Nous allons tout d'abord définir quelques termes que nous utilisons dans le schéma de l'architecture du système proposé (figure 4.2 (a)) :

- Agent coordonnateur (AC) : L'agent qui coordonne le travail des agents experts.
- Agent expert (AE) : Un agent expert est un agent qui a une expertise dans un métier donné dans le cadre de la conception mécanique assistée par ordinateur.
- Salle de réunion : Une machine où se réunissent l'agent coordonnateur et les agents experts pour trouver une solution à un conflit.
- Base de connaissances : Une base de données regroupant les spécifications et les contraintes sur les produits CAO.

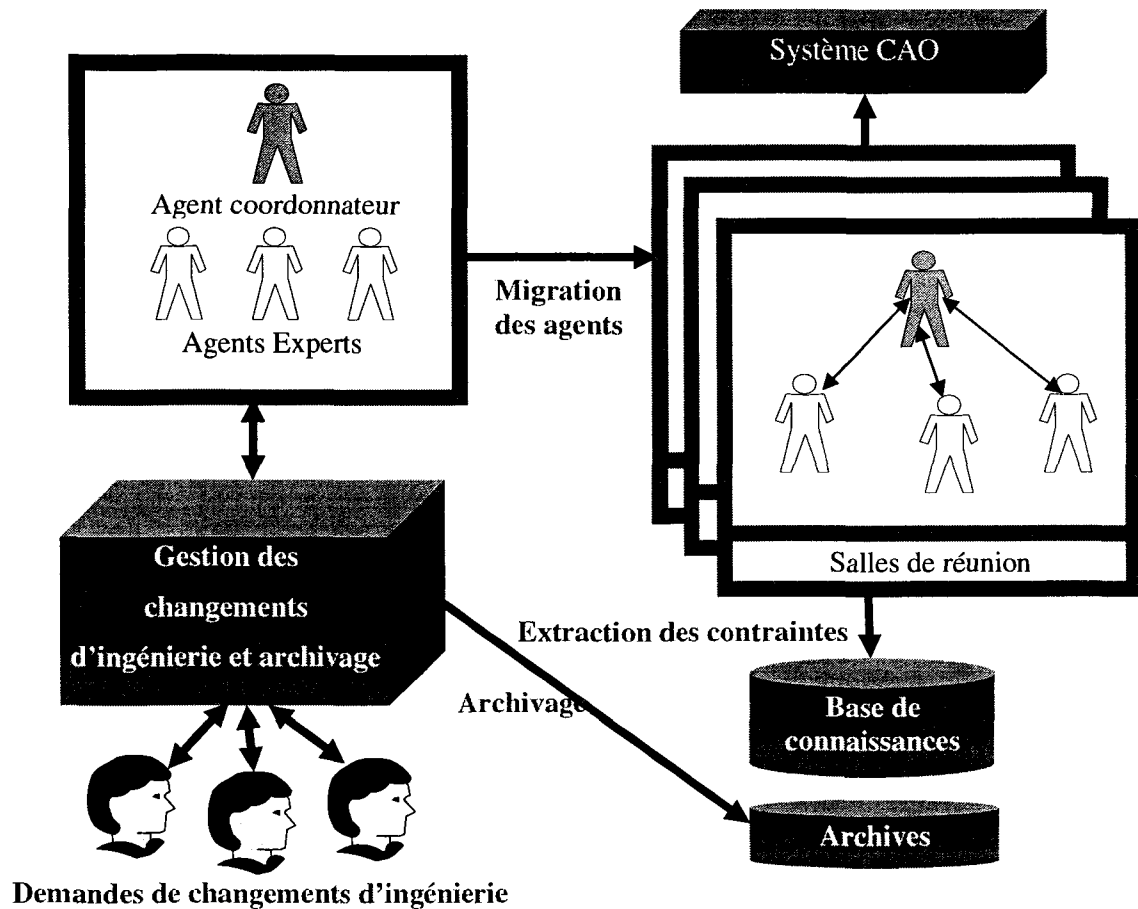


Figure 4.2 : Architecture du système proposé (a)

Lorsqu'un ou plusieurs concepteurs génèrent des demandes de changements, l'application de «Gestion des demandes de changements d'ingénierie et archivage» récupère toutes les demandes de changement et vérifie s'il existe déjà des réponses à ces demandes dans la base d'archivage. Si c'est le cas, elle notifie les utilisateurs concernés, sinon elle démarre le processus de vérification. Elle regroupe ensuite les demandes par projet et trie les ensembles de demandes par priorité. Pour chaque projet, l'application envoie les demandes à l'agent coordonnateur approprié (un agent coordonnateur par projet). Ce dernier trie les demandes reçues par priorité et récupère les vues (les métiers concernés par le modèle CAO) associées à chaque demande ainsi que la valeur du paramètre changé à partir du système CAO (propagation externe du changement). Il crée par la suite un agent expert (AE) pour chaque vue. L'agent coordonnateur ainsi que les agents experts vont se diriger vers une salle de

réunion pour traiter les demandes de changements. Les salles de réunion sont toutes situées dans une même machine assez performante et pouvant supporter les gros calculs lors de la négociation. Nous avons décidé de créer les agents logiciels localement avant de les faire bouger vers la machine destinée à la vérification car nous jugeons que probablement ces derniers auront besoin d'extraire de la machine courante (base de données locale) des informations utiles à la vérification avant de migrer. La mobilité des agents est dans ce cas forte. Dans la réunion, chaque agent expert va recevoir le nom de la pièce modifiée et la valeur du paramètre modifié (changement fait par le concepteur). Chaque AE va charger les contraintes le concernant et ce, à partir de la base de connaissances. Les contraintes sont stockées par projet et par métier dans la base et elles sont considérées, une fois chargées, comme une base de connaissance pour chaque AE. Les AEs vont récupérer à partir du modèle CATIA de la pièce modifiée, les valeurs de tous les paramètres des pièces reliées (liens technologiques) à la pièce modifiée pour faire la vérification.

Si toutes les contraintes sont vérifiées et qu'il n'y a aucun conflit, l'AC envoie un message au concepteur pour valider son changement. Si par contre certains AEs se rendent compte qu'il y a des conflits et que la demande de modification doit être rejetée, l'AC va diriger une négociation entre les différents AEs pour se mettre d'accord sur des paramètres qui s'approchent le plus possible de ce que désire l'utilisateur et qui vérifient toutes les conditions issues des différents AEs. Une fois la négociation terminée, les nouveaux paramètres trouvés par les AEs sont envoyés à l'AC qui les renvoie aux concepteurs. L'AC et les AEs se détruisent par eux-mêmes à la fin du processus de vérification. Le module de gestion des demandes et les agents ACs et AEs peuvent être implémentés sur la même machine. Les salles de réunion se trouvent dans une machine différente. Plusieurs réunions peuvent être déclenchées simultanément. Nous ne pouvons donner le nombre exact de réunions que la machine pourrait supporter qu'après avoir analysé les performances de notre système. Si jamais il y a plusieurs réunions qui sont lancées en même temps et que la machine ne peut plus en supporter d'autres, l'affectation des demandes de changements aux ACs est suspendue jusqu'à ce qu'une salle se libère. Les solutions trouvées sont archivées. Les contraintes sur les produits sont définies par l'utilisateur. Le système dispose d'une section pour gérer les



contraintes définies par l'utilisateur.

Notre système est indépendant du système de CAO utilisé. Même la génération de la demande de changement se fait automatiquement sans l'intervention du concepteur. En effet, quand l'ingénieur de conception démarre la demande de changement à partir du système CAO utilisé, certaines informations pertinentes à la demande sont extraites directement du document de conception. L'information sur les paramètres à changer est propagée par la suite à l'extérieur du système CAO où se trouve toute l'intelligence de notre système.

#### **IV.2.2. Architecture détaillée du système**

Nous présentons dans cette section un peu plus en détail l'architecture présentée ci-haut (figure 4.2 (b)) et ce en utilisant un diagramme modulaire. Avant de présenter l'architecture nous définissons quelques acronymes :

AC : Agent Coordonateur.

AE : Agent Expert.

DC : Demande de changement.

MADC : Module d'affectation des demandes de changements.

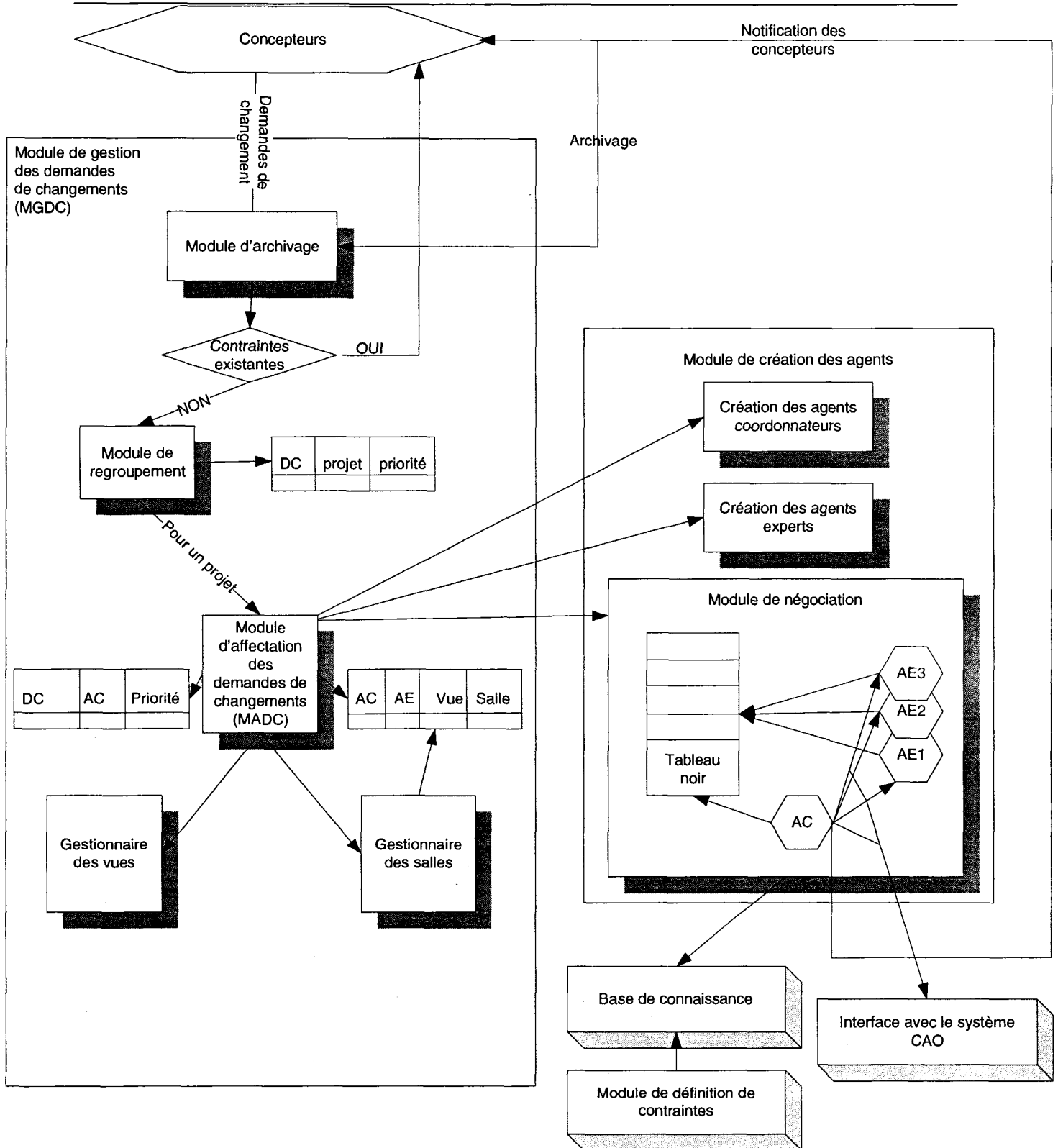


Figure 4.2 : Architecture du système proposé (b)

Une fois les demandes de changement (DC) initiées par les différents concepteurs, le

module d'archivage envoie le résultat de la vérification des contraintes déjà vérifiées et transmet le reste au module de regroupement. Ce dernier regroupe les contraintes reçues par projet et les trie par priorité. Le module d'affectation des demandes de changements crée un AC pour chaque groupe de demandes via le module de création des agents. On se retrouve donc avec un AC pour chaque projet. Chaque AC trie les demandes qui lui sont affectées par priorité et communique avec le gestionnaire des vues pour récupérer les vues pour chaque demande, et pour récupérer la valeur du paramètre changé. Chaque AC communique également avec le gestionnaire des salles pour avoir une salle libre afin de diriger une négociation. Il crée par la suite les AEs correspondants aux vues qu'il a récupérées précédemment. Le module de négociation implémente l'algorithme de négociation entre les AEs sous le contrôle de l'AC. Le module de création des contraintes permet d'alimenter la base de connaissances. Une fois la vérification terminée, chaque AC notifie les concepteurs et les experts concernés et envoie le résultat vers le module d'archivage.

### IV.3. Algorithme de négociation

Nous avons choisi d'implémenter la stratégie de tableau noir comme mécanisme de négociation entre les agents AEs. C'est la stratégie qui s'approche le plus de notre architecture puisque nous avons plusieurs agents experts qui désirent communiquer pour arriver à une solution satisfaisante pour tout le monde. L'agent AC va contrôler la session de négociation (figure 4.3).

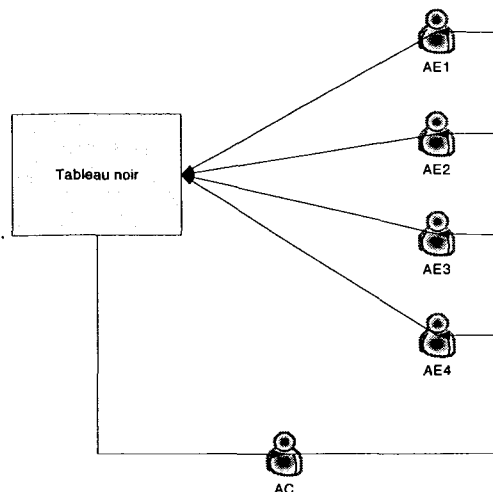


Figure 4.3 : Tableau noir

Une fois que l'AC est créé, il crée à son tour les AEs correspondant aux vues sur la pièce à vérifier. L'AC leur renvoie par la suite la valeur du paramètre modifié spécifié par le concepteur. Si jamais la valeur satisfait aux conditions des différents AEs, il n'y a pas de négociation et la modification proposée par le concepteur est acceptée, sinon les AEs envoient leurs propositions vers le tableau noir. L'AC aide les différents AEs à formuler des solutions satisfaisantes. Si jamais aucune solution satisfaisante n'est trouvée, l'AC garde la modification faite par le concepteur, les différents AEs essaient d'étudier l'impact de cette modification sur d'autres paramètres de la pièce et renvoient une autre proposition aux concepteurs.

La négociation prend fin quand tous les agents sont d'accord sur une proposition ou qu'ils n'arrivent pas à converger vers une solution (dépassement d'un TIMEOUT). Dans le premier cas, ils se mettent d'accord sur les mêmes intervalles de solution et notifient les concepteurs et les experts que la valeur proposée au départ n'est pas acceptée et qu'il y a une autre alternative (nouvel espace de solutions négocié). Dans le deuxième cas, les concepteurs sont invités à faire une autre proposition. Nous présentons dans ce qui suit l'algorithme de négociation.

#### **IV.4. L'originalité de l'architecture proposée**

Nous résumons l'originalité de l'architecture multi-agents que nous proposons dans les points suivants :

- **Gestion du changement assistée** : Notre système permet d'assister les concepteurs à prendre des décisions vis-à-vis des changements qu'ils désirent effectuer sur des modèles. Ainsi, si les changements demandés ne peuvent être effectués, le système propose des solutions alternatives.
- **Indépendance du système de CAO** : L'intelligence de notre système est complètement indépendante du système CAO utilisé.
- **Performance** : L'architecture multi-agents utilisée promet un gain en performance par rapport à l'architecture Client / Serveur classique. Certes, puisque nous utilisons un algorithme de négociation dans notre système; le gain sur le nombre d'instructions qui auraient circulé dans le réseau dans le cas où nous aurions utilisé l'architecture Client/Serveur est énorme. En plus, nous avons prévu de rassembler les

agents négociants dans une même machine pour éviter qu'ils ne se déplacent avec des données et surchargent par conséquent le réseau.

- Robustesse : Les pannes de réseaux n'influencent pas le travail des agents puisque la négociation se fait sur une seule machine.
- Adaptabilité : Notre système pourrait être adapté à tous les domaines de l'ingénierie simultanée dans le cadre de la conception mécanique assistée par ordinateur.

