



HAL
open science

L'introduction de la gestion du cycle de vie produit dans les entreprises de sous-traitance comme vecteur d'agilité opérationnelle et de maîtrise du produit

Muriel Milhaud Pinel

► To cite this version:

Muriel Milhaud Pinel. L'introduction de la gestion du cycle de vie produit dans les entreprises de sous-traitance comme vecteur d'agilité opérationnelle et de maîtrise du produit. Gestion et management. Université de Grenoble, 2013. Français. NNT : 2013GRENA009 . tel-00852852

HAL Id: tel-00852852

<https://theses.hal.science/tel-00852852>

Submitted on 21 Aug 2013

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **sciences pour l'ingénieur**

Arrêté ministériel : 7 août 2006

Présentée par

Muriel PINEL (née MILHAUD)

Thèse dirigée par **Christian BRAESCH** et **Laurent TABOUROT**

préparée au sein du **Laboratoire SYMME, SYstèmes et Matériaux pour la MEcatronique**
dans l'**École Doctorale SISEO, Sciences et Ingénierie des Systèmes de l'Environnement et des Organisations**

L'introduction de la gestion du cycle de vie produit dans les entreprises de sous-traitance comme vecteur d'agilité opérationnelle et de maîtrise du produit

Thèse soutenue publiquement le **30 mai 2013**,
devant le jury composé de :

Mlle Aline BERGER

Directrice Projet - Système d'Information à Thésame, Annecy, Membre

M. Christian BRAESCH

Maître de conférences Hors Classe à l'IAE Savoie Mont-Blanc, Directeur

M. Vincent CHAPURLAT

Professeur à l'École des Mines d'Alès, Rapporteur

M. Christophe MERLO

Maître de conférences HDR à l'ESTIA, Rapporteur

M. Lionel ROUCOULES

Professeur à l'ENSAM d'Aix en Provence, Président

M. Laurent TABOUROT

Professeur à l'École Polytech'Annecy-Chambéry, Codirecteur



Remerciements

« Nous ne sommes jamais au bout de nos surprises », cet adage s'est révélé on ne peut plus exact pour ma soutenance de thèse. J'y suis arrivée très stressée, et pourtant je l'ai vécu comme un moment très agréable. J'avais enfin réussi à structurer un support qui présentait de manière claire nos travaux et les échanges avec les membres du jury ont été riches et complémentaires. Je remercie tous ceux qui étaient là pour partager ce moment.

Ce moment, précieux à mes yeux, est le résultat de trois années pendant lesquelles de nombreuses personnes m'ont aidée dans le cadre de nos travaux. Je tiens à les remercier vivement.

Tout d'abord, je remercie mes encadrants Christian Braesch et Laurent Tabourot d'avoir eu confiance dans mon potentiel de chercheuse, d'avoir réussi à faire « taire » l'ingénieur en moi et à réveiller la future chercheuse profondément enfouie et d'avoir eu les bonnes clés qui m'ont permis d'accomplir la dernière ligne droite : rédiger le mémoire et préparer la soutenance. Je remercie Christian Braesch de m'avoir donné le virus de la modélisation, domaine qui résonne avec ma structure d'esprit cartésienne. Je remercie Laurent Tabourot d'avoir permis au PLM de devenir une thématique du laboratoire SYMME, thématique qui est en accord avec les problématiques industrielles actuelles et qui est passionnante.

Je remercie Aline Berger de co-animer avec moi le Club PLM dans la bonne humeur.

Je remercie le laboratoire SYMME pour son chaleureux accueil.

Je remercie les financeurs, Thésame, l'Assemblée des Pays de Savoie et le pôle de compétitivité Arve Industries, sans qui ces travaux n'auraient pu être réalisés.

Je remercie les rapporteurs et le président du jury pour leur intérêt et pour leurs retours éclairés sur nos travaux. Je remercie Lionel Roucoules et Christophe Merlo pour avoir partagé avec nous leur vision du PLM. Je remercie Vincent Chapurlat pour m'avoir encouragée à passer ma qualification à la fin de la soutenance.

Je remercie tous les « PLMistes » qui m'ont fait bénéficier de leur expérience : les membres industriels du Club PLM, Gauthier Chauvin et Lilia Gzara qui m'a initiée aux plaisirs de la recherche dans le domaine du PLM.

Je remercie les doctorants aux enthousiasmes débordants (mon « condisciple » Bertrand Nicquevert et ses bons conseils et Mehdi Iraqi-Houssaini et nos discussions passionnées sur l'IDM).

Je remercie les informaticiens qui m'ont aidée sans baisser les bras devant l'ampleur de mon ignorance (Maz (alias Patrick Mandallaz), Maxime Hugot et Marc Philippe Huget).

Je remercie Chloé et Loïc pour m'avoir aidée à surmonter les moments difficiles et les moments de doutes.

Et je remercie ma maman pour la relecture de ce mémoire et bien plus bien sûr.

Table des Matières

Glossaire

Introduction

Chapitre 1 : Formulation du problème et identification de deux objectifs

1	Introduction	23
2	Deux buts majeurs de l'entreprise, moyens pour les mettre en œuvre et frein rencontré....	24
2.1	<i>Introduction.....</i>	24
2.2	<i>Identification de deux buts majeurs pour l'entreprise.....</i>	24
2.2.1	Le triplet qualité coût délai	24
2.2.2	Trois défis de l'entreprise	26
2.3	<i>Les projets d'entreprise.....</i>	27
2.3.1	Définition et classification des projets d'entreprise.....	27
2.3.2	Deux perceptions de l'entreprise	28
2.3.3	Séparation des échelles et des langages.....	28
2.4	<i>Conclusion.....</i>	29
3	Le cycle de vie d'un produit	30
3.1	<i>Cycles de vie commerciaux et techniques.....</i>	30
3.2	<i>Les boucles retour dans le cycle de vie du produit.....</i>	31
3.3	<i>Les processus transversaux du cycle de vie.....</i>	33
3.4	<i>Les acteurs du cycle de vie : de l'entreprise à l'entreprise étendue</i>	33
3.5	<i>Conclusion.....</i>	34
4	Etat de l'art : l'approche Product Lifecycle Management (PLM) ou la gestion du cycle de vie produit	35
4.1	<i>L'approche PLM</i>	35
4.1.1	Les origines de l'approche PLM	35
4.1.2	PLM, un concept encore jeune	36
4.1.3	Les qualificatifs du PLM	37
4.1.4	Les buts de l'approche PLM.....	38
4.1.5	Des écarts entre le nom et les définitions associées	38
4.1.6	La définition de l'approche PLM choisie dans le cadre de nos travaux	39
4.1.7	Conclusion.....	39
4.2	<i>Les principes de l'approche PLM et la mise en œuvre du projet PLM</i>	39
4.2.1	Les principes de l'approche PLM.....	40
4.2.2	Le projet PLM : quels sont les éléments à considérer ?	41
4.2.3	Le projet PLM : les différentes étapes.....	42
4.2.4	Conclusion.....	42
4.3	<i>Bilan</i>	43

5	La création du Club PLM.....	43
6	La mise en œuvre d'un projet PLM est-elle devenue une nécessité pour les PME soustraitantes ?	44
7	Conclusion : formulation du problème et identification de deux objectifs	45

Chapitre 2 : Les bases de notre proposition

1	Introduction	49
2	Du besoin de modéliser au besoin de paradigmes	49
2.1	<i>La modélisation des produits.....</i>	<i>50</i>
2.2	<i>La mission du système PLM</i>	<i>51</i>
2.3	<i>Les langages et la métamodélisation.....</i>	<i>51</i>
2.4	<i>Une nouvelle définition du système PLM</i>	<i>54</i>
2.5	<i>La nécessité de paradigmes.....</i>	<i>55</i>
3	Les paradigmes à la base du système de représentation.....	56
3.1	<i>Le paradigme systémique</i>	<i>57</i>
3.2	<i>Le paradigme d'ambivalence</i>	<i>59</i>
3.3	<i>Modélisation du système PLM.....</i>	<i>60</i>
4	Les cadres de modélisation	61
4.1	<i>Les cadres de modélisation</i>	<i>61</i>
4.2	<i>La dynamique de passage entre les modèles</i>	<i>63</i>
4.2.1	<i>Utilisation d'un langage commun.....</i>	<i>64</i>
4.2.2	<i>Transformation de modèles</i>	<i>65</i>
4.3	<i>Le cadre de modélisation de notre proposition.....</i>	<i>66</i>
5	Conclusion.....	68

Chapitre 3 : Méthode d'élaboration du cahier des charges du système PLM

1	Introduction	71
2	Etat de l'art : de l'identification des besoins à l'élaboration du CDC.....	71
2.1	<i>Quels modèles pour définir les buts, les besoins et les contraintes ?.....</i>	<i>72</i>
2.2	<i>Quels langages pour modéliser les buts, les cas d'utilisation et les contraintes ?.....</i>	<i>75</i>
2.2.1	<i>Langage usuel de formalisation des buts.....</i>	<i>75</i>
2.2.2	<i>Langage usuel de formalisation des cas d'utilisation</i>	<i>76</i>
2.2.3	<i>Langage usuel de formalisation des exigences.....</i>	<i>77</i>
2.2.4	<i>Formalisation du cahier des charges.....</i>	<i>78</i>
2.2.5	<i>Autres langages existants</i>	<i>78</i>
2.3	<i>Quel processus pour élaborer le cahier des charges ?</i>	<i>78</i>
2.4	<i>Les difficultés rencontrées pour élaborer le CDC.....</i>	<i>79</i>