#### IV.5. Exemple d'application

Nous proposons, comme exemple d'application, la gestion du changement de l'épaisseur du mur de la poutre d'attaque de l'aile d'un avion miniature présenté à la figure 4.4. Les trois départements qui pourraient être touchés par ce changement sont : le département structure, le département poids et centrage et le département fabrication. Les deux autres paramètres de l'aile qui peuvent être modifiés sont la longueur et la largeur.

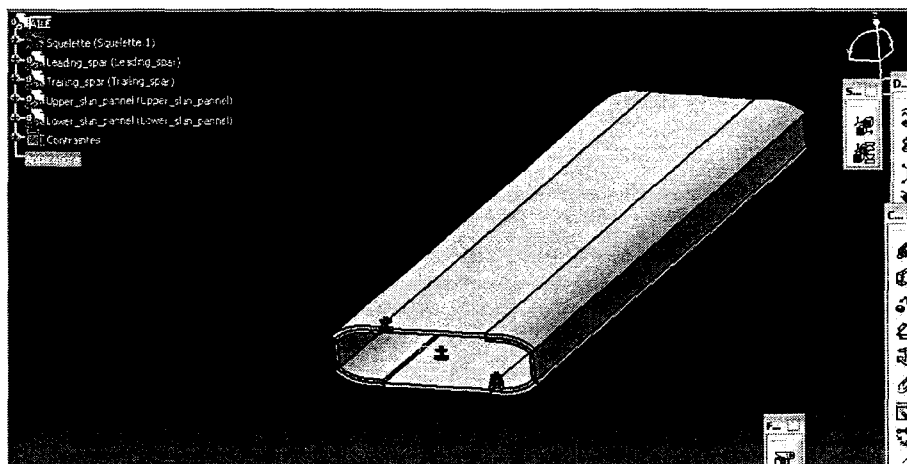


Figure 4.4 : Aile d'un avion miniature

Dès que la proposition de changement de l'épaisseur de la poutre d'attaque est reçue par l'application de gestion des changements d'ingénierie, le module d'archivage l'achemine vers le module de regroupement puisqu'elle n'a jamais été vérifiée. Le module de regroupement va l'acheminer à son tour au module d'affectation des demandes de changement puisqu'il n'y a aucun regroupement à faire (il n'y a qu'une seule demande de changement). Ce dernier va créer un agent coordonnateur (AC) pour la demande. L'AC va récupérer les trois vues

Structure, poids et centrage et fabrication du modèle CAO à modifier, soit à partir du système CAO lui-même, soit à partir d'une base de données de vues que l'utilisateur du système alimentera. Il récupère aussi la valeur du paramètre changé (du système CAO) et il crée par la suite les trois agents suivants et leurs transmet la valeur du paramètre modifié :

AES (Agent Expert Structure) : Fait une analyse par éléments finis pour voir si les contraintes d'élasticité sont respectées ou non; ce type de contrainte fait partie des règles de conception générales de l'entreprise.

AEPC (Agent Expert Poids et Centrage): Cet agent vérifie que la masse de l'aile respecte toujours les spécifications du cahier des charges.

AEF (Agent Expert Fabrication): L'agent fabrication vérifie si, après le changement de l'épaisseur de la poutre d'attaque, le coût de fabrication est toujours respecté.

Tous les agents AEs vont accéder au logiciel de CAO (via Visual Basic) pour récupérer les valeurs des paramètres des contraintes. Ils vont, par la suite, comparer le résultat obtenu avec les spécifications du cahier des charges. Si toutes les contraintes sont vérifiées et validées, les agents autorisent le changement demandé, sinon ils vont négocier tous les trois un nouvel intervalle de solution pour l'épaisseur de la poutre qui va satisfaire aux contraintes des trois disciplines et vont le proposer aux concepteurs. Si jamais ils ne trouvent aucun ensemble de solutions pour l'épaisseur, ils gardent l'épaisseur proposée par le concepteur et proposent des intervalles de solution pour la longueur et la largeur. Ce sont les deux paramètres liés à l'épaisseur que nous avons choisi d'utiliser dans le processus de négociation.

#### **IV.5.1. Algorithme de validation**

Une fois que la demande de changement est initiée (voir Annexes), le système ne la trouve pas dans la base d'archivage. Le module d'archivage l'achemine donc vers le module de regroupement. Ce dernier n'a aucun regroupement à faire puisqu'il s'agit d'une seule demande de changement, il l'achemine donc à son tour vers le module d'affectation des demandes de changements (MADC). Le MADC crée l'AC et récupère les trois vues sur le modèle modifié et qui sont la vue structure (S), la vue poids et centrage (PC) et la vue

fabrication (F). Il crée alors les trois agents experts correspondants; AES (Agent Expert Structure), AEPC (Agent Expert Poids et Centrage) et AF (Agent Fabrication). Les différents agents experts et l'agent coordonnateur migrent vers la salle de réunion qui leur est affectée par le gestionnaire des salles de réunion. Les salles de réunion se trouvent dans une machine performante qui peut supporter les calculs faits lors de la négociation des agents. La salle de réunion est en fait une notion abstraite; le nombre de salles de réunion est exactement le nombre de processus de négociation qui peuvent être déclenchés simultanément sur la machine dédiée aux calculs.

Chaque AE récupère ensuite les contraintes qui lui sont associées à partir de la base de connaissance ainsi que les valeurs des paramètres de ces contraintes à partir du système CAO. Si toutes les contraintes sont vérifiées, l'AC notifie le concepteur et l'expert que la modification peut être faite. Sinon, les AEs négocient sous la direction de l'AC selon trois paramètres, l'épaisseur, la longueur et la largeur de l'aile. Si la négociation converge, l'AC propose l'ensemble de solutions trouvé au concepteur et à l'expert sinon il demande au concepteur de changer le paramètre épaisseur.

#### **IV.6. Performance du système proposé**

Pour mesurer le gain souhaité de notre architecture agents par rapport à une architecture client/serveur (CS) classique il faut prendre en considération la plateforme agent utilisée dans la mesure de performance. En effet, certaines plateformes agent génèrent beaucoup de trafic simplement pour enregistrer les régions, les agences et les agents. Si nous décidons de prendre en considération ce trafic, il faut donc comparer le trafic généré par la plateforme plus la taille de l'ensemble des agents générés par notre système (qui vont se déplacer et générer un trafic qui est proportionnel à leur taille) avec le trafic généré par les requêtes au cas où nous utiliserions l'architecture client serveur.

Si  $\text{Trafic (plateforme)} + \text{Taille (agents)} < \text{Taille (requêtes CS)}$  : L'architecture agent est préférable.

Si  $\text{Trafic (plateforme)} + \text{Taille (agents)} > \text{Taille (requêtes CS)}$  : L'architecture client serveur

est préférable.

Si nous choisissons de ne pas considérer le trafic généré par la plateforme, et si nous commençons notre mesure de performance quand tous nos agents sont créés et enregistrés, nous allons comparer dans ce cas, la taille des agents avec la taille des requêtes.

Si Taille (agents) < Taille (requêtes CS) : L'architecture agent est préférable.

Si Taille (agents) > Taille (requêtes CS) : L'architecture client serveur est préférable.

Nous pouvons suivre le modèle de [HAITAMI et coll., 2005] pour exprimer le trafic dans un réseau :

$$T = \text{trafic}(S_I) = \Gamma_I S_I \quad ; \Gamma_I > 1 \quad (1)$$

$\Gamma_I$  est une fonction appelée *Overhead*.  $(\Gamma_I - 1)S_I$  représente les entêtes ajoutées à l'information  $I$  par la couche réseau et toutes les couches de niveau supérieur du modèle de référence OSI (Open Systems Interconnection).

Soient :

- $S_P$  la taille maximale d'un paquet transmis
- $S_D$  la taille maximale des données d'un paquet transmis
- $S_{CI}$  la taille de l'entête d'un paquet transmis.

On a donc :

$$S_P = S_D + S_{CI}$$

Il est parfois nécessaire d'envoyer l'information  $I$  sous forme de plusieurs paquets.

Dans ce cas, le nombre de paquets est :

- $S_I \div S_D$  paquets de taille  $S_P$  ajouté à cela, 1 paquet de taille  $(S_I \text{ modulo } S_D) + S_{CI}$ , si  $(S_I \text{ modulo } S_D) \neq 0$ .

On obtient alors l'expression de l'*Overhead* suivante :

- Si  $(S_I \text{ modulo } S_D) \neq 0$



$$\Gamma_I = 1 + \frac{(1 + S_I \div S_D)S_{Cl}}{S_I} \quad (2)$$

- Si (SI modulo SD) = 0

$$\Gamma_I = 1 + \frac{(S_I \div S_D)S_{Cl}}{S_I} \quad (3)$$

En prenant en considération (2) et (3) dans l'expression du trafic (1) on obtient :

- Si (SI modulo SD)  $\neq$  0

$$T = \text{trafic}(S_I) = \Gamma_I S_I = S_I + (1 + S_I \div S_D)S_{Cl}$$

- Si (SI modulo SD) = 0

$$T = \text{trafic}(S_I) = \Gamma_I S_I = S_I + (S_I \div S_D)S_{Cl}$$

Nous pouvons utiliser un logiciel de capture de paquet (Ethereal par exemple) pour comparer le trafic, le coût et le délai de notre architecture avec l'architecture Client/Serveur.

## IV.7. Environnement de développement

### IV.7.1. Langage Java

Pour développer le module de gestion des modifications ainsi que le système agents, nous allons utiliser le langage Java qui est parfaitement adapté à n'importe quel type de machine. Cette notion d'adaptabilité doit être reliée à la notion de portabilité assurée par la machine virtuelle Java (dans la mesure où celle-ci existe pour chaque type de machine).

### IV.7.2. Grasshopper

Grasshopper est la première plateforme agent qui a été conforme au standard MASIF [DEJEAN et coll., 1998]. Elle possède un mécanisme de sécurité sophistiqué (protection contre les agents malicieux et protection des agents entre eux) ainsi que des fonctionnalités de gestion assez simples.

### **IV.7.3. Langage Visual Basic**

Le langage VB permet d'accéder aux fonctions de l'API (Application Programming Interface) de CATIA assez facilement via des objets COM.

### **IV.7.4. Système de CAO CATIA**

Nous utilisons comme logiciel de CAO le logiciel CATIA (Conception Assistée Tridimensionnel Interactive Appliquée). C'est un logiciel de conception assistée par ordinateur créé par Dassault Systèmes.

## **IV.8. Conclusion**

Dans les premières sections de ce chapitre, nous avons parlé de la gestion du changement dans un environnement multi-vues et nous avons présenté la technologie agent logiciel que nous avons utilisé dans le cadre de ce projet ainsi que les algorithmes de négociation existant. Dans les dernières sections, nous avons présenté l'architecture multi-agents de notre système, l'algorithme de négociation que nous avons choisi et un exemple d'application. Nous avons aussi présenté une méthode qui pourrait être utilisée pour analyser les performances de notre architecture ainsi que l'environnement de travail dans lequel nous avons développé notre application. Dans les chapitres qui suivent, nous présenterons le prototype réellement implémenté ainsi que les résultats du projet.

## **IV.9. Intégration de la solution proposée avec le système à entités parlantes développé à l'ETS**

Khaled Soufi, diplômé de l'ETS, propose un système de propagation des changements à un niveau de granularité plus fine qu'un fichier [SOUFI et coll., 2006]. Il suggère de représenter tous les documents reliés à la conception (CATPart, Word, Excel...) sous forme d'entités parlantes, et de définir les liens entre ces entités afin de gérer la propagation des changements d'ingénierie. Dès qu'un concepteur établit un changement dans une entité, cette dernière envoie un signal vers un module central de gestion des modifications spécifiant les paramètres

à changer. Celui-ci communique ensuite avec le module de gestion des liens afin de propager les changements demandés vers les autres entités concernées. Les liens entre entités sont définis par l'utilisateur.

Puisque notre système fait aussi de la gestion du changement, et que nos deux travaux font partie du projet IP2CM (Integrated Process Product Change Management) du CRIAQ (Consortium de Recherche et d'Innovation en Aérospatiale au Québec), une intégration des deux travaux doit être envisagée. Nous avons donc décidé d'étudier en profondeur, dans le cadre de la thèse de doctorat, les possibilités d'intégration de notre système avec celui de Khaled Soufi. Nous présentons, dans ce qui suit, les possibilités de complémentarité ou de comparaison entre les deux systèmes.

#### **IV.9.1. Complémentarité**

À première vue, la complémentarité entre les deux travaux semble faisable. Khaled crée des entités qui envoient une notification dès leurs modifications; nous pouvons donc utiliser cette notification comme entrée de notre système de gestion des changements. De cette manière, nous n'aurons pas à développer le module qui permet de détecter les changements faits à l'intérieur du système CAO (figure 4.5).

Dans le système proposé, nous gérons les liens de changement seulement entre les pièces CAO. Par contre, dans le système de Khaled la propagation des changements utilise des liens vers tout type de document intervenant dans le cycle de vie d'un produit, que ce soit des pièces, des documents Word, des images ou des fichiers Excel. Ceci s'avère très intéressant, car à chaque fois qu'une pièce est modifiée, les autres départements – par exemple les départements fabrication et commande, marketing et finances qui utilisent essentiellement des fichiers Word, Excel et des images- sont avisés du changement établi.

Nous pouvons ainsi prévoir intégrer le module de gestion des liens dans notre système. De cette manière, nos agents n'auront pas à créer les liens eux même; ils vont tout simplement utiliser la base de liens déjà faite dans le système d'entités parlantes. Les agents pourraient utiliser aussi les liens vers les documents Word, Excel ou autres puisque notre système gère

déjà les liens entre les pièces CAO dynamiquement. De cette manière, nous pourrions gérer non seulement les changements d'ingénierie entre pièces CAO, mais aussi leurs impacts sur les autres départements de l'entreprise (marketing, finance, fabrication et commandes...) figure 4.5.

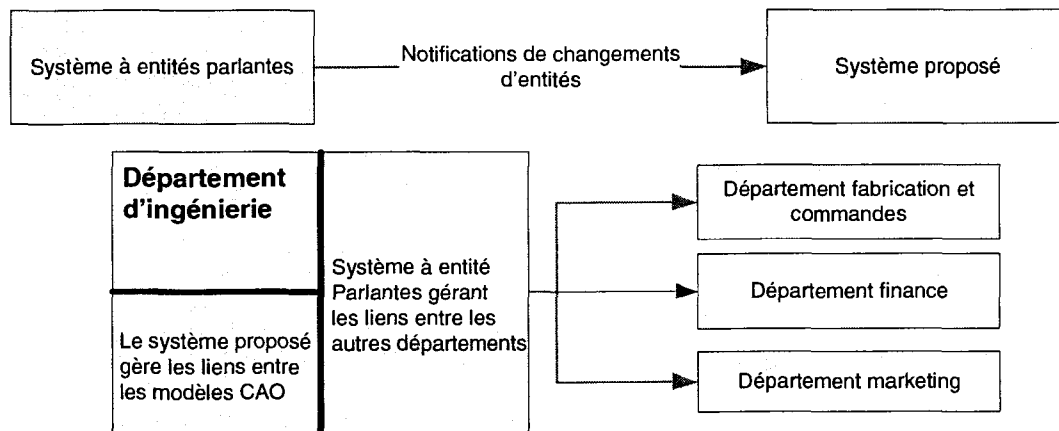


Figure 4.5 : Intégration du système proposé avec le système à entités parlantes

#### IV.9.2. Comparaison

Nous pouvons aussi prévoir dans des travaux futurs faire une comparaison entre les performances des deux systèmes puisqu'ils ont pratiquement le même objectif, à savoir la gestion des changements d'ingénierie. Une évaluation du coût de la création et de la conservation des liens dans les deux systèmes par exemple donnera une bonne idée de leurs performances. Il s'agirait donc, dans le cadre d'un travail futur, d'étudier les limites de chacune des deux approches pour savoir laquelle est la plus réalisable et la plus proche de ce qui est demandé dans l'industrie.

## CHAPITRE V

### V. Le prototype de gestion des changements proposé : MAS-ECM (Multi-Agent System-Engineering Change Management)

#### V.1. Introduction

Nous présentons dans cette section (figure 5.1) les différents modules de notre prototype ainsi que toutes les données nécessaires à son fonctionnement. Nous commençons par la présentation de l'interface d'accueil. Notre prototype se compose de cinq menus principaux :

Arbres (Trees) : Gestion des arbres de typage des modèles CAO.

Attributs : Gestion des attributs des modèles CAO.

Contraintes (Constraints) : Gestion des contraintes sur les modèles CAO.

Vérification (Verification) : Module de vérification des contraintes développé dans le projet de maîtrise [HABHOUBA et coll., 2006]

Gestion des changements (Change management) : Module de gestion des changements.

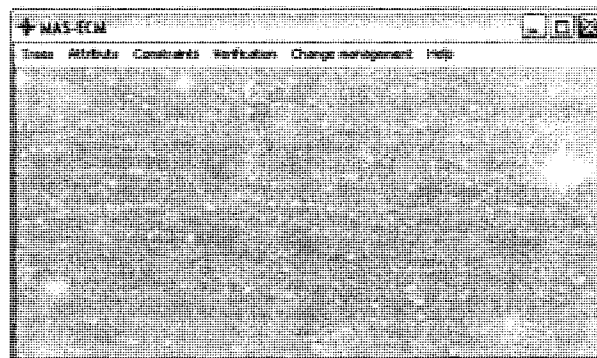


Figure 5.1 : Les différents modules du prototype développé

Nous présentons dans les parties qui suivent; le module sur la gestion des arbres de typage, celui sur la gestion des contraintes et enfin celui sur la gestion des changements d'ingénierie.

## V.2. Les différents modules du prototype

Nous présentons dans cette section les différents modules développés pour le prototype MAS-ECM.

### V.2.1. Le module de gestion des contraintes

#### V.2.1.1. Définition des contraintes

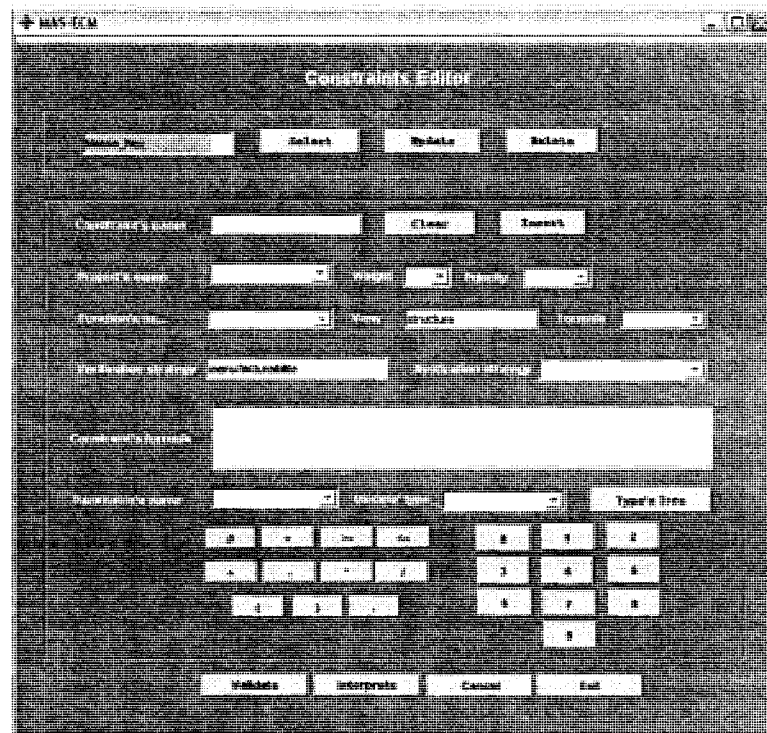


Figure 5.2 : Module de création des contraintes

Nous présentons, dans la figure 5.2, l'interface de création des contraintes. En cliquant sur le bouton « Validate » nous avons la possibilité de compiler la contrainte pour voir si elle est conforme à la syntaxe pré-établie pour les contraintes. Par la suite, nous cliquons sur le bouton « Interprets » afin de l'interpréter pour qu'elle soit comprise par les agents logiciels. En créant une contrainte, on peut lui assigner plusieurs attributs; le projet auquel elle appartient, son

poids, sa priorité, sa formule. Il faut spécifier également de quel type de fonction il s'agit puis la politique de vérification et de notification choisie. Toutes ces données sont utiles surtout pour le module de vérification des contraintes développé lors du projet de maîtrise. Pour le module de gestion des changements, la formule de la contrainte compilée et interprétée (sous formes d'actions que l'agent peut exécuter) suffit pour le bon fonctionnement du module.

## V.2.2. Le module de gestion des types de modèles

Nous avons choisi de typer le modèle CAO utilisé dans le cadre du prototype. Nous utilisons pour cela un arbre de typage XML (exemple en annexe).

### V.2.2.1. Arbre de typage

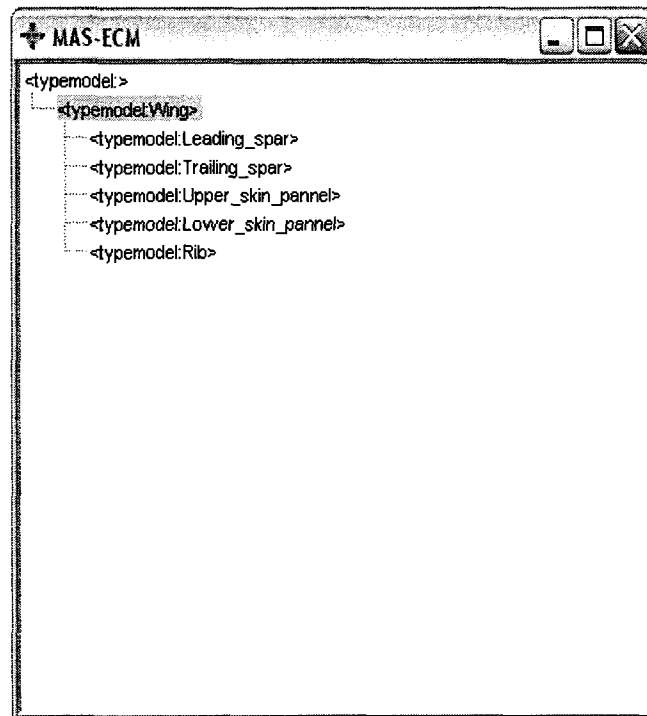


Figure 5.3 : Arbre de typage des modèles CAO

Ce module permet de créer et de modifier les arbres de typage. Toutes les contraintes définies dans le module de gestion des contraintes sont typées; c'est-à-dire qu'on applique une contrainte sur un type et non sur un modèle CAO précis. Donc, pour la vérification de la contrainte, le système a besoin de connaître tous les types « enfants » du type cité dans la

contrainte pour pouvoir prendre en considération les modèles associés. Nous utilisons XML pour stocker la structure de l'arbre de la figure 5.3.

### V.2.3. Le module de gestion des changements d'ingénierie

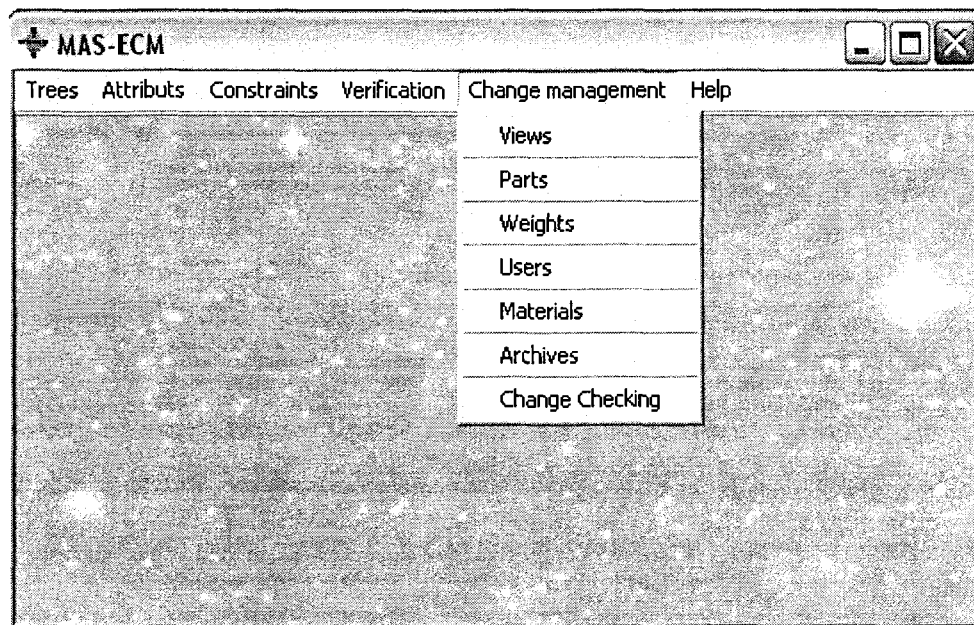


Figure 5.4 : Module de gestion des changements d'ingénierie

Lors du développement de notre prototype, pour des raisons pratiques (problème de licence), nous n'avons pas étudié la possibilité d'intégrer notre système à un système SGDT existant pour avoir toutes les données nécessaires au fonctionnement de la gestion des changements. Nous avons donc développé nos propres modules (figure 5.4) qui nous permettent de gérer et de stocker l'information sur les pièces, les vues, les poids des paramètres, les utilisateurs, les matériaux et les archives.

Ceci dit, il est tout à fait possible d'intégrer notre travail, dans le futur, à un SGDT déjà existant et qui gère toutes ces données. Il est possible aussi de développer des modules au sein du SGDT s'ils devaient être requis.



### V.2.3.1. Gestion des vues (Views)

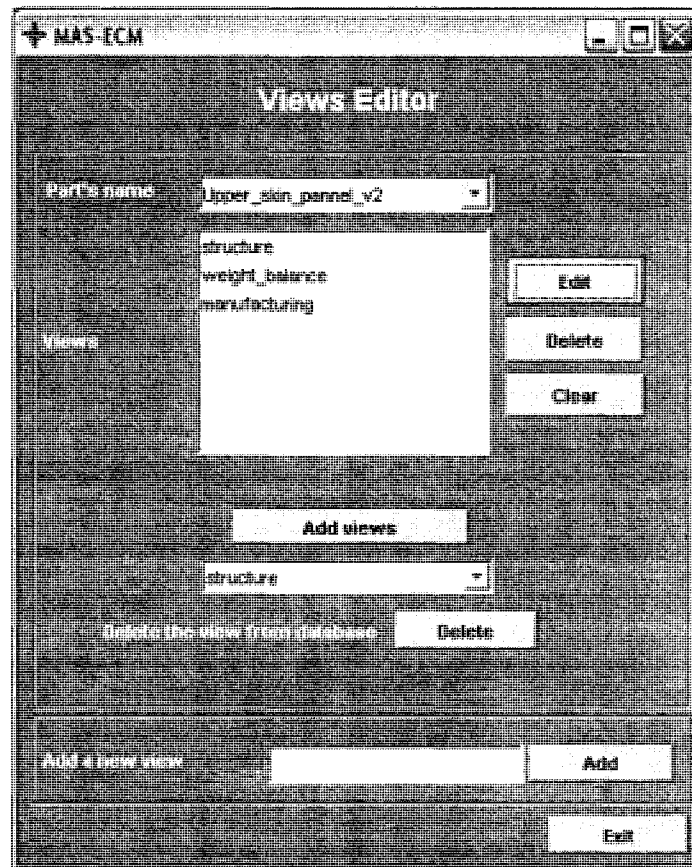


Figure 5.5 : Module de gestion des vues

Le module de gestion des vues (figure 5.5) permet d'assigner des vues à des pièces CAO, il permet aussi de créer, modifier ou supprimer des vues de la base de données. Ce module est très important pour le fonctionnement du module de gestion des changements dans la mesure où il permet à l'agent coordonnateur de savoir quels genres d'agents experts il devrait créer. Il permet aussi de connaître les experts et les concepteurs correspondant à chaque vue ou département pour pouvoir les notifier à la fin de la vérification.

### V.2.3.2. Gestion des pièces (Parts)

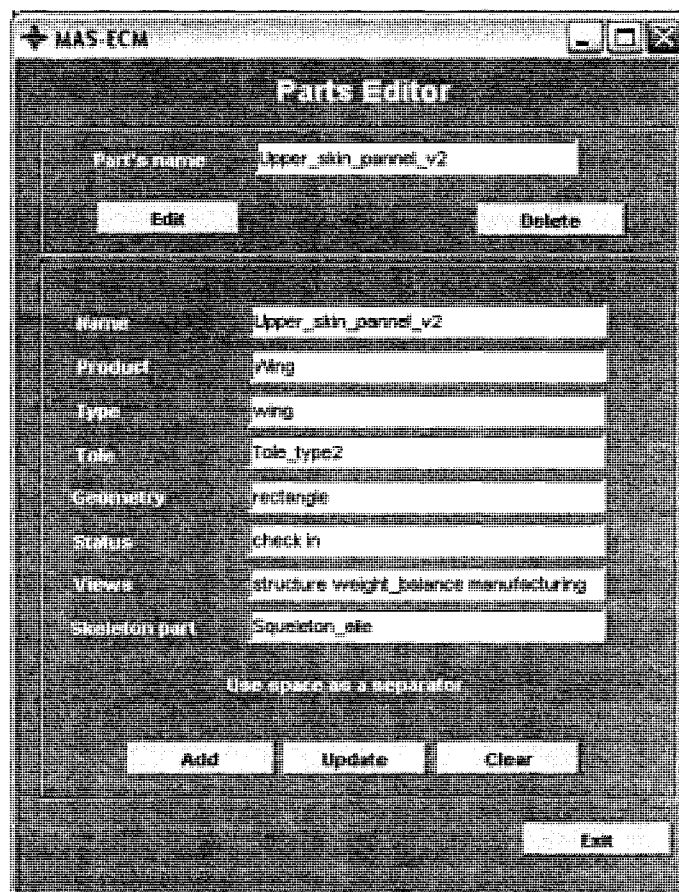


Figure 5.6 : Module de gestion des pièces CAO

Le module de gestion des pièces (figure 5.6) permet de connaître toute l'information nécessaire à la vérification des changements sur un modèle. Cette information comprend, le produit auquel la pièce appartient, le type de la pièce, sa géométrie (pour pouvoir faire des calculs de volume par la suite), son statut, les vues et le squelette du produit auquel appartient la pièce (le squelette permet de propager les modifications faites à travers les pièces de l'assemblage). Toutes ces informations sont nécessaires au fonctionnement des différents agents du système. Nous supposons, par souci de simplification, que chaque produit contient un squelette paramétré qui permet de faire les modifications nécessaires sur certains paramètres communs entre les pièces d'un produit sans pour autant être obligé d'ouvrir chaque pièce pour le faire (par exemple, le changement de la longueur ou de la largeur d'une

aille).

### V.2.3.3. Gestion des poids des paramètres (Weights)

Lors de la vérification d'un changement, notre système a besoin de connaître le poids relatif de certains paramètres pour gérer la négociation.

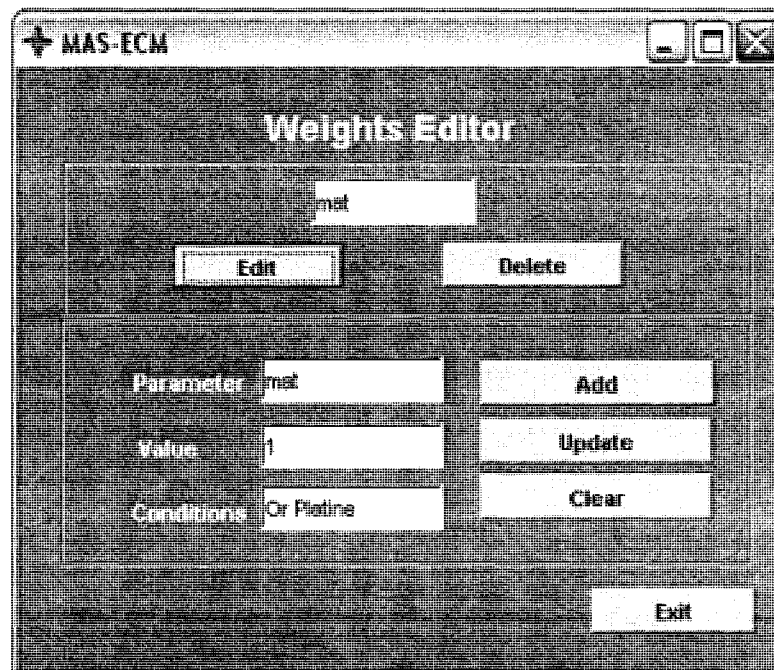


Figure 5.7 : Module de gestion des poids des paramètres

Dans notre exemple, l'agent structure a besoin de connaître le poids du paramètre « mat » (figure 5.7) pour savoir s'il est permis de changer de matériaux afin de trouver un autre espace de solutions satisfaisantes. Il est possible de spécifier des conditions, par exemple, ne pas changer le matériau courant pour Or ou Platine.