2.5	<i>Conclusion</i>	80
3	Les spécificités du système PLM et les difficultés rencontrées pour élaborer son CDC	80
3.1	<i>Spécificités liées au pôle d'observation organisationnel</i>	81
3.1.1	Spécificités relatives au périmètre du système PLM.....	81
3.1.2	Spécificités relatives à la définition du modèle métier.....	81
3.1.3	Conclusion.....	82
3.2	<i>Spécificités liées au pôle d'observation informationnel</i>	82
3.2.1	Spécificités liées à un logiciel.....	83
3.2.2	Spécificités du logiciel PLM.....	85
3.2.3	Conclusion.....	85
3.3	<i>Retours du Club PLM : confirmation des spécificités identifiées et mise en lumière de difficultés propres à l'élaboration du CDC du système PLM</i>	85
3.4	<i>Conclusion</i>	87
4	La définition de niveaux de maturité	87
4.1	<i>Maturité relative à l'organisation de l'entreprise et à ses processus métiers</i>	87
4.2	<i>Maturité relative aux compétences nécessaires pour définir le futur système PLM</i>	87
4.3	<i>Etat de l'art des modèles de maturité PLM existants</i>	88
5	Vers une méthode d'élaboration du CDC du système PLM	89
5.1	<i>Définition générique des besoins</i>	89
5.2	<i>Un processus adapté à l'élaboration du CDC du système PLM</i>	91
	Utilité des modèles de maturité et des définitions génériques des besoins.....	92
5.3	<i>Conclusion</i>	95
6	Conclusion	95

Chapitre 4 : Cadre de modélisation pour la mise en œuvre du système PLM

1	Introduction	97
2	Mise en oeuvre du système PLM : formulation de deux problèmes	97
2.1	<i>Formulation d'un premier problème : modèles utilisés pour mettre en œuvre le système PLM</i>	98
2.1.1	Le modèle personnalisé du logiciel PLM.....	98
2.1.2	Les modèles utilisés pour passer de l'expression du besoin à la personnalisation du logiciel PLM.....	99
2.1.3	Etude des usages dans le Club PLM.....	99
2.1.4	Reformulation des problèmes posés.....	100
2.2	<i>Formulation d'un second problème : cohérence et traçabilité entre les différentes représentations du produit</i>	101
2.3	<i>Conclusion</i>	102
3	Etat de l'art : les méthodes de mise en oeuvre du système PLM	103
3.1	<i>Méthodes pour définir les modèles du système PLM</i>	103
3.1.1	Les différentes méthodes.....	103

3.1.2	L'utilisation d'un unique modèle pour créer les différentes représentations du produit dans le système PLM.....	104
3.2	<i>Les mécanismes assurant les interfaces entre le système PLM et d'autres systèmes.....</i>	105
3.3	<i>Conclusions vis à vis du problème concernant les modèles utilisés pour la mise en œuvre du système PLM.....</i>	106
3.4	<i>Conclusions vis à vis du problème de cohérence et de traçabilité entre les différentes représentations du produit</i>	107
3.5	<i>Conclusion.....</i>	107
4	Notre proposition : un cadre de modélisation basé sur le paradigme d'ambivalence.....	108
4.1	<i>Les axes du cadre de modélisation.....</i>	108
4.1.1	Le premier axe.....	109
4.1.2	Le deuxième axe.....	110
4.1.3	Le troisième axe	111
4.2	<i>Positionnement des deux problèmes dans notre cadre.....</i>	113
4.3	<i>Le paradigme d'ambivalence</i>	113
4.4	<i>Utilisation du cadre de modélisation</i>	116
4.5	<i>Conclusion.....</i>	117
5	Cadre de modélisation et mise en oeuvre du système PLM (Problème 1).....	118
5.1	<i>Rappel du problème.....</i>	118
5.2	<i>Comment utiliser le cadre pour répondre aux problèmes identifiés</i>	119
5.2.1	Les métamodèles PLM	119
5.2.2	Transformation entre modèle métier PLM et modèle personnalisé du logiciel PLM .	119
5.2.3	Le métamodèle de personnalisation du logiciel PLM	119
5.3	<i>Conclusion.....</i>	120
6	Cadre de modélisation et cohérence entre les différentes représentations du produit (Problème 2).....	120
6.1	<i>Rappel du problème.....</i>	121
6.2	<i>Les représentations du produit de conception et pour fabrication.....</i>	121
6.3	<i>Réponse aux problèmes D et E : mise en œuvre de références croisées</i>	122
6.4	<i>Définition de règles de cohérence</i>	123
6.4.1	Nos réponses aux problèmes F et G	123
6.4.2	Formalisation des règles de cohérence	124
6.4.3	Exemples de mises en œuvre de règles de cohérence.....	125
6.5	<i>Conclusion.....</i>	125
7	Conclusion.....	126
Chapitre 5 - Expérimentation		
1	Introduction	129
2	Sujet d'étude : métamodèles, modèles et transformations.....	129

3	Choix d'un environnement de modélisation pour la mise en œuvre du cadre.....	131
3.1	<i>Choix d'un environnement de modélisation</i>	<i>131</i>
3.2	<i>L'environnement EMF</i>	<i>132</i>
3.2.1	<i>Modélisation dans EMF</i>	<i>132</i>
3.2.2	<i>Transformation de modèles avec EMF.....</i>	<i>135</i>
3.3	<i>Conclusion.....</i>	<i>137</i>
4	Sujet d'étude : nature des métamodèles, modèles et transformations.....	137
5	Définition du métamodèle du modèle métier PLM et exemple d'instanciation dans EMF	138
5.1	<i>1^{ère} étape : étude terrain des modèles métiers de membres du Club PLM</i>	<i>138</i>
5.1.1	<i>Objectif de l'étude</i>	<i>138</i>
5.1.2	<i>Résultats de l'étude.....</i>	<i>138</i>
5.1.3	<i>Conclusions de l'étude</i>	<i>141</i>
5.2	<i>2^{ème} étape : identification des entités du métamodèle métier PLM.....</i>	<i>141</i>
5.3	<i>3^{ème} étape : identification de la structure de notre métamodèle du modèle métier PLM et mise en œuvre dans EMF</i>	<i>144</i>
5.3.1	<i>Structure des entités principales</i>	<i>144</i>
5.3.2	<i>Gestion des versions avec ou sans élément master.....</i>	<i>146</i>
5.3.3	<i>Structuration des « éléments ».....</i>	<i>148</i>
5.3.4	<i>Conclusion.....</i>	<i>148</i>
5.4	<i>Création d'un modèle métier à partir du métamodèle métier proposé dans EMF.....</i>	<i>148</i>
5.4.1	<i>Création de modèles à partir d'un métamodèle dans EMF.....</i>	<i>149</i>
5.4.2	<i>Exemple d'instanciation du métamodèle métier</i>	<i>154</i>
5.4.3	<i>Conclusion.....</i>	<i>156</i>
5.5	<i>Conclusion.....</i>	<i>156</i>
6	Définition du métamodèle du modèle personnalisé d'un logiciel PLM dans EMF	156
6.1	<i>Définition du métamodèle de personnalisation d'Audros dans EMF</i>	<i>156</i>
6.2	<i>Conclusion.....</i>	<i>161</i>
7	Conclusion.....	161

Conclusion et perspectives

Conclusion.....	165
------------------------	------------

Perspectives.....	166
--------------------------	------------

<i>Perspectives complétant les propositions de ces travaux.....</i>	<i>166</i>
<i>Perspectives de travaux « conceptuels »</i>	<i>166</i>
<i>Perspectives de travaux relatifs à l'utilisation d'un environnement de modélisation.....</i>	<i>166</i>
<i>Perspectives fondées sur de nouvelles hypothèses</i>	<i>167</i>

Références bibliographiques

Liste des Figures

Chapitre 1

Figure 1.1 : Cycle de vie commerciale du produit [Kotler, 96]	30
Figure 1.2 : Cycle de vie du produit (adapté de [Terzi et al., 07])	31
Figure 1.3 : Exemple du cycle de vie d'un produit	32
Figure 1.4 : Exemple du cycle de vie d'un produit - suite	32
Figure 1.5 : Les processus globaux liés au cycle de vie du produit (adapté de [Terzi et al., 07])	33
Figure 1.6 : L'entreprise étendue [Capraro et al., 02]	34
Figure 1.7: Nombre de résultats à la requête « product lifecycle management » sur le site Science Direct	37
Figure 1.8 : Buts stratégique, tactique et opérationnel de l'approche PLM [Gautrot et al., 10]	39

Chapitre 2

Figure 2.1 : Différentes représentations du produit en fonction des acteurs	50
Figure 2.2 : Mécanisme d'instanciation entre métamodèle, modèle et représentation	53
Figure 2.3 : Les quatre niveaux de modélisation préconisés par l'OMG (adapté de [Huget, 10])	54
Figure 2.4 : Quels modèles pour représenter une classe de produits ou le système PLM adapté de [Genelot, 01]	56
Figure 2.5 : Relations entre classe de situations, paradigme et modèles (adapté du systémographe de Jean-Louis Le Moigne [Le Moigne, 90])	56
Figure 2.6 : Représentation d'un système [Le Moigne, 90]	58
Figure 2.7 : Différences entre les raisonnements réactif et synergique (adapté de [Chelli, 03])	60
Figure 2.8 : Les deux axes du cadre de Merise (adapté de [Gabay, 98])	62
Figure 2.9 : Le cadre de modélisation CIMOSA [Kosanke, 99]	63
Figure 2.10 : Transformations de modèles M2M et M2T	65
Figure 2.11 : Axe d'étude basé sur le cycle de vie du produit	66
Figure 2.12 : Le cycle de vie du système PLM	67
Figure 2.13 : Cadre combinant les axes « cycle de vie produit » et « cycle de vie système PLM »	67
Figure 2.14 : Cadre de modélisation de notre proposition	68

Chapitre 3

Figure 3.1 : Formalisation des buts, des besoins et des contraintes (adapté de [Wieggers, 03])	74
Figure 3.2 : Diagramme de cas d'utilisation " Parcourir en descendant les données relatives à la structure d'un produit "	77
Figure 3.3 : Processus d'élaboration du Cahier des Charges (CDC) d'un système d'information [Constantinidis, 10]	79
Figure 3.4 : Extrait du modèle métier de la société Duqueine mis en oeuvre au moyen de la personnalisation de leur logiciel PLM (Témoignage Journée du PLM 2008)	81
Figure 3.5 : Evaluation des améliorations amenées par le projet PLM dans une entreprise du Club PLM	88
Figure 3.6 : Processus d'élaboration du CDC d'un système PLM et étapes préliminaires	92
Figure 3.7 : Positionnement des questionnaires d'évaluation de la maturité et du modèle générique des besoins relatifs au système PLM dans le processus défini	94

Chapitre 4

Figure 4.1 : Extrait du modèle métier PLM de la société Duqueine	98
Figure 4.2 : Logiciel PLM : relations d'instanciation entre métamodèle, modèle et représentation du produit	99
Figure 4.3 : Modèle d'exigences PLM, modèle métier PLM et modèle personnalisé du logiciel PLM	100
Figure 4.4 : Exemple de déformations des exigences décrivant un produit	101

<i>Figure 4.5 : Les déformations dues à des erreurs d'interprétation et à des erreurs de transformation</i>	102
<i>Figure 4.6 : Extrait du modèle générique métier du Système d'Information Produit de [Gzara, 00]</i>	104
<i>Figure 4.7 : Utilisation d'un unique modèle (adapté de [Lombard, 06])</i>	105
<i>Figure 4.8 : Utilisation d'un logiciel unificateur</i>	106
<i>Figure 4.9 : Utilisation d'une architecture fédérée</i>	106
<i>Figure 4.10 : Cadre de modélisation proposé pour supporter la mise en œuvre d'un système PLM</i>	109
<i>Figure 4.11 : Premier axe du cadre proposé : les interfaces entre les différents logiciels</i>	110
<i>Figure 4.12 : Deuxième axe du cadre proposé : les différents modèles des systèmes</i>	111
<i>Figure 4.13 : Troisième axe du cadre proposé : les modèles et leurs instances (représentations produit)</i>	112
<i>Figure 4.14 : Les quatre niveaux de modélisation concernant la mise en œuvre d'un logiciel PLM</i>	112
<i>Figure 4.15 : Notre cadre de modélisation : identification des deux problèmes</i>	113
<i>Figure 4.16 : Mise en œuvre du Paradigme d'Ambivalence (PA) sur les deux axes identifiés</i>	115
<i>Figure 4.17 : Notre cadre de modélisation : identification des correspondances préconisées pour la mise en œuvre du paradigme d'ambivalence</i>	116
<i>Figure 4.18 : Notre cadre de modélisation : identification des mécanismes de transformations et de vérification</i>	117
<i>Figure 4.19 : Positionnement du cas d'étude dans le cadre de modélisation</i>	122
<i>Figure 4.20 : Exemple de mise en œuvre de références croisées</i>	123
 Chapitre 5	
<i>Figure 5.1 : Positionnement du sujet d'étude dans le cadre de modélisation</i>	130
<i>Figure 5.2 : Métamodèles, modèles et transformations de notre sujet d'étude</i>	131
<i>Figure 5.3 : Ecore comme méta-métamodèle ou comme métamodèle (adapté de [Vanwormhoudt, 12])</i>	133
<i>Figure 5.4 : Extrait des entités de Ecore [Steinberg, 08]</i>	133
<i>Figure 5.5 : Différences de formalisation d'une relation entre deux classes avec le MOF et avec Ecore</i>	134
<i>Figure 5.6 : Forme arborescente et forme graphique dans EMF</i>	134
<i>Figure 5.7 : Modélisation d'une relation n-aire avec EMF</i>	134
<i>Figure 5.8 : Importations et générations possibles à partir d'Ecore [Steinberg, 07]</i>	135
<i>Figure 5.9 : Exemple de programme ATL</i>	136
<i>Figure 5.10 : Exemple de mise en œuvre d'une transformation ATL</i>	136
<i>Figure 5.11 : Nature des métamodèles, modèles et transformations de notre sujet d'étude</i>	138
<i>Figure 5.12 : Les trois manières de gérer les évolutions avec un système PLM</i>	141
<i>Figure 5.13 : Relation n-aire entre l'entité « article » et l'entité « relation »</i>	142
<i>Figure 5.14 : Les entités article, association et relation de notre métamodèle métier</i>	143
<i>Figure 5.15 : Métamodèle métier de données proposé (produit mécatronique)</i>	145
<i>Figure 5.16 : Exemple de gestion des versions sans élément master</i>	146
<i>Figure 5.17 : Exemple de gestion des versions avec un élément master</i>	147
<i>Figure 5.18 : Relations entre des entités « relation », « association », « élément » et « fichier »</i>	148
<i>Figure 5.19 : Exemple d'instanciation d'un métamodèle avec EMF – première action</i>	149
<i>Figure 5.20 : Exemple d'instanciation d'un métamodèle avec EMF – instanciation d'une EClass</i>	150
<i>Figure 5.21 : Exemple d'instanciation d'un métamodèle avec EMF – création d'une structure par les EReferences</i>	151
<i>Figure 5.22 : Exemple d'instanciation d'un métamodèle avec EMF – menu contextuel</i>	152
<i>Figure 5.23 : Exemple d'instanciation d'un métamodèle avec EMF – plusieurs valeurs pour une propriété</i>	153
<i>Figure 5.24 : Extrait du modèle métier de la société Duqueine mis en oeuvre au moyen de la personnalisation de leur logiciel PLM (Témoignage Journée du PLM 2008)</i>	154
<i>Figure 5.25 : Extrait de modèle métier de la société Duqueine créé à partir du métamodèle métier proposé</i>	155

<i>Figure 5.26 : Création du métamodèle de données de personnalisation du logiciel PLM Ecore à partir du métamodèle relationnel</i>	157
<i>Figure 5.27 : Transformations dont le métamodèle 4 fait partie</i>	158
<i>Figure 5.28 : Extrait du métamodèle 4 initial avec deux EClass</i>	159
<i>Figure 5.29 : Instanciation du métamodèle de la figure 5.28</i>	159
<i>Figure 5.30 : Fichier XMI correspondant au modèle de la figure 5.29</i>	160
<i>Figure 5.31 : Métamodèle de la figure 5.28 amélioré</i>	160
<i>Figure 5.32 : Instanciation du métamodèle de la figure 5.31</i>	160
<i>Figure 5.33 : Fichier XMI correspondant au modèle de la figure 5.32</i>	161

Conclusion

<i>Figure 6.1 : Buts stratégiques, tactiques et opérationnels des projets Lean et PLM [Gautrot et al., 2010]</i>	168
--	-----

Liste des Tableaux

Chapitre 1

Tableau 1.1 : Classification des projets d'entreprise [Arto et al., 05] 27

Chapitre 2

Tableau 2.1 : Définitions des niveaux de visibilité et des représentations du produit 51

Tableau 2.2 : Définitions de la modélisation, du modèle et de la représentation 53

Tableau 2.3 : Définitions des représentations du produit 55

Tableau 2.4 : Définitions du modèle métier PLM et du modèle personnalisé du logiciel PLM 61

Chapitre 3

Tableau 3.1 : Exemple de décomposition d'un but [Zancul, 09] 76

Tableau 3.2 : Acteurs du cas d'utilisation "Parcourir en descendant les données relatives à la structure d'un produit" 76

Tableau 3.3 : Buts identifiés par [Zancul, 09] et [Le Duigou, 10] 90

Chapitre 5

Tableau 5.1 : Correspondances entre les entités du métamodèle proposé et celles du modèle de [Gzara, 00] 144

Glossaire

Agilité opérationnelle (cf. chapitre 1) : capacité de l'entreprise à se reconfigurer pour s'adapter aux sollicitations toujours nouvelles des tiers dont dépendent son développement, sa rentabilité et son fonctionnement [Chelli, 03].