#### V.2.3.4. Gestion des utilisateurs (Users)

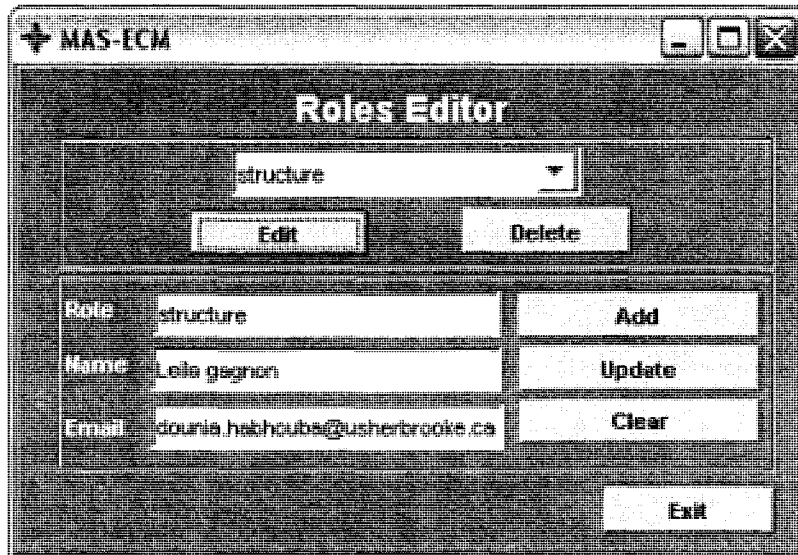


Figure 5.8 : Module de gestion des utilisateurs

Ce module (figure 5.8) permet d'assigner un expert à chaque département pour que l'agent coordonnateur sache à qui transmettre les résultats de la vérification du changement effectué.

#### V.2.3.5. Gestion des matériaux (Materials)

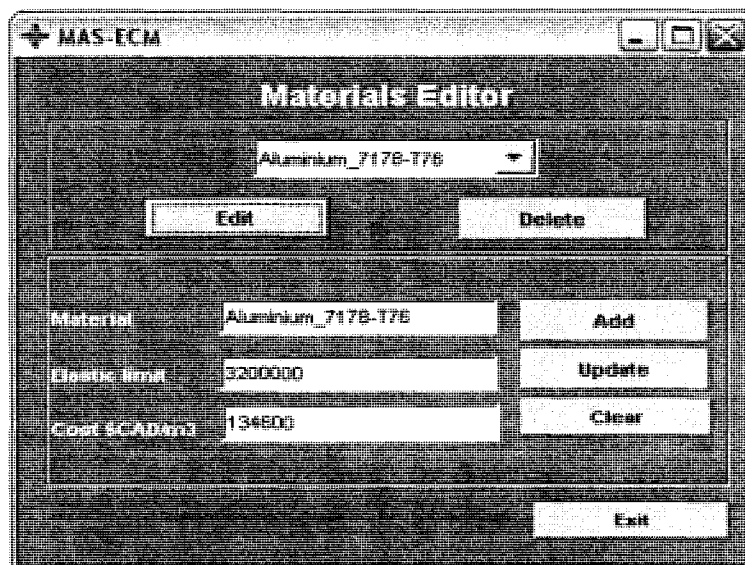


Figure 5.9 : Module de gestion des matériaux

Le module de gestion des matériaux (figure 5.9) est essentiellement utilisé par l'agent

structure (AES) et l'agent fabrication (AEF). Il permet d'assigner un coût et une limite élastique à chaque matériau. L'agent structure utilise la limite élastique pour calculer le facteur de sécurité. L'agent fabrication a besoin du coût pour vérifier sa contrainte.

#### V.2.3.6. Gestion des archives (Archives)

Avant la vérification d'un changement d'ingénierie, le système d'agents a besoin d'aller vérifier dans les archives si la vérification n'a pas déjà été faite. Pour ce faire, le système a besoin de stocker, après chaque vérification, le nom de la pièce concernée, la valeur de l'épaisseur modifiée ainsi que le nom de fichier du résultat (figure 5.10). Si la vérification a déjà été faite, le système renvoie le fichier de résultats aux personnes concernées.

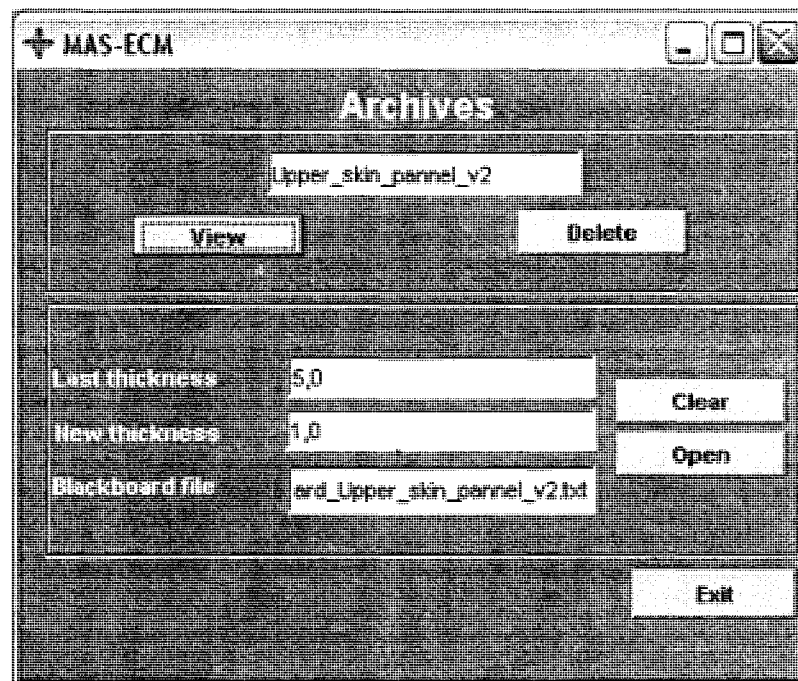


Figure 5.10 : Module de gestion des archives

#### V.2.3.7. Gestion des changements (Change checking)

Quand l'utilisateur clique sur « Change checking » ou gestion des changements, le système va chercher les pièces qui ont un statut « disponible » dans la voûte et il vérifie s'il y a plus d'une version pour chaque pièce. Si c'est le cas, le système sait qu'une pièce a déjà été modifiée.

Il va donc vérifier s'il s'agit d'une modification d'épaisseur dans le cas de notre exemple. Si c'est le cas, le système d'agents est créé.

#### **V.2.4. Le module de négociation**

L'algorithme de négociation utilisé entre les trois agents experts créés est un algorithme tableau noir, les étapes de l'algorithme sont expliquées ci-dessous :

- Récupération de la valeur du paramètre modifié à partir du système CAO par AC
- Récupération des vues sur le modèle modifié à partir de la base de données
- Création des AEs par l'AC
- Transmission du nom de la pièce (ou assemblage) et du nom et la valeur du paramètre modifié (épaisseur) aux agents AEs par l'agent AC
- Vérification des contraintes par chaque AE. Si toutes les contraintes des agents AEs sont vérifiées, ces derniers écrivent le résultat sur le tableau noir
- Notification du concepteur et des experts par l'AC
- S'il existe des contraintes non vérifiées, chaque agent AE trouve l'intervalle de solutions du paramètre modifié qui satisfait toutes ses contraintes et l'écrit sur le tableau noir
- L'AC vérifie s'il y a une intersection entre les différents intervalles trouvés. S'il y a une intersection, l'AC écrit le résultat de l'intersection sur le tableau noir et notifie le concepteur et les experts de l'intervalle de solution trouvé. Si le délai maximal de la négociation est dépassé avant de trouver une solution, la négociation s'arrête. Si par contre, il n'y a pas d'intersection, l'AC garde la valeur du paramètre proposé par le concepteur et demande à AEPC et à AEF de trouver des intervalles de solutions pour d'autres paramètres liés au paramètre modifié. Le processus est donc répété à partir de l'étape de vérification des contraintes, et ce, en prenant en considération le nouveau paramètre choisi. Les AEs écrivent

leurs propositions à nouveau sur le tableau noir.

- Si l'AC trouve une intersection, il écrit l'intervalle de solutions sur le tableau noir et déclenche le travail de AES. Si le facteur de sécurité est respecté (voir VI.5) la solution retenue est l'intervalle d'intersection. Dans ce cas, l'AC envoie la proposition au concepteur et aux experts. Si par contre, aucune intersection n'est trouvée ou dans le cas où le facteur de sécurité n'est jamais respecté, l'AC demande au concepteur de modifier sa demande de changement initiale.

### **V.3. Définition des contraintes du prototype**

En plus de la définition des pièces, des vues, des poids des paramètres, des utilisateurs et des matériaux, on a besoin de définir les contraintes qui constituent une base de connaissance pour chaque agent expert. Dans le prototype nous avons défini une contrainte pour chaque vue par souci de simplification. Cependant, il est clair que dans un cas réel on devrait définir une multitude de contraintes pour chaque type d'agent ou chaque type de vue.

#### **a) La définition de la contrainte de poids et centrage**

La première contrainte à vérifier par l'agent poids et centrage est la contrainte poids et centrage :

$$Masse\_Aile \leq 1kg \quad (1)$$

Pour la vérification de cette contrainte l'agent poids et centrage a besoin de connaître toutes les pièces de type aile et ses enfants. C'est pour cela qu'il va utiliser l'arbre de typage défini dans la figure 4.3.

#### **b) La définition des contraintes de fabrication**

La deuxième contrainte à vérifier par l'agent fabrication est la contrainte fabrication :

$$\text{Cout\_Aile} \leq 10\text{CAD\$} \quad (2)$$

Pour les mêmes raisons que l'agent poids et centrage, l'agent fabrication a besoin d'accéder à l'arbre XML de types.

L'autre contrainte à vérifier par l'agent fabrication (AEF) est la contrainte de tôle. Si on change l'épaisseur de la tôle d'une pièce, il se trouve que d'autres pièces utilisent probablement la même tôle. L'agent fabrication attire donc l'attention du concepteur et de l'expert sur cette modification qui pourrait être obligatoirement faite sur l'autre pièce également.

### c) La définition de la contrainte de structure

La dernière contrainte à vérifier par l'agent structure est la contrainte structure :

$$\text{FS} \geq 1.2 \quad (3)$$

Elle stipule que le facteur de sécurité doit être supérieur à une valeur limite. Le facteur de sécurité est défini comme étant le rapport de la limite élastique du matériau appliqué à la pièce et la contrainte maximale obtenue lors d'une analyse par éléments finis sur le produit auquel appartient la pièce étudiée. Nous savons que l'analyse par élément finis donne un résultat satisfaisant si ce facteur de sécurité est supérieur à 1.2. En fait le FS peut varier de 1 jusqu'à 1.5 selon le modèle à analyser. Nous avons choisi pour notre prototype une valeur cible de 1.2, mais dans un cas réel on pourrait choisir la valeur qui convient le mieux.

$$\text{FS} = \frac{\text{Limite élastique}}{\text{Contrainte maximale}} \quad (4)$$

Une fois que l'analyse par éléments finis déclenchée, par l'agent structure, est terminée, les résultats sont exportés vers un fichier Excel (figure 5.11). Nous avons développé un module qui enlève les valeurs dites « aberrantes » des contraintes pour obtenir la contrainte maximale correcte. Le graphe en bâtonnet de la distribution est sous forme d'une cloche et il montre donc clairement que la distribution suit une loi normale (figure 5.12). Nous avons remarqué



que les valeurs significatives se trouvent presque toujours dans l'intervalle «  $3 \cdot \text{Écart type}$  et  $-3 \cdot \text{Écart type}$  », alors on décide d'enlever toutes les contraintes qui se trouvent au-delà de cet intervalle. On obtient ainsi un ensemble de contraintes raisonnable.

	A	B	C	D
1	Axe de valeur Global			
2	x(mm)	y(mm)	z(mm)	Toutes(N_m2)
3	0	80	-39,1077	1,82E+07
4	500	80	-39,1077	2,01E+07
5	0	150	-22,5	6,40E+07
6	500	150	-22,5	2,99E+07
7	0	150	22,5	3,14E+07
8	500	150	22,5	5,72E+07
9	500	80	39,1077	3,07E+07
10	0	80	39,1077	6,13E+06
11	15,0287	80	-39,1077	5,85E+06
12	51,7501	80	-39,1077	6,45E+06
13	88,446	80	-39,1077	6,36E+06
14	123,607	80	-39,1077	6,98E+06
15	156,58	80	-39,1077	9,44E+06
16	187,792	80	-39,1077	9,48E+06

Figure 5.11 : Résultat de l'analyse par éléments finis

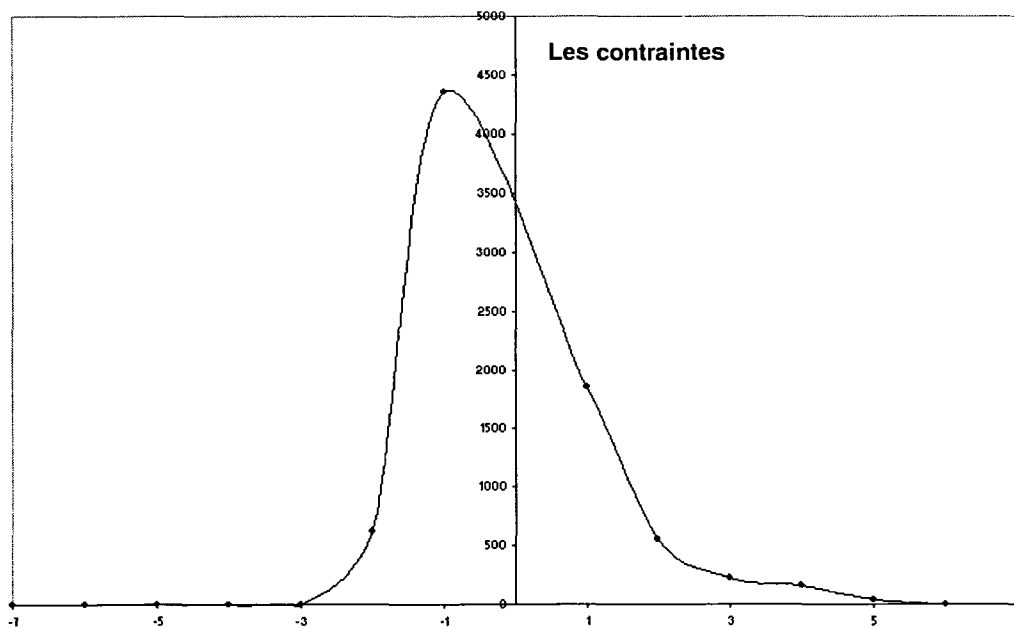


Figure 5.12 : Graphe de la distribution des contraintes

\* Écart type

## V.4. Modèle CAO utilisé dans le prototype

Nous décrivons dans cette section le modèle CAO utilisé dans le cadre du prototype développé.

### V.4.1. Modèle de conception : Aile d'un avion miniature

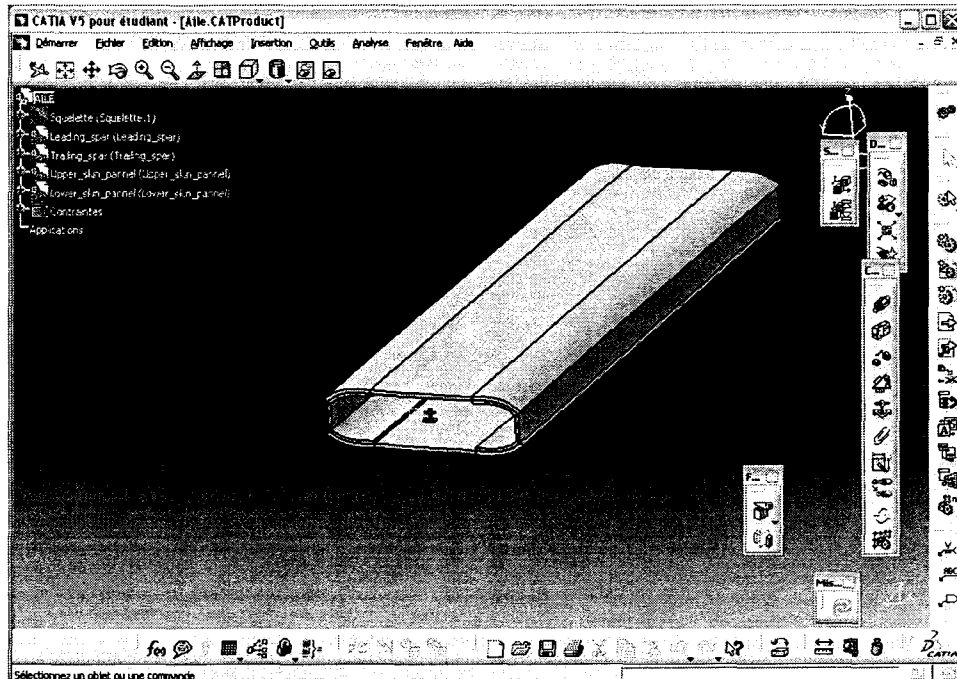


Figure 5.13 : Le modèle CAO utilisé dans le prototype

Pour tester le prototype, nous utilisons une aile d'avion miniature que nous avons conçue (figure 5.13). L'aile est assez simple, mais elle valide bien le fonctionnement du système. Elle se compose de quatre pièces principales; la poutre de bord d'attaque, la poutre de bord de fuite, le panneau supérieur et le panneau inférieur. Le squelette de l'aile permet de modifier certains paramètres communs entre ces différentes pièces par exemple la longueur, qui n'est autre que « l'extrusion », et la largeur de l'aile.