Approche PLM (Product Lifecycle Management) (cf. chapitre 1) : cadre de référence identifiant des buts, des principes, des méthodes et des outils adéquats pour mettre en œuvre un management centré sur les produits, sur la collaboration et sur les savoirs faire.

Cadre de modélisation (cf. chapitre 2) : structure qui positionne les différents modèles proposés par une méthode suivant des axes d'étude définis, appelés aussi dimensions.

Langage de modélisation (cf. chapitre 2) : ensemble de symboles (vocabulaire), de règles de composition des symboles (syntaxe) et de sens attribués aux symboles et aux composition de symboles (sémantique). Un langage de modélisation est utilisé pour construire un modèle.

Métamodèle (cf. chapitre 2) : représentation d'un langage qui identifie les différents symboles du langage (vocabulaire) et les règles d'organisation de ces symboles (syntaxe). Le métamodèle ne fournit pas d'indication sur la sémantique.

Modélisation (cf. chapitre 2) : action de construction intentionnelle d'un modèle d'une classe de produits ou d'un modèle du système PLM.

Modèle (cf. chapitre 2) : représentation intelligible, artificielle et partielle d'une classe de produits ou du système PLM à un instant donné. Par définition, un modèle est le résultat de l'instanciation d'un métamodèle.

Modèle métier PLM (cf. chapitre 2) : modèle représentant le pôle d'observation organisationnel lié à la gestion du cycle de vie du produit.

Modèle personnalisé du logiciel PLM (cf. chapitre 2) : modèle représentant le pôle d'observation informationnel lié à l'utilisation d'un logiciel PLM.

Niveau de visibilité (cf. chapitre 2) : ensemble des représentations qu'un acteur a du produit. Il est défini en fonction des actions que l'acteur doit réaliser sur le produit.

Paradigme (cf. chapitre 2) : conception théorique dominante ayant cours à une certaine époque dans une communauté scientifique donnée, qui fonde les types d'explication envisageables et les types de faits à découvrir dans une science donnée [CNRTL, 13]. Par extension, nous identifions un paradigme comme une manière de percevoir le monde.

Paradigme d'ambivalence (cf. chapitre 2) : perception d'une réalité ambivalente issue de plusieurs pôles d'observation. Henri Chelli définit ainsi l'entreprise comme une réalité ambivalente qui est à la fois « organisationnelle » et « informationnelle ». L'ambivalence d'une réalité implique que les perceptions correspondant à chaque pôle d'observation ne doivent pas être juxtaposées mais considérées de manière complémentaire et corrélée [Chelli, 03].

Perception informationnelle (cf. chapitre 1) : perception qui correspond au pôle d'observation informationnel de la réalité opérationnelle de l'entreprise. Le pôle d'observation informationnel considère les services automatisés qui manipulent les informations pour fournir les fonctions [Chelli, 03].

Perception organisationnelle (cf. chapitre 1) : perception qui correspond au pôle d'observation organisationnel de la réalité opérationnelle de l'entreprise. Le pôle d'observation organisationnel considère les hommes et les processus qui fournissent les produits et les services [Chelli, 03].

Représentation ou instance (cf. chapitre 2) : représentation intelligible, artificielle et partielle d'un produit particulier à un instant donné. Par définition, une représentation est le résultat de l'instanciation d'un modèle.

Système PLM (cf. chapitre 1) : système d'information supportant la création, la circulation, l'utilisation et l'évolution du patrimoine informationnel de représentation du produit. D'une part, ce patrimoine inclut les informations qui déterminent les exigences et les spécifications du produit et celles qui définissent comment le produit doit être fabriqué, utilisé, maintenu et recyclé. D'autre part, il inclut les informations qui décrivent le produit physique dans différents états (utilisé, fabriqué, maintenu, recyclé). Le support informatique de ce dispositif organisationnel est souvent réalisé au moyen d'un logiciel PLM. *Définition adaptée de celle du Système d'Information Produit ou SIP de Lilia Gzara [Gzara, 00].*

Introduction

Ce projet de thèse, qui s'est déroulé d'octobre 2009 à avril 2013, est le résultat de la conjonction de trois éléments.

Le premier élément est un facteur déclencheur issu d'un contexte. Pour ces travaux, le facteur déclencheur est un besoin venant des entreprises industrielles locales. Ces entreprises doivent faire face à des évolutions fréquentes et nombreuses de leur environnement. Certaines de ces évolutions constituent des sources d'opportunité comme les technologies de l'information. D'autres constituent de nouvelles contraintes. D'une part, les exigences réglementaires comme les exigences de traçabilité des produits impliquent un périmètre d'action de plus en plus grand (traçabilité des produits, des substances, etc.) [Grieves, 06]. D'autre part, les exigences venant des clients sont de plus en plus importantes. De nos jours, il n'est plus possible d'adopter une stratégie de domination par les coûts face à la concurrence des pays « low cost ». Ainsi, l'intégration d'innovations majeures relatives au produit ou au procédé de fabrication associé est devenue un facteur important de différenciation aux yeux du client [Bonjour, 06] [Vermette, 91]. L'étape de développement du couple « produit – procédé » est ainsi identifiée comme une étape essentielle. Pour les industriels locaux du Club PLM¹, la construction d'un **système PLM*** fait partie des éléments de réponse pour faire face aux exigences de traçabilité et au besoin d'intégration d'innovations. En effet, ce système supporte la création, la circulation, l'utilisation et l'évolution des informations relatives à la définition des produits. Il permet ainsi d'une part d'assurer la traçabilité des évolutions des produits tout au long de leur vie et d'autre part de développer la robustesse de l'étape de développement. La mise en œuvre du système PLM est souvent réalisée dans le cadre d'un projet d'entreprise. L'adoption de ce qui est alors appelé une « **approche** » PLM*, ou approche de gestion du cycle de vie produit, traduit la volonté de l'entreprise de se recentrer sur ses produits et de favoriser la collaboration entre les services et avec ses partenaires. Pour favoriser l'adoption d'une telle approche, le pôle de compétitivité Arve Industries², l'Assemblée des Pays de Savoie³ et Thésame⁴ ont financé ce projet de thèse pour explorer la question : « un projet d'entreprise relatif à la gestion du cycle de vie produit peut-il être vecteur de performance pour les entreprises de sous-traitance ? ».

Le deuxième élément nécessaire à un projet de thèse est la mise en œuvre de compétences (celles du doctorant et de ses encadrants) dans un environnement de recherche. Le PLM peut être abordé selon plusieurs pôles d'observation : stratégique, technique, métier. Les travaux actuels sur le PLM focalisent ainsi sur un problème précis : innovation et PLM [Merminod, 07], interopérabilité et PLM [Pavot, 10], ou bien ils ont pour sujet la mise en œuvre de l'approche PLM. Dans ce cas, le spectre thématique est large : métier, informatique, modélisation... (par exemple, les travaux de Julien Le Duigou [Le Duigou, 10]). Nous avons choisi de positionner nos travaux dans cette deuxième catégorie. En effet, nous avons la volonté de créer un projet pluridisciplinaire qui mette à profit les compétences de Christian Braesch en Système d'Information et en modélisation, de Laurent Tabourot en PLM et de moi-même en industrialisation (de profil mécanicien, j'ai été ingénieur méthodes pendant cinq ans dans l'industrie).

Le **troisième élément** nécessaire à un projet de thèse est l'identification d'un point de départ, c'est à dire la formulation d'un **problème** de recherche et d'une solution potentielle pour y répondre. L'élaboration de cette solution devient alors l'**objectif** à atteindre. Pour justifier le plan de ces travaux de thèse, rappelons quelques principes de la recherche scientifique. Il y a une activité de recherche lorsqu'il existe un point de départ connu et un point d'arrivée non défini. Il faut alors produire un

¹ Créé en mars 2010, le Club PLM est constitué de représentants d'entreprises utilisatrices de logiciels PLM et de personnes du laboratoire SYMME, du laboratoire G-SCOP et de Thésame.

² Le pôle Arve Industries (du décolletage à la mécatronique) a six domaines d'activités stratégiques dont la qualité des produits, le tolérancement, la performance industrielle et la conception collaborative.

³ L'Assemblée des Pays de Savoie regroupe les conseillers généraux de Savoie et Haute-Savoie.

⁴ Thésame, centre de ressources en mécatronique, gestion industrielle et management de l'innovation.

travail itératif qui, à partir d'un problème, recense les hypothèses, fixe les objectifs et met en œuvre un processus de résolution pour atteindre l'objectif. Les aspects exploratoires et d'apprentissage de l'étude font que le problème et l'objectif identifiés initialement peuvent être à tout instant remis en cause. L'étude est ainsi constituée de différentes étapes. A la fin de chaque étape, le problème est réexaminé, précisé et reformulé, puis un nouvel objectif est défini. Le travail de recherche constitue également une exploration non linéaire. En effet, il fournit continuellement des enseignements qui permettent de mieux appréhender l'essence du problème et la variété de solutions. Ainsi, le mémoire présente le fruit de nos réflexions aujourd'hui. Il constitue une analyse ponctuelle dans laquelle nous identifions un problème, une solution et un protocole de recherche scientifique en nous fondant sur nos connaissances actuelles. Ces connaissances sont le résultat d'activités académiques, mais également le résultat d'analyses menées dans le cadre d'un groupe constitué de chefs de projet PLM industriels et d'universitaires, le Club PLM.

Le **problème** formulé au début des trois ans était : « un projet d'entreprise relatif à la gestion du cycle de vie produit peut-il être vecteur de performance pour les entreprises de sous-traitance ? ». Aujourd'hui, le problème que nous identifions plus précisément est « comment construire un système PLM en adéquation avec les buts de l'entreprise ? » Deux buts nous intéressent plus particulièrement. Le premier but consiste à avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût. Le deuxième but consiste à développer l'**agilité opérationnelle***.

Une méthode d'analyse d'un système d'information a pour objectif de préconiser un ensemble de modèles et un processus adaptés à l'étude à réaliser [Rieu, 99]. Ainsi, la **solution** proposée dans ces travaux pour construire un système PLM est d'utiliser un **cadre de modélisation*** adapté. Le cadre proposé est basé sur le paradigme systémique et sur le **paradigme d'ambivalence*** défini par Henri Chelli [Chelli, 03]. Il permet de raisonner de manière conjointe sur le système PLM (au travers des modèles qui lui sont relatifs) et sur le produit (au travers de ses représentations). Ce cadre guide le modélisateur dans la construction des modèles qui lui sont nécessaires pour définir le système PLM. Il guide également l'identification de mécanismes de cohérence entre les différents modèles relatifs au système PLM d'une part et entre les différentes représentations du produit utilisées tout au long de son cycle de vie d'autre part. Enfin, il supporte l'adoption d'un raisonnement synergique entre les différentes familles d'acteurs de l'entreprise. D'une part, l'adoption de ce raisonnement par les responsables métier et par les techniciens de l'information permet de développer l'agilité opérationnelle recherchée. D'autre part, l'adoption de ce raisonnement par les responsables métiers des différentes étapes du cycle de vie du produit est source d'amélioration pour le triplet coût - qualité - délai du produit.

Pour réaliser une construction efficace du système PLM, il faut comprendre pourquoi ce système est nécessaire, c'est-à-dire le but de sa construction. Tout but s'inscrit dans une hiérarchie de buts [van Lamsweerde, 00]. Ainsi, un but est à la fois un « pourquoi » pour les sous-buts qui vont en dériver et un « comment » pour le sur-but dont il est issu. L'usage consiste à classifier les buts en trois catégories : stratégique (qui correspond à un horizon long terme), tactique (qui correspond à un horizon moyen terme) et opérationnel (qui correspond à un horizon court terme). Ainsi, dans le **chapitre 1**, le contexte de l'entreprise est présenté puis un état de l'art des travaux relatifs à l'approche PLM est réalisé. Ceci nous permet d'identifier et confronter les buts de l'entreprise avec ceux associés à l'approche PLM. Ceci nous permet également d'identifier dans quelle hiérarchie de buts s'inscrit la construction du système PLM. Deux buts de l'entreprise sont mis en avant. Le premier consiste à avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût. Le deuxième consiste à développer l'agilité opérationnelle. Un problème est alors défini : la mise en œuvre d'un système PLM doit être faite en adéquation avec ces deux buts. Une hypothèse est alors formulée : il faut une méthode de mise en œuvre du système PLM adaptée à la prise en compte des deux buts de l'entreprise. Une méthode de construction d'un système d'information préconise un ensemble de modèles et un processus appropriés. Ainsi, notre premier objectif est d'identifier les concepts de modélisation qui seront à la base de notre proposition.

De ce fait, le **chapitre 2** est un préambule qui présente les différents concepts de modélisation utilisés tout au long des chapitres suivants. A la fin de ce chapitre, une première ébauche du cadre de modélisation, qui constitue la proposition des travaux de ce mémoire, est définie. L'élaboration du

cahier des charges constituant la première tâche de la construction du système PLM, le chapitre suivant a alors pour objectif de proposer une méthode pour élaborer ce cahier des charges.

L'élaboration du cahier des charges (ou CDC) d'un système d'information est maintenant reconnue comme étant une étape très importante de son cycle de vie qui conditionne toutes les étapes ultérieures. En effet, une identification des besoins et des contraintes incomplète ou erronée conduit à l'obtention de systèmes ne répondant pas aux attentes du client. L'objectif du **chapitre 3** consiste ainsi à définir une méthode adaptée à l'élaboration du cahier des charges d'un système PLM. La méthode proposée est définie à partir d'un état de l'art des travaux relatifs à l'ingénierie des besoins, d'un état de l'art des travaux relatifs aux exigences des systèmes PLM et de conclusions tirées des expériences de chefs de projet PLM industriels.

Dans le **chapitre 4**, le cadre de modélisation qui constitue notre proposition est défini de manière détaillée et il est positionné par rapport aux travaux PLM existants. Tout au long du chapitre, la construction du cadre est réalisée en se basant sur le paradigme systémique et sur le paradigme d'ambivalence. Pour terminer, des exemples illustrent comment appliquer ce cadre pour les problèmes identifiés (cohérence entre les modèles du système PLM et cohérence entre les représentations du produit).

Pour finir notre étude, une mise en œuvre opérationnelle du cadre est réalisée dans le **chapitre 5**. L'objectif de cette expérimentation n'est pas de valider le cadre mais de faire une étude de faisabilité de sa mise en œuvre avec un environnement de modélisation existant. En effet, valider le cadre signifierait être capable de prouver qu'il constitue un outil de raisonnement adapté pour toutes les entreprises faisant des produits mécatroniques et souhaitant construire un système PLM. Valider le cadre signifierait également être capable de prouver qu'il a une valeur ajoutée par rapport aux méthodes proposées dans les travaux existants. Ne pouvant amener ces preuves, nous avons fait le choix d'illustrer comment l'instrumenter.

Chapitre 1

Formulation du problème et identification de deux objectifs

Sommaire

- 1 Introduction
 - 2 Deux buts majeurs de l'entreprise, moyens pour les mettre en œuvre et frein rencontré
 - 3 Le cycle de vie d'un produit
 - 4 Etat de l'art : l'approche Product Lifecycle Management (PLM) ou la gestion du cycle de vie produit
 - 5 La création du Club PLM
 - 6 La mise en œuvre d'un projet PLM est-il devenu une nécessité pour les PME sous-traitantes ?
 - 7 Conclusion : formulation du problème et identification de deux objectifs
-

1 Introduction

De nos jours, l'entreprise doit faire face à des évolutions de plus en plus rapides de son environnement. Ces évolutions sont liées à la mondialisation du marché, au changement rapide de la demande, à la transformation de l'environnement social, à la disponibilité de techniques et de technologies toujours plus performantes, etc. Ces évolutions favorisent entre autres des réorganisations continues, la recherche de nouveaux partenariats et la fusion de différentes structures. Dans cet environnement changeant, deux buts majeurs de l'entreprise restent constants. Le premier est la préservation, voire l'amélioration de l'**agilité opérationnelle** [Sherehiy et al., 07] [Zhang, 11]. Cette agilité permet à l'entreprise de s'adapter à son environnement et d'être proactive en étant capable de mettre en œuvre l'innovation nécessaire pour rester concurrentielle. Le deuxième but est la mise sur le bon marché du bon produit au bon moment et avec le bon coût. Ce but est atteint par des améliorations régulières sur la qualité, le coût et le délai tout au long du cycle de vie des produits. Ce triplet « qualité-coût-délai » représente en effet les indicateurs clés que l'entreprise utilise pour s'implanter sur de nouveaux marchés ou simplement pour conserver sa position sur des marchés existants.

L'entreprise développe ainsi différents projets d'entreprise pour atteindre ces deux buts. Ces travaux ont trait plus particulièrement au projet d'entreprise qui a pour objectif de gérer et maîtriser le cycle de vie du produit, i.e. le **projet PLM** (Product Lifecycle Management). Ce type de projet concerne différents acteurs qui ont tous leur propre perception de l'entreprise. Ces différentes perceptions rendent souvent difficile la mise en œuvre de l'agilité opérationnelle [Chelli, 03]. L'objectif de ce chapitre est de décrire clairement le contexte de l'entreprise et les enjeux du projet PLM pour formuler de façon précise notre problème.