## V.4.2. Modèle d'analyse de structure

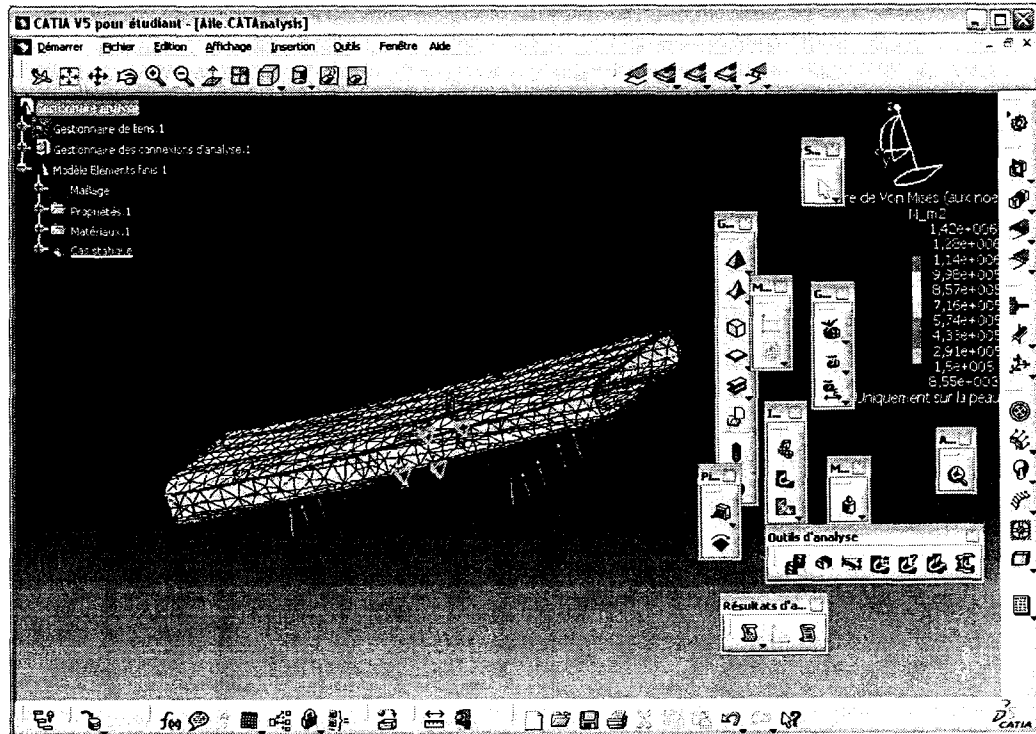


Figure 5.14 : L'analyse par éléments finis du modèle CAO utilisé

La figure 5.14 présente l'analyse par éléments finis du produit aile. Nous avons utilisé un encastrement à une extrémité et une charge de  $1000\text{N/m}^2$  à l'autre. Les résultats de cette analyse sont utilisés par l'agent structure (AES) afin de vérifier le facteur de sécurité.

## V.4.3. Bibliothèque de matériaux utilisés

Nous voulions utiliser les différents grades d'aluminium comme matériaux pour l'aile, mais dans la bibliothèque de matériaux de CATIA il n'y avait qu'un seul grade d'aluminium défini. Nous avons donc décidé de créer notre propre bibliothèque de grades d'aluminium (figure 5.15) qui se compose de : Aluminium-6063-O, Aluminium-6063-T1, Aluminium-6063-T4, Aluminium-6063-T5, Aluminium-6063-T6, Aluminium-7178-O, Aluminium-7178-T6, Aluminium-7178-T76. Les différents grades d'aluminium sont stockés dans la base de données par ordre croissant de coût pour permettre à l'agent fabrication (AEF) de trouver le

matériau le moins cher et qui vérifie ses contraintes en premier.

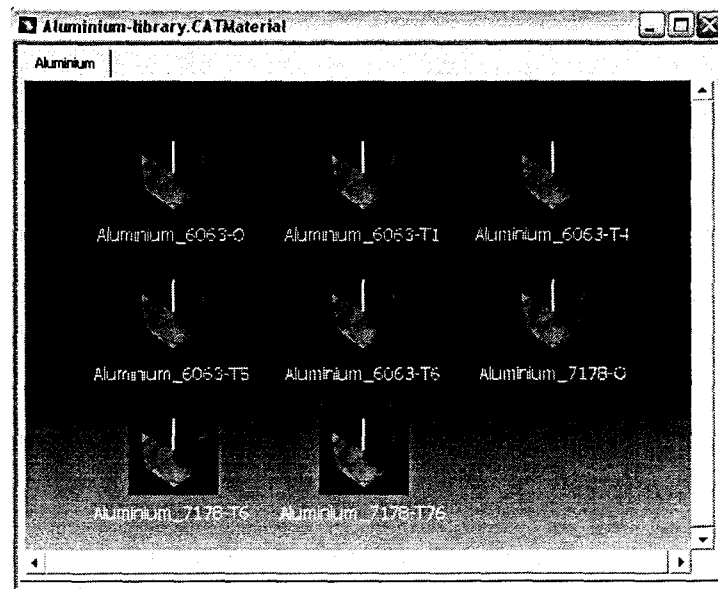


Figure 5.15 : Bibliothèque de grades d'aluminium

## V.5. Création du système d'agents

Le système multi-agents créé se compose de quatre agents logiciels; l'agent AC (Agent coordonnateur), l'agent APEC (Agent expert poids et centrage), l'agent AES (Agent expert structure) et l'agent AEF (Agent expert fabrication). Les trois agents experts représentent bien les trois départements impliqués dans le changement d'ingénierie effectué et qui sont, le département structure, le département poids et centrage et le département fabrication (figure 5.16). Lors du déclenchement du système, l'agent coordonnateur vérifie s'il y a des changements sur les pièces qui se trouvent dans la voûte. Dans le cas où il trouve un changement, il crée les différents agents experts correspondant aux vues sur la pièce modifiée, et qui correspondent aussi aux différents départements impliqués. Les Agents experts AEPC et AEF négocient un intervalle de solution commun selon l'épaisseur de la pièce ou la longueur ou la largeur de l'aile. S'ils arrivent à un intervalle commun de solution, l'agent AES vérifie à son tour le FS. Ce dernier change le matériau de la pièce concernée et modifie son épaisseur, la longueur ou la largeur de l'aile dans l'intervalle fourni par l'AEPC et l'AEF jusqu'à ce qu'il trouve une solution satisfaisante. Les différents agents inscrivent tous les résultats qu'ils trouvent sur le tableau noir (figure 5.17). Il est préférable que les agents se créent dans la

machine où se trouve les pièces (on suppose que le stockage des pièces est centralisé) pour éviter les échanges sur le réseau, et ce, par souci de robustesse (ne pas subir les pannes du réseau).

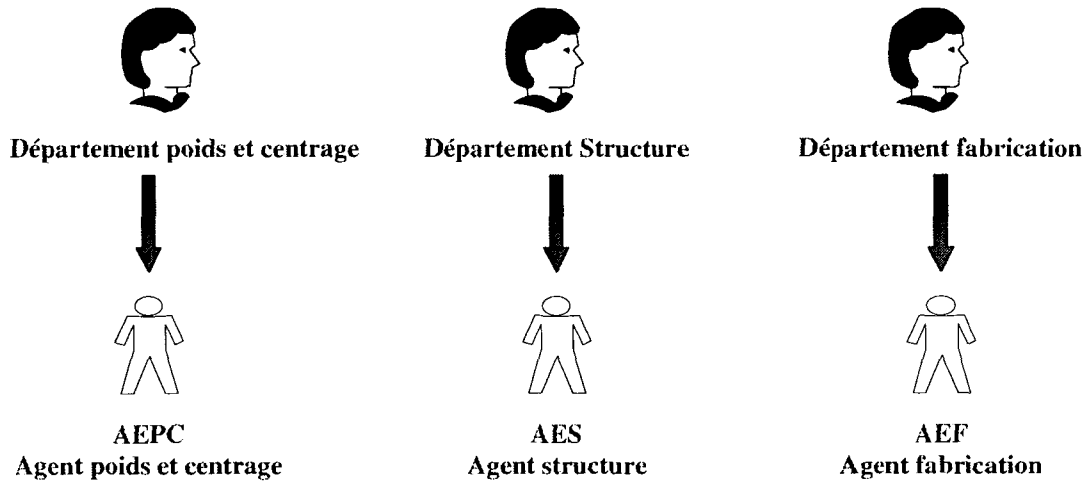


Figure 5.16 : Création du système multi-agents

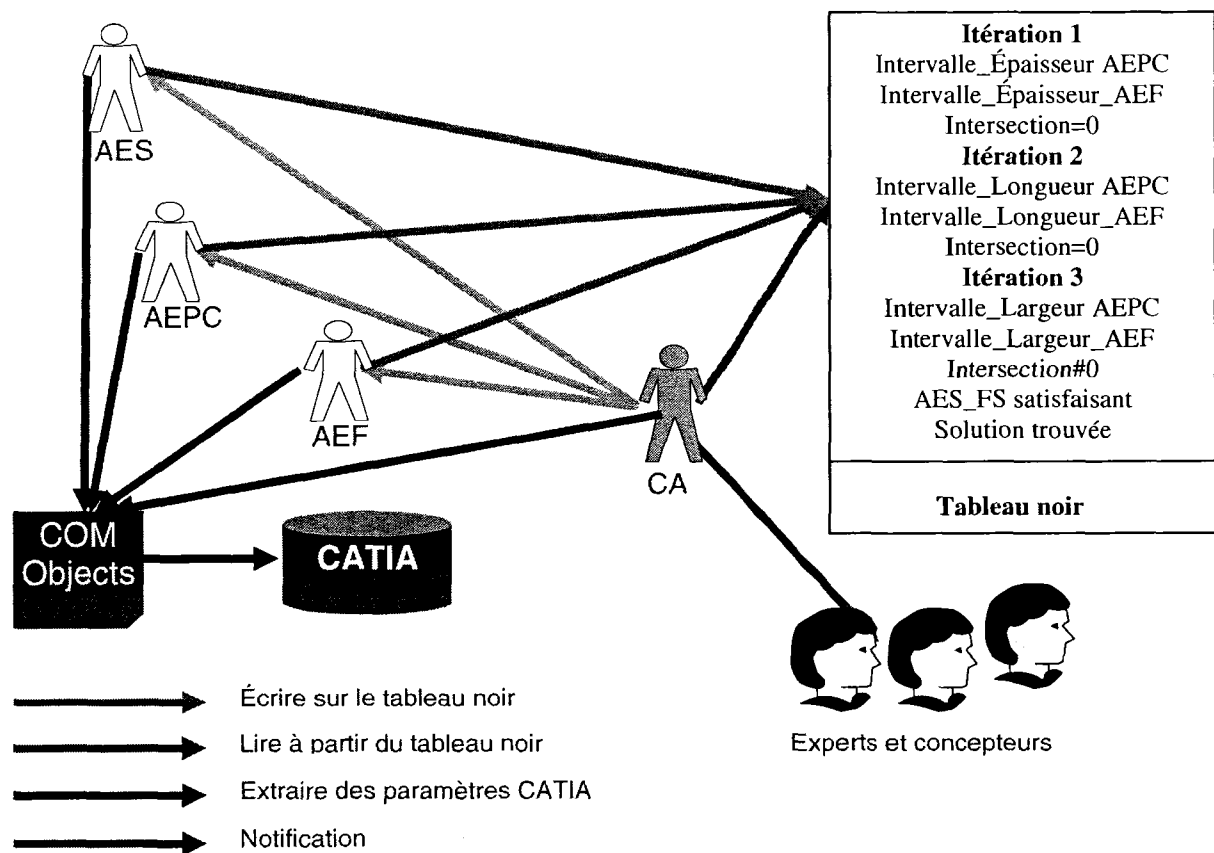


Figure 5.17 : Déploiement du système multi-agents

## CHAPITRE VI

### VI. Les résultats de l'exécution du prototype

Nous présentons dans ce chapitre le fonctionnement du prototype développé ainsi que les résultats obtenus.

#### VI.I. Exemple d'exécution du prototype

Nous décrivons dans cette section le modèle utilisé dans le prototype ainsi que les différents processus agents déclenchés lors de la vérification de la demande de changement sur la pièce « Upper\_skin\_annel\_v1 » qui fait parti de l'assemblage « Aile ».

##### VI.I.1. État initial du modèle

Nous définissons dans ce qui suit le paramétrage initial des quatre pièces composant l'assemblage « Aile ».

###### a) Paramétrage du « Upper\_skin\_annel »

Épaisseur : 5mm

Largeur : 30mm

Longueur : 500mm

Matériau : Aluminium-6063-O

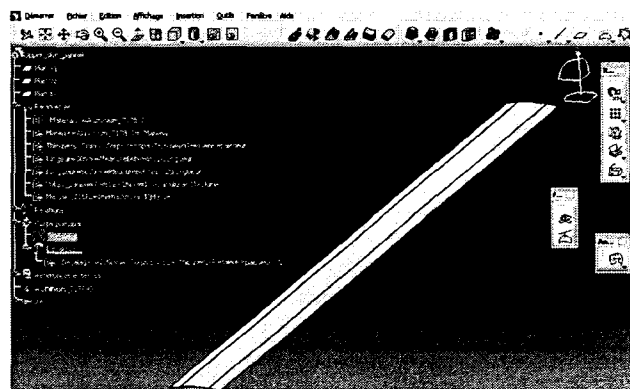


Figure 6.1 : La pièce « Upper\_skin\_annel »

**b) Paramétrage du Lower\_skin\_panel»**

Épaisseur : 1mm

Largeur : 30mm

Longueur : 500mm

Matériau : Aluminium-6063-O

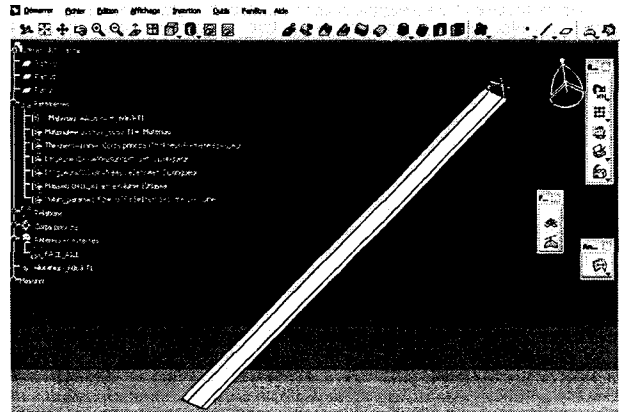


Figure 6.2 : La pièce « Lower\_skin\_panel »

**c) Paramétrage du « Leading\_spar »**

Épaisseur : 1mm

Largeur : 6,7mm

Longueur : 500mm

Matériau : Aluminium-6063-O

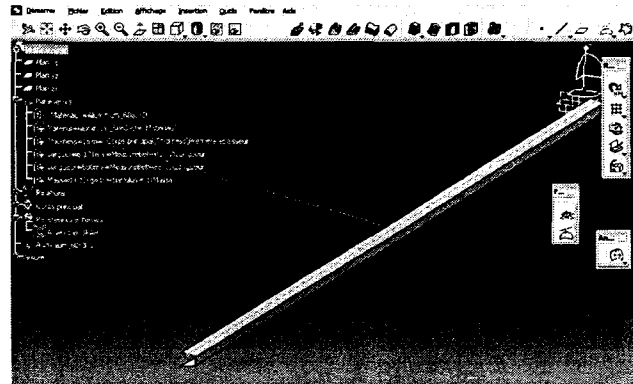


Figure 6.3 : La pièce « Leading\_spar »

#### d) Paramétrage du « Trailing\_spar »

Épaisseur : 1mm

Largeur : 6.7mm

Longueur : 500mm

Matériau : Aluminium-6063-O

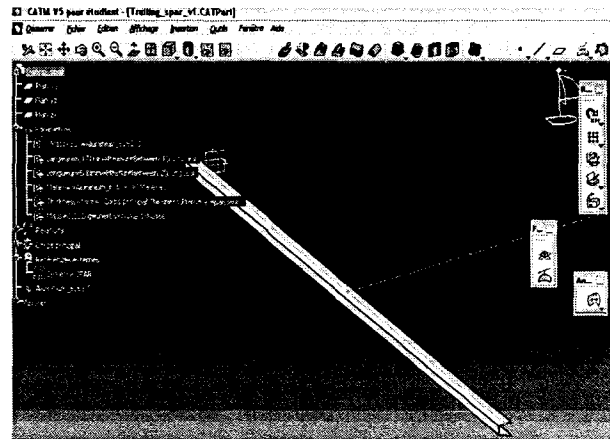


Figure 6.4 : La pièce «Trailing\_spar »

### VI.1.2. Travail des agents logiciels

Nous proposons dans cet exemple d'exécution de changer l'épaisseur du «Upper\_skin\_annel\_v1» de 5mm à 1mm et d'enregistrer la pièce sous une deuxième version; «Upper\_skin\_annel\_v2».