Dans un premier temps, ce chapitre rappelle le contexte de l'entreprise manufacturière et sa mission principale (cf. §2). Il décrit ensuite quelques spécificités des projets d'entreprise. Puis, il met en lumière le fait que la différence de perceptions peut être un frein à l'agilité opérationnelle. Suite à la description du contexte de ces travaux, il convient de présenter d'une part les différents concepts du cycle de vie d'un produit (cf. §3) et, d'autre part, un état de l'art de l'approche PLM (cf. §4). Le champ d'application de ces travaux étant les PME sous-traitantes, ce chapitre présente le Club PLM, groupe de travail composé de chefs de projet PLM industriels et d'universitaires, qui nous a aidé à identifier les facteurs potentiels pouvant déclencher un projet PLM dans ce type d'entreprise (cf. §5 et §6). Ce chapitre se conclut par une formulation claire du problème et de l'hypothèse qui seront traités dans la suite de ce mémoire.

2 Deux buts majeurs de l'entreprise, moyens pour les mettre en œuvre et frein rencontré

2.1 Introduction

L'entreprise est un système ouvert qui interagit avec son environnement. Cet environnement est constitué de clients, de fournisseurs mais également de concurrents, d'organismes réglementaires, etc. La survie de l'entreprise dans cet environnement fortement concurrentiel lui impose d'une part de maîtriser la qualité, le coût et le délai des produits qu'elle fournit et, d'autre part, de répondre aux attentes des clients et même de les anticiper. Pour ce faire, elle s'appuie sur des projets d'entreprise.

2.2 Identification de deux buts majeurs pour l'entreprise

Depuis de nombreuses années, la maîtrise du triplet « qualité-coût-délai » est reconnue comme étant un élément nécessaire, mais pas toujours suffisant, à la survie de l'entreprise dans un environnement fortement concurrentiel.

2.2.1 Le triplet qualité coût délai

Aujourd'hui, la « démarche Qualité » est déployée dans la plupart des entreprises [Cherfi, 02]. La mise en œuvre de cette démarche correspond à la volonté de l'entreprise d'effectuer au mieux sa mission, c'est à dire fournir les produits les plus aptes à satisfaire ses clients. Il ne s'agit pas ici de développer de longs propos sur l'intérêt d'une telle démarche mais plutôt d'en rappeler les concepts essentiels et de montrer leur incidence sur le cycle de vie d'un produit. Les normes ISO 900x définissent la *qualité* d'un produit ou d'un service comme l'aptitude à satisfaire des besoins explicites (ou exprimés) et implicites. D.A. Garvin identifie huit critères pour caractériser les aptitudes que le produit doit satisfaire [Cherfi, 02]. Nous illustrons la définition de ces critères au travers de l'exemple d'un système de navigation de type GPS (Global Positioning System). Pour rappel, les fonctions principales que doit posséder un système GPS sont d'être capable d'identifier un positionnement, de définir des trajectoires en ayant pour données d'entrée un point de départ et un point d'arrivée et de calculer des temps de parcours pour des véhicules motorisés. Les huit critères de qualité peuvent ainsi être définis et illustrés comme suit :

- la **performance** définit la manière avec laquelle le produit ou le service remplit la mission attendue : le système GPS est-il capable de proposer un parcours en fonction d'une destination donnée ?
- la **fiabilité** définit la robustesse de fonctionnement du produit ou du service dans le temps : avec le temps, le système GPS est-il toujours capable de fournir un parcours sans erreur ?
- la **durée de vie** définit le temps de fonctionnement du produit ou du service : au bout de combien de temps le système GPS tombe-t-il en panne sans être réparable ?
- la **maintenabilité** définit les caractéristiques de remise en état du produit ou du service suite à une défaillance : combien de temps faut-il pour réparer le système GPS ? Où le système GPS peut-il être réparé ? Est-il possible de disposer d'un système de remplacement pendant la durée de la réparation ?
- la **conformité aux standards** définit les caractéristiques que le produit ou le service doit satisfaire : le système GPS correspond-il aux spécifications annoncées ? le système GPS est-il en accord avec le code de la route en vigueur ?
- l'**aspect** définit des caractéristiques spécifiques telles que la couleur, le style, la forme, etc. : le système GPS a-t-il des couleurs d'affichage agréables ?
- les **fonctionnalités supplémentaires** différencient le produit ou le service pour une réponse à des besoins exprimés : le système GPS offre-t-il la possibilité de calculer des temps de parcours pédestres ?
- la **qualité perçue** définit les caractéristiques subjectives du produit ou du service perçues par le client : la marque du système GPS influence-t-elle le choix ?

Quand elle fournit un produit ou un service, l'entreprise se doit de respecter les règles (normes, directives, etc.) en vigueur. Parmi ces règles, nous pouvons citer la directive RoHS (Restriction of the

use of certain Hazardous Substances in electrical and electronic equipment), le règlement REACH (Registration, Evaluation and Authorisation of CHEMicals) et la norme ISO 9000 concernant les exigences en traçabilité. RoHS et REACH visent à interdire ou contrôler l'utilisation de substances toxiques dans les produits. La norme ISO 9000 définit la traçabilité comme « *l'aptitude à retrouver l'historique, la mise en œuvre ou l'emplacement de ce qui est examiné* ». [Terzi, 05] décrit la traçabilité comme « *la capacité d'un utilisateur (fabricant, fournisseur, vendeur,...) de tracer un produit dans le temps au travers de ses procédures de traitement. Physiquement, la traçabilité des produits consiste à maintenir des enregistrements informationnels de tous les matériaux, composants, processus opérants tout au long d'un cycle de vie défini (par exemple de l'achat des matières premières à la vente des produits finis), en utilisant une identification codée de chaque instance de produit.* » La réponse à ces règles implique ainsi que l'entreprise doit être capable de gérer les informations relatives aux produits physiques tout au long des étapes de leur vie.

Face à la concurrence accrue, l'entreprise se doit de proposer un produit de la meilleure qualité possible, c'est à dire qui corresponde le mieux aux besoins du client. Elle peut pour cela faire évoluer son portefeuille de produits de plusieurs manières. Tout d'abord, elle développe des fonctionnalités supplémentaires sur ses produits en intégrant différentes technologies. Cette tendance est illustrée par le développement des produits mécatroniques qui, outre des aspects mécaniques, intègrent de l'électronique, de l'automatique et de l'informatique. Cette évolution oblige l'entreprise à introduire des compétences issues de différents « métiers » (mécanique, informatique, automatique, etc.) tout au long du cycle de vie du produit. Une deuxième façon de faire évoluer le portefeuille de produits consiste à proposer des variantes qui permettent au client de commander un produit qu'il a composé « sur mesure ». Par exemple, lors d'une commande d'un véhicule automobile, le client dispose d'une large variété d'options allant de la configuration esthétique aux performances du moteur. Enfin, pour se démarquer des concurrents, l'entreprise peut renouveler sans cesse son portefeuille de produits par des innovations régulières. Pour cela, elle introduit de nouveaux produits dans sa gamme ou elle fait évoluer les produits existants par l'intégration de fonctions innovantes. Cette tendance peut être illustrée par l'exemple de la société Apple qui propose régulièrement des nouveautés sur sa gamme de téléphones portables (iPhone 3, iPhone 4, etc.). L'entreprise doit alors être capable de gérer simultanément plusieurs versions d'un même produit. En effet, le lancement d'une nouvelle version (iPhone 5, par exemple) n'implique pas l'arrêt du service de maintenance sur la version précédente (iPhone 4, par exemple). Ces deux derniers points, gestion de variantes et gestion d'évolutions, induisent une augmentation du nombre de produits présents dans le portefeuille de produits d'une entreprise.

Pour répondre le mieux possible aux besoins du client, l'entreprise ne doit pas limiter la notion de qualité à ses seuls produits. Elle doit déployer une organisation et des moyens (matériels ou humains) adaptés à ses ambitions. En d'autres termes, la maîtrise de la qualité induit la maîtrise des processus se déroulant tout au long du cycle de vie des produits.

Le deuxième élément du triplet est le **coût**. Dans une économie de marché, le prix de vente d'un produit ou d'un service est fixé par la concurrence. Rappelons que, dans un environnement concurrentiel, l'entreprise est soumise à cinq forces [Porter, 03]. La première force est le pouvoir de négociation des clients. Grâce à Internet, le client est aujourd'hui capable de comparer facilement des produits dans un marché mondialisé. Il peut alors négocier le prix d'un produit auprès d'un fournisseur en s'appuyant sur ceux proposés par la concurrence. La deuxième force est le pouvoir de négociation des fournisseurs. Par exemple, il est parfois difficile de changer de fournisseur lorsqu'un produit a suivi un processus de qualification. L'entreprise est alors liée à son fournisseur et aux coûts de fourniture que ce dernier lui applique. La troisième force est la menace des produits de remplacement. Parmi ces produits, on trouve les produits issus d'innovations technologiques. Par exemple, la photographie numérique est devenue un produit de remplacement de la photographie argentique. La quatrième force est la menace d'entrants potentiels. Nous pouvons citer en exemple l'entrée de la société Apple sur le marché de la téléphonie mobile. Enfin, la cinquième force est la rivalité entre les concurrents. Avec la mondialisation, cette concurrence s'est intensifiée avec l'apparition de produits conçus et fabriqués dans des pays « low cost » (Europe de l'Est, Asie, etc.).

La maîtrise des coûts englobe toutes les étapes du cycle de vie du produit (coût de développement, coût de fabrication, coût de distribution, coût de maintenance, coût de retrait). Elle nécessite d'une part d'identifier les activités qui ajoutent de la valeur au produit et, d'autre part, de piloter les processus se déroulant tout au long du cycle de vie du produit. De nos jours, beaucoup d'entreprises ont délocalisé la réalisation des opérations impliquant peu de savoir-faire dans des pays « low-cost » pour diminuer leurs coûts et rester concurrentielles. Avec ces délocalisations, le nouveau défi à relever est la coordination d'acteurs éloignés géographiquement et qui ont souvent des cultures différentes.

Le troisième élément du triplet est le *délai*. L'entreprise est jugée par son client sur trois délais. Le premier est le temps de mise sur le marché d'un nouveau produit : combien de temps faut-il attendre entre l'annonce d'un nouveau produit et la possibilité de le commander ? Le deuxième délai est le temps de livraison : combien de temps faut-il attendre pour réceptionner un produit suite à sa commande ? Enfin, le dernier délai est celui lié à la maintenance : combien de temps faut-il pour réparer une défaillance sur un produit ? Durant ces dernières années, les entreprises se sont essentiellement focalisées sur l'amélioration des temps de production. Tout d'abord, elles se sont appuyées sur des logiciels de gestion de production (GPAO – Gestion de Production Assisté par Ordinateur) pour déterminer et maîtriser les différents temps associés à la production (temps de réglage, temps de fabrication, temps d'approvisionnement, etc.). Elles ont ensuite cherché à améliorer ces temps par l'élimination des gaspillages (Juste-à-temps, Lean) en standardisant les processus de la production.

A l'heure actuelle, les entreprises ne limitent plus cette recherche d'amélioration des délais à l'étape de production, elles interviennent sur les différentes étapes qui conduisent à la réalisation des produits. L'étape de développement assure la date de mise sur le marché, l'étape de production assure la fourniture dans des délais courts et l'étape de maintenance assure un entretien rapide. Ces trois délais participent à répondre aux besoins des clients mais aussi à les fiabiliser envers une marque.

En résumé, la maîtrise du triplet « qualité-coût-délai » concerne l'ensemble des étapes du cycle de vie du produit. En effet, la définition de chacun de ces trois critères et l'identification des directives à respecter ont montré que la maîtrise du triplet ne se limite pas aux étapes de conception et de fabrication d'un produit mais qu'elle doit prendre en compte l'usage et, de plus en plus, le retrait.

2.2.2 Trois défis de l'entreprise

Pour faire face à un environnement en constante évolution, l'entreprise doit relever les trois défis suivants [CIMData, 02] :

- Améliorer la connaissance des clients : ce défi requiert une compréhension et une réponse rapide aux clients actuels et potentiels qui souhaitent disposer de produits répondant à leurs besoins.
- Atteindre l'excellence industrielle : l'entreprise doit développer une exploitation efficace, efficace et flexible. Elle doit également travailler avec ses partenaires pour réduire les coûts et les temps nécessaires pour fournir des produits qui satisfont les attentes des clients dans des délais acceptables.
- Assurer un leadership produit : l'entreprise doit fournir des produits et des solutions de pointe adaptés aux besoins des clients. Ce défi impose à l'entreprise de se développer sur de nouveaux marchés avec des produits innovants. Ceci nécessite l'exploitation et la réutilisation du capital intellectuel de l'entreprise et de celui créé par une chaîne de valeur étendue (chaîne de valeur appliquée à l'entreprise et à l'ensemble de ses partenaires).

Ces défis ont pour objectif de donner à l'entreprise la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût. En d'autres termes, ces défis contribuent à la maîtrise du triplet « qualité-coût-délai » évoqué précédemment. Ils concourent également à développer l'agilité opérationnelle de l'entreprise. L'**agilité opérationnelle*** est définie comme la capacité de l'entreprise à se reconfigurer pour s'adapter aux sollicitations toujours nouvelles des tiers dont dépendent son développement, sa rentabilité et son fonctionnement [Chelli, 03]. Elle qualifie le délai de mise en œuvre d'une évolution opérationnelle. Cette agilité permet à l'entreprise de faire face à un environnement très changeant et de développer de nouvelles innovations. Ainsi, au-delà de

l'adaptation à l'environnement (école du positionnement) [Porter, 03], le but recherché est de passer à un fonctionnement basé sur un mode proactif (école du mouvement) [Hamel et al., 90] [D'Aveni, 94] [Brown et al., 98]. Ce mode proactif est atteint quand l'innovation devient un outil de déstabilisation stratégique [Brion, 11] : les compétences clés de l'entreprise changent alors les règles du jeu du marché [Hamel et al., 90] et l'innovation en rafale surprend sans cesse les concurrents [D'Aveni, 94] [Brown et al., 98].

Ce paragraphe nous a montré que la fourniture du bon produit sur le bon marché au bon moment et avec le bon coût tout en maintenant, voire en développant, l'agilité opérationnelle est un des buts stratégiques de l'entreprise. L'atteinte de ce but implique tous les acteurs de l'entreprise, ce qui rend nécessaire le développement de projets d'entreprise.

2.3 Les projets d'entreprise

Quand on parle de projets d'entreprise, les deux questions suivantes se posent : quel est le rôle des projets dans l'atteinte des buts fixés par l'entreprise ? Quel est le rôle de l'entreprise dans le déroulement des projets qu'elle entreprend ? Ce paragraphe a pour objectif de répondre à ces questions en s'appuyant sur une étude bibliographique. Il identifie également différentes causes qui peuvent conduire ce type de projet à des échecs. Parmi ces causes, ce paragraphe présente plus précisément les disjonctions et les contradictions qui peuvent apparaître entre les perceptions que les différents acteurs ont de l'entreprise.

2.3.1 Définition et classification des projets d'entreprise

Différentes études montrent que le concept de projet d'entreprise n'est pas toujours aisé à définir : « *Project business is the part of business that relates directly or indirectly to projects, with a purpose to achieve objective of a firm or several firms* » [Artto et al., 05]. Certains travaux proposent d'introduire la notion de management de programme pour définir la gestion d'un ensemble de projets interdépendants qui concourent à l'atteinte des buts de l'entreprise ; ces travaux montrent que ces buts ne peuvent pas être réalisés si ces projets sont gérés indépendamment [Lycett et al., 04]. D'autres travaux proposent un cadre pour analyser et classifier les contributions de recherche concernant les projets et la gestion de projet [Söderlund, 04]. Ces différentes études établissent que les considérations organisationnelles et managériales sont contextuelles à l'entreprise, à son activité et à son environnement. Une étude propose également une classification des projets d'entreprise (cf. tableau 1.1) selon trois pôles d'observation de l'entreprise : les *processus*, l'*organisation* et le *changement technologique et sociologique* [Artto et al., 05].

Processus	Projets relatifs au processus et au succès du développement du produit
	Projets relatifs à l'accélération du développement du nouveau produit
	Projets relatifs à la performance en production et au développement industriel
	Projets relatifs à l'organisation de la R&D et à la compréhension des approches de management
	Projets relatifs au management multi-projets
Organisation	Projets relatifs à la théorie de l'organisation et à la conception organisationnelle
	Projets relatifs à l'apprentissage, au transfert et à l'accumulation de connaissances organisationnelles
	Projets relatifs à l'interface entre R&D et marketing
	Projets relatifs à la collaboration entre les organisations
Changement technologique et sociologique	Projets relatifs aux théories sociologique et psychométrique
	Projets relatifs au changement technologique et économique

Tableau 1.1 : Classification des projets d'entreprise [Artto et al., 05]

En résumé, un projet d'entreprise conditionne l'évolution de cette dernière en dimensionnant son ambition. De plus, il définit les défis à relever et il motive les salariés en leur proposant un cadre pour créer et agir ensemble de manière cohérente.

Cependant, force est de constater que certains projets d'entreprise se soldent par un échec. Parmi les causes d'échecs identifiables, nous retenons le manque de cohérence entre les actions menées par les différents acteurs de l'entreprise. Dans le cadre de ces travaux, nous nous intéressons plus particulièrement aux actions menées dans les différentes étapes du cycle de vie du produit et à celles qui visent à développer ou déployer des solutions informatiques pour automatiser certaines tâches ou pour assister les acteurs de l'entreprise dans leurs prises de décision. Lorsque ces deux familles d'acteurs participent à un même projet d'entreprise, des échecs apparaissent lorsque les actions qui concernent la réalisation du produit sont revues sans se préoccuper des conséquences sur les applications logicielles ou lorsque l'informatisation est réalisée sans prise en compte des « besoins du terrain » [Chelli, 03]. Ce constat montre que les échecs peuvent être liés au fait que les différents acteurs impliqués dans un projet d'entreprise n'ont pas la même perception de cette dernière. Le paragraphe suivant présente les caractéristiques spécifiques aux perceptions des deux familles d'acteurs qui sont au centre des préoccupations abordées dans le cadre de ce mémoire.