#### a) Processus de vérification de l'agent AC

Une fois que le concepteur fait sa demande de changement en modifiant le paramètre désiré dans le fichier du modèle CAO, la pièce devient en statut «disponible» c'est-à-dire qu'elle peut maintenant être modifiée par un autre concepteur ou être utilisée par notre système. On lance par la suite le système de gestion des changements. Le système va vérifier dans la voûte si le paramètre épaisseur a déjà été modifié dans l'une des pièces qui ont un statut «disponible» et trouve «Upper\_skin\_annel\_v1», mais constate que le changement de 5mm à 1mm n'a jamais été traité. Le programme de vérification est officiellement lancé et le système d'agents logiciels est créé. Nous créons le système d'agents directement dans la machine où les vérifications auront lieu. Dans un système plus complexe où il y a plusieurs postes de travail avec des bases de données de pièces CAO locales, nous aurions pu créer les agents localement et les faire bouger par la suite selon les disponibilités sur la machine destination afin qu'ils transportent avec eux les données dont ils ont besoin. Mais vu que ce prototype vise tout simplement à valider un ensemble restreint de changements d'ingénierie issu de la même machine, nous avons supposé que les pièces CAO sont stockées d'une manière centralisée



et donc nous créons les agents directement sur la machine destinée à la vérification et au stockage des pièces.

Premièrement, l'agent coordonnateur est créé (AC ou CA en anglais), il prend comme paramètres d'entrée le nom de la pièce. Il extrait les vues sur la pièce à partir de la base de données. Il trouve poids et centrage, fabrication et structure. Il crée par la suite les trois agents experts AEPC (Agent Expert Poids et Centrage : WBEA), AEF (Agent Expert Fabrication : MEA) et AES (Agent Expert Structure : SEA) (figure 6.5).

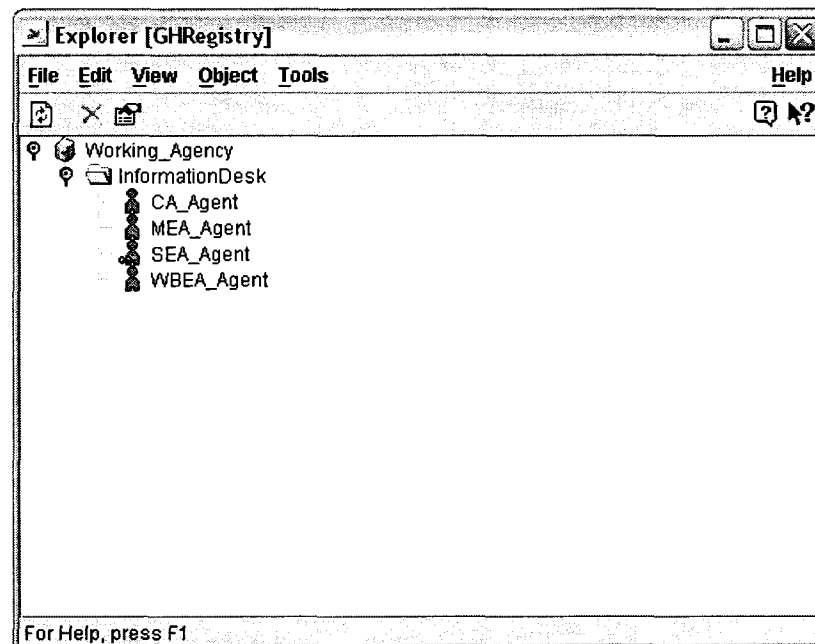


Figure 6.5 : Le système d'agents créé après le lancement du prototype

### b) Processus de vérification de l'agent AEPC

L'agent AEPC va charger la contrainte poids et centrage et va extraire le type traité dans la contrainte, qui est dans ce cas de figure, «aile». Il va consulter par la suite l'arbre de typage pour récupérer les pièces associées aux cinq types enfants du type «aile» : «panneau supérieur», «panneau inférieur», «poutre de bord de fuite», «poutre de bord d'attaque» et «nervure» (figure 4.3). Il n'y pas de pièces associées au type «nervure», l'agent récupère donc seulement quatre pièces.

**-Upper\_skin\_annel\_v2**

**-Lower\_skin\_annel\_v2**

**-Trailing\_spar\_v1**

**-Leading\_spar\_v1**

Puisque la pièce étudiée est le « Upper\_skin\_annel\_v2 », les pièces qui seront prises en compte dans les calculs sont : « Lower\_skin\_annel\_v2 », « Trailing\_spar\_v1 » et « Leading\_spar\_v1 ».

L'AEPC a toutes les données nécessaires maintenant pour proposer son intervalle de solution pour l'épaisseur et l'inscrire sur le tableau noir. L'intervalle de solution est calculé comme suit :

$$\text{Épaisseur} \leq \frac{\left( \frac{1\text{kg} - \text{Masse\_pièces\_liées\_par\_type}}{\text{Masse\_Volumique\_aluminium}} \right)}{\text{Longueur\_Upper\_skin\_annel\_v2} * \text{Largeur\_Upper\_skin\_annel\_v2}}$$

$$\text{Masse\_pièces\_liées\_par\_type} = \text{Masse\_Lower\_skin\_annel\_v2} + \text{Masse\_Trailing\_spar\_v1} + \text{Masse\_Leading\_spar\_v1}$$

### **c) Processus de vérification de l'agent AEF**

L'agent AEF est déclenché par la suite; il charge ses deux contraintes et fait le même travail qu'AEPC en récupérant les pièces correspondantes au type de la contrainte. Il trouve aussi les mêmes pièces et puisque la pièce étudiée est le « Upper\_skin\_annel », il garde seulement ces trois pièces :

**-Lower\_skin\_annel\_v2**

**-Trailing\_spar\_v1**

**-Leading\_spar\_v1**

L'AEF vérifie s'il y a d'autres pièces qui utilisent la même tôle que le «Upper\_skin\_annel\_v2» et trouve «Lower\_skin\_annel\_v2». Il écrit donc sur le tableau noir qu'il faut modifier l'épaisseur de «Lower\_skin\_annel\_v2» à 1mm. L'AEF passe par la suite à sa deuxième contrainte. Il considère que :

$$\begin{aligned} \text{Coût\_Upper\_skin\_annel\_v2} &= \text{Volume\_Upper\_skin\_annel\_v2} * \\ \text{Coût\_matériau\_Upper\_skin\_annel} \end{aligned}$$

Et donc, l'intervalle de solution pour l'épaisseur serait tout simplement.

$$\text{Épaisseur} \leq \frac{(10\$ - \text{Coût\_matériau\_pièces\_liées\_par\_types})}{\frac{\text{Longueur\_Upper\_skin\_annel\_v2} * \text{Largeur\_Upper\_skin\_annel\_v2} * \text{Coût\_Upper\_skin\_annel\_v2}}$$

$$\begin{aligned} \text{Coût\_pièces\_liées\_par\_types} &= \\ &(\text{Volume\_Lower\_skin\_annel\_v2} * \text{Coût\_matériau\_Lower\_skin\_annel\_v2}) + \\ &(\text{Volume\_Trailing\_spar\_v1} * \text{Coût\_matériau\_Trailing\_spar\_v1}) + \\ &(\text{Volume\_Leading\_spar\_v1} * \text{Coût\_matériau\_Leading\_spar\_v1}) \end{aligned}$$

La formule du volume utilisée pour extraire l'épaisseur du «Upper\_skin\_annel\_v2» est choisie selon l'indice géométrique donné dans les propriétés de chaque pièce. Dans ce cas, nous avons utilisé le volume d'un rectangle puisque le «Upper\_skin\_annel\_v2» peut être enveloppé dans une coquille rectangle; l'erreur dans notre cas n'est pas énorme. AEF trouve l'intervalle de solution pour l'épaisseur et l'inscrit sur le tableau noir.

**d) Intervention de l'agent AC**

C'est au tour de l'AC maintenant d'intervenir pour trouver l'intersection entre l'intervalle d'épaisseur trouvé par AEPC et AEF. L'AC ne trouve justement pas d'intersection, il garde l'épaisseur proposée par le concepteur et va modifier la longueur de l'aile, qui est la deuxième variable à modifier dans le cas où aucune solution n'est trouvée pour l'épaisseur, pour trouver un ensemble de solutions pour la longueur qui va satisfaire l'épaisseur désirée. L'AC relance

donc le travail des agents en leur demandant de trouver des intervalles de longueur cette fois et non d'épaisseur.

#### e) Reprise du processus de vérification de l'agent AEPC

L'AEPC trouve un intervalle de longueur pour l'aile selon la formule suivante :

$$\text{Longueur\_Aile} \leq \frac{\left( \frac{1\text{kg} - \text{Masse\_pièces\_liées\_par\_type}}{\text{Masse\_Volumique\_Aluminium}} \right)}{\text{Épaisseur\_désirée} * \text{Largeur\_Upper\_skin\_panel\_v2}}$$

Puisque la longueur du «Upper\_skin\_panel\_v2» correspond parfaitement à la longueur de l'aile, il suffit de trouver l'intervalle de longueur pour cette dernière. L'AEPC inscrit l'intervalle de longueur trouvé sur le tableau noir.

#### f) Reprise du processus de vérification de l'agent AEF

L'AEF trouve son intervalle de solution pour la longueur de l'aile selon la formule suivante :

$$\text{Longueur\_Aile} \leq \frac{(10\$ - \text{Coût\_matériau\_pièces\_liées\_par\_types})}{\text{Épaisseur\_désirée} * \text{Largeur\_Upper\_skin\_panel\_v2} * \text{Coût\_Upper\_skin\_panel\_v2}}$$

Il l'inscrit par la suite sur le tableau noir.

#### g) Intervention de l'agent AC

L'AC trouve cette fois un intervalle de solution pour la longueur de l'aile commun entre les intervalles proposés, il lance donc le travail de l'AES.

#### h) Processus de vérification de l'agent AES

L'AES trouve que l'épaisseur désirée ne vérifie pas le facteur de sécurité, il décide alors de changer le matériau courant Aluminium\_6063\_O par Aluminium\_6063\_T1. Cependant, en vérifiant le FS, il trouve que ce n'est toujours pas satisfaisant. Il décide alors de changer la

longueur de l'aile dans l'intervalle trouvé par AEPC et AEF. Pour chaque valeur de longueur comprise dans l'intervalle en question, AES vérifie le FS et il trouve que ce n'est toujours pas satisfaisant. Cette vérification a un délai maximal que l'AES ne peut pas dépasser, arrivant donc à ce « TIMEOUT », AES arrête la vérification sur l'intervalle de longueur et va changer de matériau de nouveau pour Aluminium\_7178-T76. Le « TIMEOUT » est choisi par l'expert selon les contraintes des ressources utilisées. AES refait exactement la même procédure, sa stratégie étant de parcourir tous les matériaux existants et toutes les valeurs de l'intervalle de longueur pour trouver une solution en respectant bien entendu le délai permis pour la vérification. Pour le matériau choisi, l'AES trouve une valeur de longueur de l'aile qui est incluse dans l'intervalle proposé par les autres agents experts et qui vérifie bien la contrainte du FS, il l'écrit donc sur le tableau noir.

#### **i) Reprise du processus de vérification de l'agent AEPC et de l'agent AEF**

Puisque l'AES a changé de matériau et de longueur, le coût et la masse de l'aile ont probablement changé. L'AEF et l'AEPC veillent à la vérification du coût et de la masse pour voir si ces contraintes sont toujours vérifiées ou pas.

### **VI.1.3. Conclusion**

Ce que nous avons présenté dans cette section n'est qu'un exemple de l'exécution du prototype. Il y a d'autres exemples où on trouve une solution mais après un long moment de vérification tout comme on peut ne pas avoir de solution du tout et dans ce cas on demande au concepteur de changer la valeur de l'épaisseur désirée. Quand le système d'agents termine le processus de vérification il remet tout le modèle à son état initial, car il n'est pas autorisé à le modifier. Quand les agents travaillent sur un modèle, son statut devient «non-disponible», c'est-à-dire que les concepteurs ne peuvent pas y accéder; c'est pour cela qu'il est préférable de faire la vérification la nuit et non le jour.

## **VI.2. Interprétation du contenu du tableau noir**

Au fur et à mesure que les agents travaillent, ils inscrivent les résultats qu'ils trouvent dans le

tableau noir. Nous présentons dans ce qui suit un exemple de tableau noir pour la vérification de l'épaisseur :

Début :

AC : Je commence les calculs

AEPC : Je travaille sur l'épaisseur

AEPC : L'intervalle proposé est [0 22]

AEF: Je travaille sur l'épaisseur...

AEF: L'intervalle proposé est [24 25]

AC : Pas d'intersection !

AEPC : Je travaille sur la longueur

AEPC : L'intervalle proposé est [0 11254mm]

AEF: travaille sur la longueur

AEF: L'intervalle proposé est [0 1522mm]

AC : L'intersection est [0 1522mm]!

AES : Je commence mes calculs...

AES : Le changement peut être effectué pour le matériau Aluminium\_7178-T76 et la longueur 190mm, cependant il faut changer l'épaisseur du Lower\_skin\_annel\_v2 pour 1mm

AEF : La masse est respectée pour ce changement

AEF : Le coût est respecté pour ce changement

Fin.

### **VI.3. Notification des utilisateurs**

Selon les vues sur la pièce modifiée, l'agent AC sait qui aviser du résultat de la vérification en consultant la table « archives » de la base de données. AC envoie un courriel à chaque intervenant avec le fichier tableau noir correspondant à la vérification. C'est aux experts maintenant de décider s'ils vont accepter la modification proposée par le concepteur ou la rejeter, et ce, selon les résultats inscrits sur le tableau noir.

### **VI.4. Performance du prototype**

Nous avons décidé de créer les agents directement dans la machine destinée à la vérification, nous n'avons donc pas besoin de faire l'étude de performance évoquée dans la section III.5, étant donné qu'on ne génère aucun trafic sur le réseau. En effet, pour avoir des résultats significatifs de l'étude de performance, il faut déployer le prototype dans un environnement de conception distribué réel. Ce qui était dans notre cas difficile à réaliser.

Nous pensons que l'étude de performance devrait s'attarder sur le temps d'exécution moyen du système avant de trouver une solution au problème. Il est clair que si on décidait de parcourir les intervalles de solution sans spécifier un délai maximal, la recherche de la solution pourrait prendre beaucoup de temps et pourrait donc coûter cher en termes de ressources (machines) également. D'après les tests que nous avons effectués, le délai que nous avons spécifié permet de trouver une solution, mais qui n'est pas forcément optimale. En effet, quand on décide d'arrêter la recherche dans un intervalle pour changer le matériau, nous avons peut-être raté une solution qui coûte moins cher en terme de matériau, mais plus cher en terme de temps. Donc, afin d'améliorer encore plus les performances de notre système, il faudrait voir ce qui intéresse l'entreprise dans laquelle le système pourrait être implanté, est-ce l'optimalité en terme de temps ou en terme de coût du matériau utilisé? Si par exemple l'entreprise décide de faire les calculs la nuit, le temps de calcul n'aurait pas vraiment d'importance et dans ce cas l'intervalle de solutions serait discrétisé d'une manière plus finement, et la solution trouvée coûterait plus cher en terme de temps de calculs et moins cher en terme de matériau.

## Conclusion

Nous avons présenté dans ce document une architecture multi-agents qui va aider les concepteurs à faire des changements d'ingénierie et qui va les assister dans la prise de décision vis-à-vis de ces changements.

Nous avons décomposé notre thèse en cinq parties. Dans la première partie, nous avons présenté les différents systèmes d'information existants dans les entreprises de conception mécanique, à savoir les systèmes CAO/FAO, les SGGTs et les PGI. Nous avons spécifié ensuite les différents problèmes rencontrés par ces entreprises dans le processus d'intégration de ces différents systèmes afin d'assurer une meilleure gestion de l'information.

La deuxième partie contient l'état de l'art relatif à notre recherche. Nous avons remarqué que tous les travaux présentés dans cette section aident à informer les bonnes personnes au bon moment du changement fait sur les documents de conception. Ils permettent aussi de propager les modifications. Cependant, ils n'offrent aucune assistance aux décideurs. Même les systèmes à base d'agents logiciels qui ont été réalisés dans ce sens ne parlent pas explicitement des problèmes de gestion des modifications d'ingénierie; ils se focalisent plutôt sur les problèmes de conception. En plus, ce sont des solutions complètement intégrées aux systèmes CAO ce qui n'est pas vraiment avantageux dans un cadre d'ingénierie distribuée, multiplateformes, intégrant de nombreux sous traitants.

En tenant compte de ce que nous avons présenté dans les sections précédentes, nous proposons, dans la troisième section, l'architecture d'un système de gestion du changement capable d'automatiser le processus de modifications d'ingénierie, tout en assurant la cohérence des données des différentes disciplines touchées. En plus de cela, le système proposé assiste les concepteurs et les experts dans leurs décisions quand ils font face à une situation de conflit. Notre système est à base d'agents logiciels, chaque agent représentant une discipline donnée. Nous utilisons un protocole de négociation afin que les agents convergent vers la solution la plus adéquate pour l'utilisateur. Nous avons présenté un exemple



d'application qui illustre clairement le fonctionnement du dit système. Nous avons aussi jugé intéressant d'étudier les possibilités d'intégration de notre solution avec le système à entités parlantes développé à l'ETS.

Dans la quatrième section, nous présentons le prototype que nous avons développé pour valider le concept proposé dans la section qui précède. Enfin, nous donnons les résultats de notre recherche dans la dernière section.

L'originalité de notre système multi-agents se justifie par le fait qu'il permet d'assister les concepteurs à prendre des décisions vis-à-vis des changements qu'ils désirent effectuer sur des modèles CAO. Ainsi, si les changements demandés ne peuvent être effectués, le système propose des solutions alternatives. Notre système est également indépendant du système de CAO utilisé; toute son intelligence se trouve à l'extérieur du système CAO. Il est robuste; cela veut dire que les pannes de réseaux n'influencent pas le travail des agents puisque la négociation se fait sur une seule machine. Enfin, il pourrait être adapté à tous les domaines de l'ingénierie simultanée dans le cadre de la conception mécanique assistée par ordinateur.

Pour utiliser le prototype développé les modèles CAO utilisés doivent être paramétrables ce qui n'est probablement pas le cas dans les gros projets dans le domaine de l'aéronautique ou de l'automobile par exemple. Donc, cette application serait idéale pour des projets contenant des petites pièces parfaitement paramétrables. Ceci dit, cela ne va pas contre le fait que notre architecture pourrait traiter aussi des modèles CAO non paramétrables, il faudrait donc trouver une autre manière que le paramétrage pour extraire et modifier des données directement dans les modèles CAO.

Nous jugeons que nous avons accompli notre mandat en ce qui concerne notre projet de recherche. Le travail que nous avons effectué est la validation d'un concept qui constitue une solution pour le problème de la gestion des changements d'ingénierie dans le domaine de la conception assistée par ordinateur. Nous pensons que le plus dur est fait, l'idée de l'intégration des agents à la CAO pour l'automatisation des processus liés aux changements d'ingénierie est validée. Le prototype a besoin maintenant d'être perfectionné pour qu'il puisse être testé en supportant les contraintes d'une vraie entreprise utilisant la conception

assistée par ordinateur. Enfin, une étude de performance doit être faite pour démontrer que la solution que nous proposons est une solution optimale en termes de temps et d'argent.

Dans nos recherches futures, nous comptons étudier les possibilités d'intégrer le domaine de l'intelligence artificielle et le domaine de la CAO collaborative. Nous pensons que des techniques pourraient être mises en place afin d'améliorer le domaine de la conception collaborative. Dans nos projets de maîtrise et de doctorat, nous avons intégré les agents logiciels à la CAO et nous avons ensuite étudié les algorithmes de négociation qui pourraient être appliqués au sein d'un système multi-agents œuvrant pour la CAO. Nous comptons aussi mettre l'accent sur l'intelligence de ces agents logiciels en y intégrant des algorithmes issus de l'intelligence artificielle, par exemple des réseaux de neurones ou des algorithmes génétiques. De cette manière, les agents seraient plus au moins indépendants et pourraient ainsi raisonner d'une manière autonome pour résoudre des problèmes liés à la conception collaborative au sein des systèmes de CAO

## Bibliographies

3D XML, [ <http://www.3ds.com/3dxml> ], (Consulté en 2005).

AGILE, [ <http://www.agile.com/> ], (Consulté en 2005).

ALBIN, S., L., CREFELD, P., (1996) *Getting started: Concurrent engineering for a medium-sized manufacturer*, Journal of manufacturing systems, Elsevier, Netherlands, 10 p.

AMR Research, [ <http://www.amrresearch.com/Content/Topic.asp> ], (Consulté en 2005).

ANDRAS, J., (1995) *Petri nets and AI in modeling and simulation*, Mathematics and computers in simulation, Elsevier, Netherlands, 39 p.

BERKER, I., BROWN, D., C., (1996) *Conflict and negotiation in single agent based design systems*, Concurrent engineering : Research and applications, Technomic Publishing, USA, 16 p.

BERNUS, P., NEMES, L., WILLIAMS, T., J., (1994) *A framework to define a generic enterprise reference architecture and methodology*, ICARV'94, Singapore, 4 p.

BOURKE, D., BOYER, T., (2003) *Product & Portfolio Management: Overview and application software*, 5 p.

BROWN, D., C., (1996) *Single Function Agents for Design*, Knowledge Intensive CAD, Volume I, (Eds.) T. Tomiyama, M. Mantyla & S. Finger, Chapman & Hall (IFIP), 5 p.

BURKETT, M., CARILLO, L., O'MARAH, K., (2003) *CAD versus ERP Versus PDM: How best anchor a PLM strategy?*, AMR Research Report, 36 p

BURKETT, M., KEMMETER, J., O'MARAH, K., (2003) *Product lifecycle management:*

*What's real now?*, AMR Research Report, 23 p.

CHAN, F., T., S., ZHANG, J., LI, P., (2001) *Modelling of integrated, distributed, and cooperative process planning system using an agent-based approach*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, UK, 15 p.

CHEN, S., J., LIN, L., (2002) *A Project task coordination model for team organization in concurrent engineering*, Concurrent engineering: research and applications, Sage publications UK, 16 p.

CHEN, Y., M., SHIR, W., S., SHEN, C., Y., (2002) *Distributed engineering change management for allied concurrent engineering*, Journal of computer integrated manufacturing, Taylor & Francis, UK, 25 p.

CHENG, G., TORU, I., (January 1996) *Analyzing the Social Behavior of Contract Net Protocol*, Modelling Autonomous Agents in a Multi-Agent World, Netherlands, 12 p.

CHEUNG, W. M., GAO, J. X., HAYDAR, A., MAROPOULOS, P. G. (2003) *Application of product data management technologies for enterprise integration*, Journal of computer integrated manufacturing, Taylor & Francis, UK, 10 p.

CLARK, K., B., FUJIMOTO, T., (1991) *Product development performance: Strategy, organisation and management in the world auto industry*, Harvard Business school press, Cambridge, MA, 409 p.

COLIN, P., (2001) *MySAPPLM Product Lifecycle Management "Driving Product and Process Innovation through Product Lifecycle Management"*, SAP Africa, 23 p.

CRM, [<http://www.destinationcrm.com/articles/default.asp?ArticleID=2308>], (Consulté en 2005).

CROSS, N. (2000) *Engineering Design Methods*, New York, USA, 212 p.

CUTKOSKY, M. R., ENGELMORE, R. S., FIKES, R.E., GENESERETH, M. R.,

GRUBER, T. R., MARK, W. S., TENENBAUM, J. M., WEBER, J. C., (1993) *PACT: An Experiment in Integrating Concurrent Engineering Systems*, Sanford, USA, 9 p.

D'AMBROSIO, j., DARR, T., BIRMINGHAM, W., (1996) *Hierarchical concurrent engineering in a multiagent framework*, Concurrent engineering: Research and applications, Technomic publishing, USA, 11 p.

DEC, FUJITSU, IBM, MATRIX ONE, MÉTAPHASE, SDRC, SHERPA, (1998) *PDM enablers*. Joint proposal to the OMG in response to OMG manufacturing domain task force RFP1, mgf/1998-01-01, 248 p.

DECKER, K., LI, J., (2000) *Coordinating mutually exclusive resources using GPGP, Autonomous agents and multi-agent systems*, Kluwer academic publisher, Netherlands, 25 p.

DEJEAN, M., MARKUS, B., INGO, B., JOHN, C., STEFAN, C., BARRY, F., KAZUYA, K., DANNY, L., KOUICHI, O., MITSURU, O., CYNTHIA, T., SANKARA, V., JIM, W., (1998) *MASIF The OMG Mobile Agent System Interoperability Facility*, 18 p.

DERELI, T., BAYKASOGLU, A., (2004) *Concurrent engineering utilities for controlling interactions in process planning*, Journal of intelligent manufacturing, Kluwer academic publishers, Netherlands, 8 p.

Design Space Colonization, [<http://www-cdr.stanford.edu/DSC/>] (Consulté en 2005).

EDS, [<http://www.eds.com/services/alliances/ugs/>], (Consulté en 2005).

ENOVIA - Collaborative Product Lifecycle Management, [[www.enovia.com](http://www.enovia.com)] (Consulté en 2005).

EUSTACHE, J., MARANZANA, R., LANUEL, Y. GARDAN, Y., (7-9 October 2001) *Managing complexity in a CAD environment*, Change Management and the New Industrial Revolution, New York, USA, 5 p.

FEIJÒ, B., BENTO, J., (1998) *A logic-based environment for reactive agents in intelligent*

*CAD systems*, Advances in engineering software, Elsevier, Netherlands, 8 p.

FININ, T., WEBER, J., WIEDERHOLD, G., GENESERETH, M., FRITZSON, R., MCKAY, D., MCGUIRE, J., PELAVIN, R., SHAPIRO, S., BECK, C., (1993) *Specification of the KQML Agent-communication language*, The DARPA knowledge sharing initiative external interfaces working group, 34 p

Function Blocks for Industrial –Process Measurement and Control Systems, 2000. IEC TC65/WG6, IEC-TC65/WG6 committee.

FOUGÈRES, A., J., (10-13 October 2004) *Agents to Cooperate in Distributed Design*, 2004 IEEE International conference on systems, man and cybernetics, Hague, Netherlands, 6 p.

FROST, R., H., CUTKOSKY, M., R., (18-22 August 1996) *Design for manufacturability via agent interaction*, 1996 ASME design for manufacturability conference, Irvine, California, 11 p.

GIBSON, N., HOLLAND, B. (1999) *A Critical Success Factors Model for Enterprise Resource Planning Implementation*, IEEE Computer Society Press, USA, 14 p.

GRUNINGER, M., FOX, M.S., (May 1996) *"The Logic of Enterprise Modelling"*, Modelling and Methodologies for Enterprise Integration, Proceedings of the Industrial Engineering Research Conference, IIE, Nashville, USA, 15 p.

HABHOUBA, D., DESROCHERS, A., CHARKAOUI, S., (7-10 May 2006) *Verification of product technical specifications in CAD systems using software agents*, Canadian conference on electrical and computer engineering, Ottawa, Canada, 7 p.

HAITAMI, M. KHOUMSI, A., (2005) *Évaluation de performance de methodes Client-Serveur et agent mobile de résolution d'interactions de services téléphoniques*, Mémoire de maîtrise, Sherbrooke, Canada, 91 p.

HARHALAKIS, G., (1986) *Engineering changes for made-to-order products: how an MRP II system should handle them*, Engineering management international, Elsevier, Netherlands,

18 p.

HUANG, G., Q., MAK, K., L., (1999) *Current practices of engineering change management in UK manufacturing industries*, International Journal of operations and production management, Emerald, Bingley, UK, 16 p.

HUANG, G., Q., YEE, W., Y., MAK, K., L., (2001) *Development of a web-based system for engineering change management*, Robotics and computer integrated manufacturing, Pergamon Press, UK, 13 p.

IBM PLM Solutions, [<http://www-1.ibm.com/solutions/plm/>], (Consulté en 2005).

ICARESPACE, [[http://www.icare-espace.com/ing\\_sim1.html](http://www.icare-espace.com/ing_sim1.html)] (Consulté en 2005).

KALATOKA, R., ROBINSON, M., TAPSCOTT, D., (1999) *EBusiness– Roadmap to Success*, Addison-Wesley, Reading, MA. 378 p.

KAO, H., P., SU, E., WANG, B., (2002) *I<sup>2</sup>QFD: A blackboard-based multiagent system for supporting concurrent engineering projects*, International journal of production research, Taylor & Francis, UK, 28 p.

KORNIENKO, S., KORNIENKO, O., LEVI, P., (February 2003) *Flexible manufacturing process planning based on the multi-agent technology*, Proceedings 21<sup>st</sup> IASTED, Applied informatics, Austria, 6 p.

KUNWOO, L. (1999) *Principles of CAD / CAM / CAE systems*, Addison Wesley Longman, USA, 582 p.

LIM, M., ZHANG, Z., APPSS – (3-6 July 2000) *An agent-based dynamic process planning and scheduling system*, Proceedings volume from the IFAC workshop, Russia, 6 p.

LIU, D., Z., LIU, M., ZHONG, P., S., (2004) *Method of product development process analysis and reengineering for concurrent engineering*, Materials science forum, Trans tech

publications, Switzerland, 5 p.

MARCIA, I., SICHMAN, S., J., (9-13 August 2004) *Distributed AI and Multi-agent Systems - Dependence Based Coalitions and Contract Net: A Comparative Analysis*, Pacific Rim International Conference on Artificial Intelligence, New Zealand, 10 p.

MARTINO, D., T., FALCIDIENO, B., HABINGER, S., (1998) *Design and engineering process integration through a multiple view intermediate medeller in a distributed object-oriented system environment*, Computer aided design, Elsevier, Netherlands, 16 p.

MCINTOSH, K., G., (1995) *Engineering DATA management: A guide to successful implementation*, McGraw-Hill, New York, 279 p.

MCCMAHON, C., BROWNE, J. (1998) *CAD/CAM principles, practice and manufacturing management*, 2<sup>ème</sup> édition, Addison Wesley Longman, USA, 665 p.

MATRIXONE, [<http://www.matrixone.com/>], (Consulté en 2005).

METAPHASE, [[www.sdrc.com/metaphase](http://www.sdrc.com/metaphase)] (Consulté en 2005).

MILL, F., G., SALMON, J., C., PEDLEY, A., G., (1993) *Representation problems in feature-based approaches to design and process planning*, International journal of computer integrated manufacturing, Taylor & Francis, UK, 6 p.

MOKHTAR, A., BEDARD, C., FAZIO, P., (1998) *Information model for managing design changes in a collaborative environment*, Journal of computing in civil engineering, American Society of Civil Engineers, USA, 10 p.

NASO, D., TURCHIANO, B., (2004) *A coordination strategy for distributed multi-agent manufacturing systems*, International journal of production research, Taylor & Francis, UK, 24 p.

ORACLE, [<http://www.oracle.com/applications/plm>], (Consulté en 2005).



- ORACLE, (2003) *Data sheet: Oracle CADView 3-D*, 9 p.
- OSACA, (1996) [<http://www-poleia.lip6.fr/~guessoum/asa/afia/osaca.html>], Université de Technologie de Compiègne, France.
- Parametric Technology Corporation, (2003) *Getting Started with Pro/ENGINEER® Wildfire™*, 190 p.
- PAWAR, K., S., SHARIFI, S., (1997) *Physical or virtual team location: Does it matter?* International journal of production economics, Elsevier, Netherlands, 7 p.
- PLM Community, [[http://www.cplmcommunity.com/sourcing/ptc/doc\\_p2.pdf](http://www.cplmcommunity.com/sourcing/ptc/doc_p2.pdf)], (Consulté en 2005).
- PRASAD, B., (1996) *Concurrent Engineering: Fundamentals: Integrated Product & Process Organization*, Volume I, Textbook Binding, USA, 502 p.
- Product Lifecycle Management Information Center, [[www.plmic.com](http://www.plmic.com)], (Consulté en 2005).
- PTC, [<http://www.ptc.com/>], (Consulté en 2005).
- Quest software, [<http://www.quest.com/oracle-e-business-suite>], (Consulté en 2005).
- ROSENMAN, M., WANG, F., (2001) *A component agent based open CAD system for collaborative design*, Automation in construction, Elsevier, Netherlands, 15 p.
- ROUIBEH, K., CASKEY, K., R., (2003) *Change management in concurrent engineering from a parameter perspective*, Computer in industry, Elsevier, Netherlands, 20 p.
- ROUIBEH, K., CASKEY, K., R., (2003) *A workflow system for the management of inter-company collaborative engineering processes*, Journal of Engineering design, Taylor & Francis, UK, 22 p.
- RUDAS I., I., HORVATH, L., (1997) *Modeling of manufacturing processes using a Petri net*

*representation*, Engineering application of artificial intelligence, Elsevier, Netherlands, 10 p.

SATHI, A., FOX, M., S., (1989) *Constraint-directed negotiation of resource allocations*, Gasser and M. N. Huhns, eds., Distributed artificial intelligence 2, Morgan Kaufmann, Los Altos, California, USA, 30 p.

SHAH, J., J., WOUDFILL, M., C., WILSON, A., F., (1995) *The virtual corporation*, ASME Design Theory & Methodology Conference, California, USA, 20 p.

SHEN, W., (2002) *Distributed Manufacturing Scheduling using Intelligent Agents*, IEEE Intelligent Systems and Their Applications, Integrated Mfg. Technol. Institute, National Research Council Canada, Ontario, Canada, 6 p.

SHEN, W., BARTHÉS, J. P., (1995) *DIDE: A multiagent Environment for Engineering Design*, Université de Technologie de Compiègne, France, 7 p.

SHEN, W., BARTHÈS, J. P., (1996) *An Experimental environment for exchanging design knowledge by cognitive agents*, In M, Mantyla, S. Finger and T. Tomiyama, eds., knowledge Intensive CAD, Compiègne, France, 9 p.

SHEN, W., DOUGLAS, H. N., (1998) *An Agent-Based Approach for Dynamic Manufacturing Scheduling*, University of Calgary, Calgary, Alberta, Canada, 11 p.

SHEN, W., LIHUI, W., QI, H., (2006) *Agent-Based Distributed Manufacturing Process Planning and Scheduling : A State-of-the-Art Survey*, IEEE Transactions on systems, man and cybernetics, USA, 15 p.

SHEN, W., MATURANA, F., NORRIES, D. H., (2000) *MetaMorph II: An agent-based architecture for distributed intelligent design and manufacturing*, University of Calgary, Calgary, Alberta, Canada, 14 p.

SHEN, W., XU, Y., KORBA., L., WANG, L., HAO, Q., LANG, S., (2003) *A security framework for collaborative distributed system control at the device-level*, Industrial

Informatics, Proceedings. IEEE International Conference on  
Volume , 6 p.

SHEN-CHOU, Y., CHUN-FONG, Y. (2002) *STEP-based data schema for implementing product data management system*, Journal of computer integrated manufacturing, Taylor & Francis, UK, 16 p.

SIEGEL, B., (1991) *Organization for a successful CE process*, Industrial engineering, Inderscience publishers, UK, 4 p.

SIERRA, C., JENNINGS, N., R., NORIEGA, P., PARSONS, S., (24-26 July 1997) *A framework for argumentation-based negotiation*, Proc. Fourth Int. Workshop on Agent Theories, Architectures and Languages (ATAL-97), Rhode Island, USA, 17 p.

SMITH, R., P., (1997) *The historical roots of concurrent engineering fundamentals*, IEEE transactions on engineering management, Washington, USA, 11 p.

SOLIMAN, F., GLEGG, S., TANTOUSH, T., (2001) *Critical success factors for integration of CAD/CAM systems with ERP systems*, International Journal of Operations & Production Management, Emerald, Bingley, UK, 21 p.

SOUFI, K., CHATELAIN, J.-F., MARANZANA, R., (16-19 July 2006) *A Product Change Management System based on Active Features*, CIRP Design Seminar, Kananaskis, Alberta.

STEWART, D., V., (1981) *System analysis and management structure*, Strategy and design, Petrocelli books, Princeton, NJ, 287 p.

STEWART, D., V., (1981) *The design structure system: A method for managing the design of complex systems*, IEEE transactions on engineering management, Washington, USA, 3 p.

STRATMAN, J., ROTH, A. (20-23 November 1999) *Enterprise resource planning competence: a model, Propositions and pre-test, design-stage scale development*, 30th DSI Proceedings, New Orleans, USA, 2 p.

SUTHERLAND, I.E, and MIT Lincoln Laboratory, (1963) *Sketchpad: A Man-Machine Graphical Communication System*, Detroit, Michigan, USA, 18 p.

SYCARA, K., (1989) *Multi-agent compromise via Negotiation*, L. Gasser and M. N. Huhns, eds., *Distributed artificial intelligence 2*, Morgan Kaufmann, Los Altos, California, USA, 18 p.

SYCARA, K., (1991) *Cooperative negotiation in concurrent engineering design*, *Cooperative engineering design*, Springer Verlag publications, Germany, 28 p.

TANER, B., (14-17 September 1997) *Product data management systems: State-of-the-art and the future*, ASME design engineering conferences, Sacramento, California, 7 p.

TANJA, N., DIRK, K., HOLGER, L., (2002) *Requirements for software-support in concurrent engineering teams*, Behaviour & information technology, Taylor & Francis, UK,

TERWIESCH, C., LOSH, C., H., (1999) *Managing the process of engineering change orders: the case of the climate control system in automobile development*, *Journal of product innovation management*, Elsevier, Netherlands, 12 p.

TUROCY, T., L., STENGEL, B., V., (2002) *CDAM Research Report LSE-CDAM-2001-09*, *Encyclopedia of Information Systems*, Academic Press, Elsevier, Netherlands, 39 p.

UGS, [<http://www.ugs.com/index.shtml>], (Consulté en 2005).

USHER, J., M., (1996) *Implementing concurrent engineering in small manufacturing enterprises*, *Engineering management journal*, IEEE, UK, 10 p.

WEBER, F., PAWAR, K., BARSON, R., SANTORO, R., (June 1999) *Approach and concepts for a methodology and software system for the implementation and improvement of concurrent engineering in small and medium enterprises in the aeronautics industry*, 5<sup>th</sup> *International conference on concurrent enterprising*, Netherlands, 8 p.

WEIDOG, F., MING X., T., FRAZER, H., J., (2003) *Supporting collaborative product*

*design in an agent based environment*, Design technology research center, Hong Kong, China, 14 p.

WERKMAN, K., J., (October 1990) *Knowledge-based model of using shareable perspectives*, Proceedings of the 10th international workshop on DAI, Texas, USA, 22 p.

XINZHONG, Z., JIANMIN, Z., HUIYING, X., RONG, L., (3-5 May 2006) *a Research of Collaborative CAD System based on Muti-Agent*, Proceedings of the 10<sup>th</sup> international conference on computer supported cooperative work in design, Nanjing, China, 4 p.

YAN, H., S., WANG, Z., JIANG, M., (2002) *A quantitative approach to the process modeling and planning in concurrent engineering*, Concurrent engineering: research and applications, Sage publications, UK, 15 p.

YOSHIMURA, M., TAKAHASHI, K., (2001) *Collaborative design among different fields in mobile-agent environment*, Concurrent engineering: Research and applications, Technomic publishing, USA, 9 p.

YUEFEI, X., RONGGONG, S., LARRY, K., LIHUI, W., WEIMING, S., SHERMAN, L., (2005) *Distributed Device Networks With Security constraints*, IEEE Transactions on industrial informatics, USA, 9 p.

ZAIRI, M., SINCLAIR, D. (1995) *Business process re-engineering and process management: a survey of current practice and future trends in integrated management*, Business Process Management Journal, Emerald, Bingley, UK, 23 p.

ZEID, I., (1991) *CAD/CAM theory and practice*, McGraw-Hill, Science/Engineering/Math, Boston, USA, 376 p.

ZHA, X., F., (2002) *A knowledge intensive multi-agent framework for cooperative/collaborative design modeling and decision support of assemblies*, Knowledge-Based systems, Elsevier, Netherlands, 14 p.

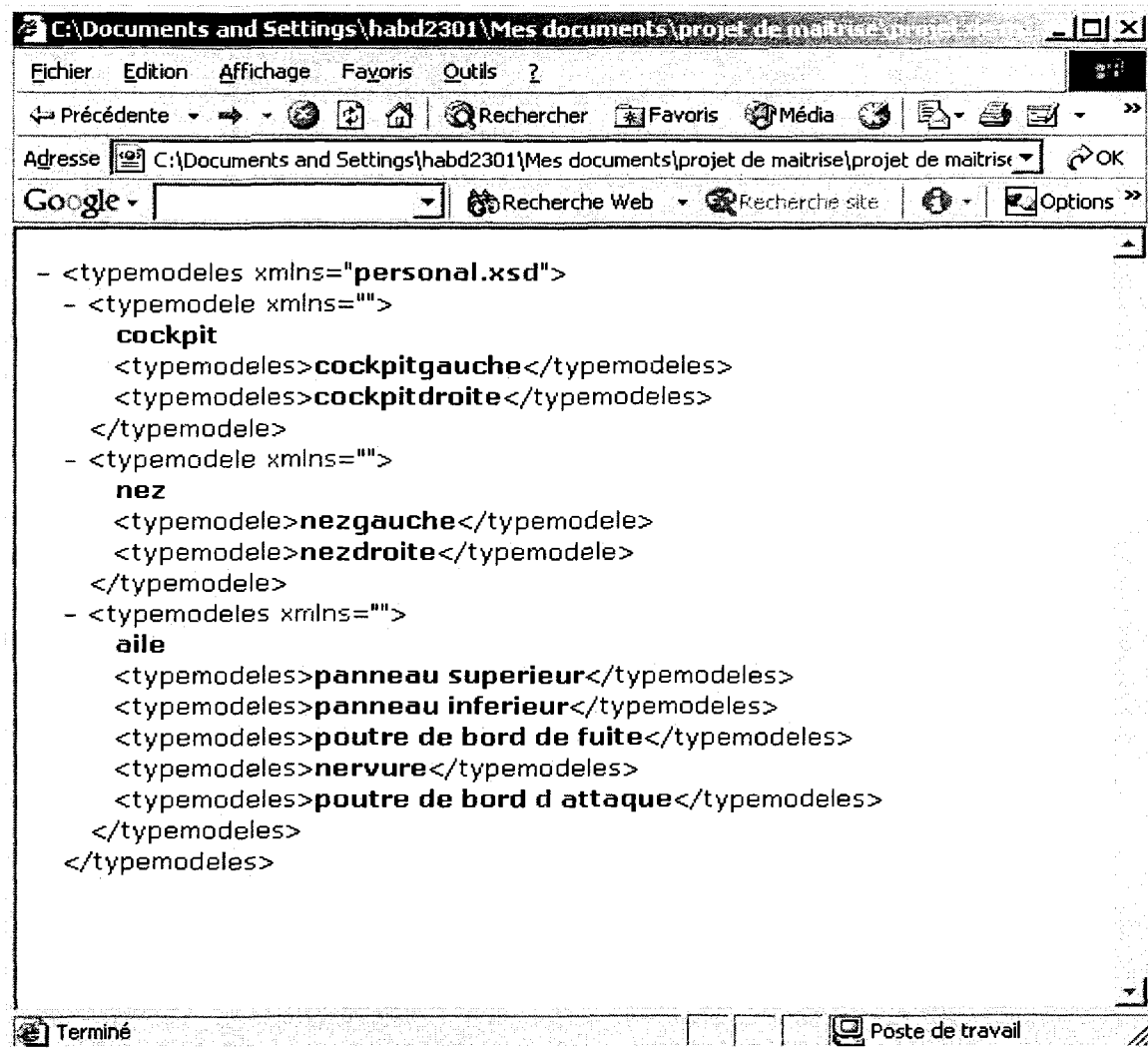
ZHA, X., F., LIM, S., Y., E., (2000) *Assembly/Disassembly task planning and*

*simulation using expert Petri nets*, International journal of production research, Taylor & Francis, UK, 38 p.

ZHONGHUI, W., CHEE, K. S., (2001) *Managing Design Changes for Multiview Models*, Journal of computing in civil engineering, American Society of Civil Engineers, USA, 9 p.

## ANNEXES

### a) Exemple arbre de typage sous format XML

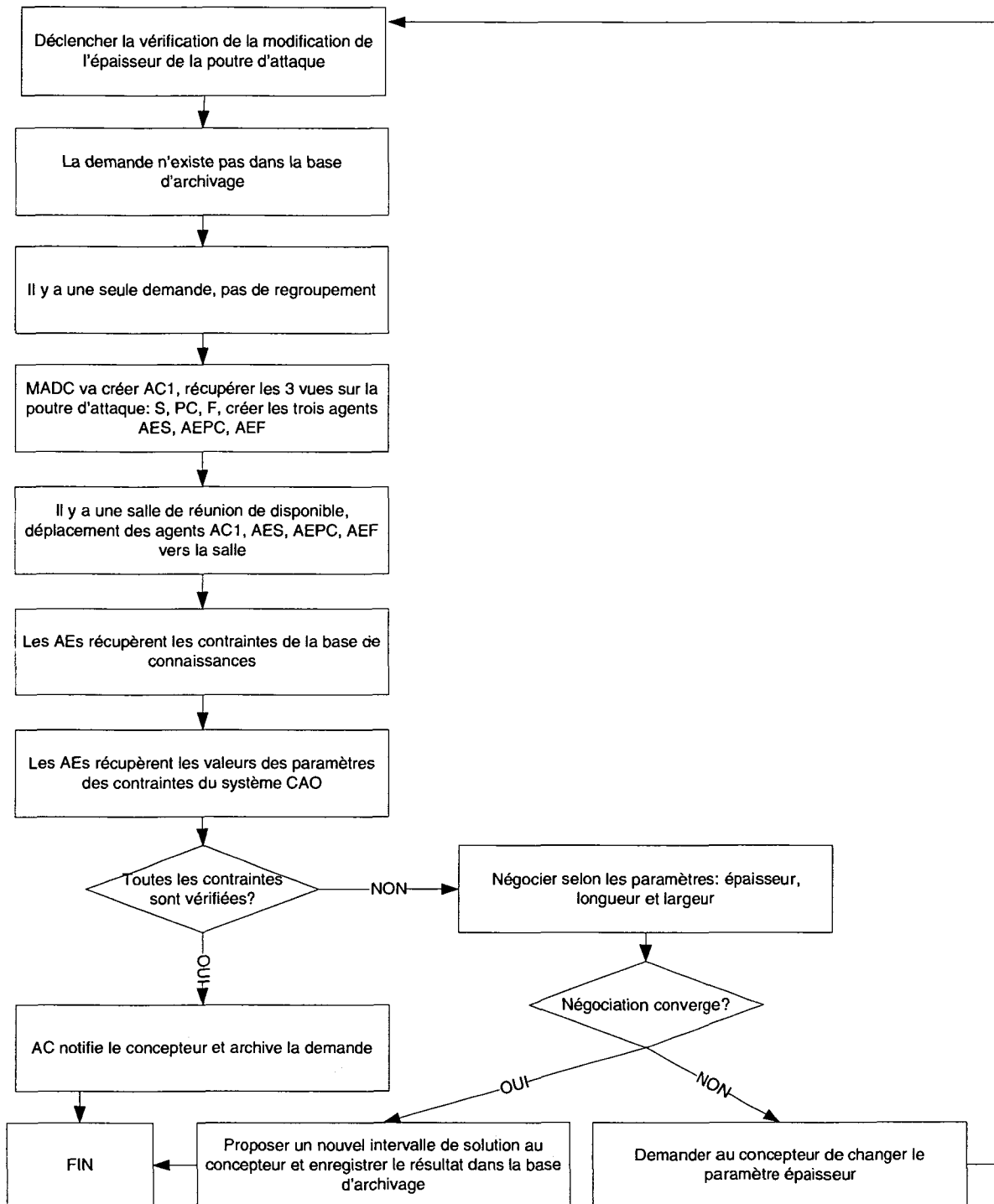


The screenshot shows a web browser window with the following content:

```
- <typemodeles xmlns="personal.xsd">
- <typemodele xmlns="">
  cockpit
  <typemodeles>cockpitgauche</typemodeles>
  <typemodeles>cockpitdroite</typemodeles>
</typemodele>
- <typemodele xmlns="">
  nez
  <typemodele>nezgauche</typemodele>
  <typemodele>nezdroite</typemodele>
</typemodele>
- <typemodeles xmlns="">
  aile
  <typemodeles>panneau superieur</typemodeles>
  <typemodeles>panneau inferieur</typemodeles>
  <typemodeles>poutre de bord de fuite</typemodeles>
  <typemodeles>nervure</typemodeles>
  <typemodeles>poutre de bord d attaque</typemodeles>
</typemodeles>
</typemodeles>
```

The browser interface includes a menu bar (Fichier, Edition, Affichage, Favoris, Outils), a toolbar with navigation and search icons, an address bar showing the local file path, and a search bar with Google and Recherche Web options. The status bar at the bottom indicates 'Terminé' and 'Poste de travail'.

## b) Algorithme de vérification





**c) Articles publiés**