2.3.2 Deux perceptions de l'entreprise

Le paragraphe précédent a introduit deux familles d'acteurs [Chelli, 03] :

- Les **responsables métiers** qui interviennent tout au long de la réalisation et de la fourniture des produits : marketing, commercial, études, industrialisation, production, maintenance, etc. Ces acteurs s'appuient généralement sur l'outil informatique pour réaliser les tâches qui leur incombent.
- Les **techniciens de l'information** qui conçoivent et construisent l'outil informatique nécessaire au fonctionnement de l'entreprise : architectes, concepteurs, analystes, programmeurs, exploitants, etc.

L'agilité opérationnelle dépend d'une part des responsables métiers pour ce qui concerne les nouvelles compétences, les nouvelles offres ou les nouveaux services et, d'autre part, des techniciens de l'information pour ce qui concerne le développement de nouveaux services automatisés, l'extension des programmes existants, l'accroissement des capacités de traitement ou la multiplication des moyens d'accès. L'entreprise est ainsi à la fois organisationnelle (monde des hommes) et informationnelle (monde de l'information numérique).

L'existence de ces deux familles d'acteurs induit deux pôles d'observation d'une même réalité opérationnelle. Le **pôle d'observation organisationnel** considère les hommes et les processus que nous désignons processus métiers, qui fournissent les produits et les services qui sont la raison d'être de l'entreprise. Le **pôle d'observation informationnel** considère les services automatisés qui manipulent les informations nécessaires au déroulement des processus métiers. Ces informations décrivent ce qu'il faut produire (les produits, les nomenclatures, etc.), la production (les gammes, l'ordonnancement des activités, les biens consommés, etc.) et l'environnement (les concurrents, les partenaires, les organismes de contrôle, le marché, etc.).

Les perceptions relatives à ces deux pôles d'observation représentent la même réalité opérationnelle de l'entreprise. Elles sont construites à partir de concepts qui sont certes différents mais qui ne doivent être ni disjoints ni contradictoires. Ainsi, l'efficacité et l'agilité de l'entreprise diminuent chaque fois qu'une séparation des deux perceptions introduit des disjonctions ou des contradictions entre les perceptions [Chelli, 2003]. Les causes de différences entre ces perceptions peuvent être nombreuses. Le paragraphe suivant présente deux causes majeures qui seront en partie traitées par les propositions de ce mémoire.

2.3.3 Séparation des échelles et des langages.

Deux causes majeures conduisent à une différence de perception entre les responsables métiers et les techniciens de l'information : la séparation des échelles et la séparation des langages.

Séparation des échelles

Les défis identifiés dans le paragraphe 2.2.2 obligent l'entreprise à minimiser le temps de mise sur le marché des nouveaux produits. Pour faire face à cette accélération, les programmes associés aux différentes activités du cycle de vie du produit cherchent à continuellement diminuer les délais : programmes de commande, de production, de livraison et de facturation. Cette évolution organisationnelle implique une évolution informationnelle pour gérer la quantité toujours plus importante des données nécessaires à la réalisation de ces activités et pour automatiser certains traitements : application pour assister la gestion de la relation client (CRM – Customer Relationship Management), la gestion de l'entreprise (ERP – Enterprise Resource Planning), l'exécution de la production (MES – Manufacturing Execution System), etc.

Cependant, le cycle organisationnel est court au regard du cycle informationnel, et ceci malgré les progrès techniques qui ont permis de considérablement réduire ce dernier au cours des dernières années. Le décalage entre les deux cycles est à la fois la cause et la conséquence de la différence entre les deux perceptions. Il en est la cause car seul un découplage (désynchronisation) permet la mise en place de solutions palliatives par les responsables métiers dans l'attente de la mise à disposition des applications informatiques (développement de nouveaux services informatisés ou améliorations des services existants). Il en est la conséquence car ce découplage a engendré des logiques et des stratégies de plus en plus différenciées de la part des deux familles d'acteurs.

Séparation des langages

Le décalage progressif entre les responsables métiers et les techniciens de l'information a conduit chaque famille d'acteurs à se construire son propre langage dont le vocabulaire décrit les concepts liés à sa perception. Cette séparation des langages est à l'origine de nombreux oublis, erreurs et non-sens dans le développement des applications informatiques.

Les responsables métiers abandonnent progressivement une description précise et détaillée des activités pour privilégier un enrichissement des tâches, une plus grande autonomie et une responsabilité des individus. Par contre, les langages utilisés dans le développement des applications informatiques sont des langages formels qui nécessitent une très grande attention. En effet, une erreur mineure (oubli d'initialisation d'une variable, inversion de deux instructions, utilisation d'un opérateur logique « OU » à la place de « ET », etc.) peut provoquer des pannes très difficiles à détecter.

Face à une telle situation, il existe deux solutions pour que les responsables métiers et les techniciens de l'information puissent échanger autour des buts, des besoins et des contraintes liés au système d'information :

- définir un langage commun partagé par les deux familles d'acteurs,
- disposer d'un traducteur capable de transcrire les discours dans le langage de chaque perception.

La première solution, promue par les techniciens de l'information, a consisté à développer des langages adaptés aux exigences de la perception informationnelle et s'appuyant sur des concepts « non techniques » pour favoriser une adoption par les responsables métiers. Cette approche a été à l'origine de nombreux langages basés sur une organisation de symboles graphiques pour décrire des besoins fonctionnels et non fonctionnels : diagramme de cas d'utilisation, modèle entité-association, etc. A l'heure actuelle, aucun langage de modélisation ne s'est réellement imposé.

La seconde solution nécessite d'étudier les correspondances entre les concepts présents dans chacun des langages. Ceci permet d'identifier des règles de transformation entre les différentes perceptions. Cette solution est celle qui est développée dans le cadre de la proposition de ce mémoire.

2.4 Conclusion

En résumé, l'entreprise manufacturière doit être capable de fabriquer et fournir des produits aptes à satisfaire les attentes des clients (qualité) avec des coûts et des délais acceptables au regard du marché. Deux buts majeurs pour l'entreprise ont ainsi été mis en lumière. Le premier but consiste à avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût. Le deuxième

but vise à développer l'agilité opérationnelle pour être capable de faire face à un environnement très changeant et afin de pouvoir adopter un mode proactif. Pour atteindre ces buts, l'entreprise met en œuvre des projets d'entreprise (Qualité, Amélioration continue, Lean, etc.).

La réalisation des deux buts identifiés n'est pas liée aux seuls processus de fabrication, elle est affectée par toutes les étapes du cycle de vie. Ceci conduit de plus en plus d'entreprises à développer une approche PLM (Product Lifecycle Management). Cette approche focalise sur le produit et le développement d'une collaboration efficace au sein de l'entreprise ou dans le cadre d'une entreprise étendue. Elle s'appuie principalement sur deux familles d'acteurs : les responsables métiers qui interviennent dans les différentes étapes du cycle de vie des produits et les techniciens de l'information qui doivent fournir un ensemble d'applications informatiques adapté aux besoins des utilisateurs.

Enfin, ce paragraphe a montré que les échecs d'un projet d'entreprise tel qu'un projet PLM peuvent être dus à des perceptions disjointes voire contradictoires de la réalité opérationnelle. Deux causes de ces différences de perception ont été identifiées et sont traitées dans la suite de ce mémoire. Mais avant d'aborder cette proposition, il nous semble essentiel de rappeler les concepts liés à la gestion du cycle de vie des produits car ils sont à la base de ces travaux.

3 Le cycle de vie d'un produit

Avant de faire un état de l'art sur la gestion du cycle de vie du produit (ou approche PLM), ce paragraphe présente dans un premier temps les différents cycles de vie du produit et les étapes qui les composent. Il positionne ensuite ces étapes sur les trois processus transversaux identifiés dans le paragraphe 2.1.1. Enfin, il rappelle les concepts relatifs à la collaboration dans le cadre de l'entreprise étendue.

3.1 Cycles de vie commerciaux et techniques

L'entreprise manufacturière considère les états de maturité de ses produits selon deux points de vue :

- la maturité commerciale est identifiée au travers des ventes des produits physiques. Le volume des ventes du produit est en croissance, en maturité, en déclin [Kotler, 96].
- la maturité technique est identifiée au travers du produit théorique et du produit physique. Le produit est défini, réalisé, exploité et retiré. [Gzara, 00]

La maturité commerciale correspond au cycle de vie commercial. Pour des produits vendus aux utilisateurs finaux et fabriqués en série, ce cycle est représenté selon le concept Product Life Cycle (cf. figure 1.1). Il comprend quatre phases séquentielles : l'introduction du produit sur le marché, la croissance des ventes, la maturité du marché puis le déclin. Ce concept est à adapter selon les types de produits au niveau de la forme de la courbe (plus ou moins anguleuse) et de l'échelle de temps (cycles de vie plus ou moins longs) [Kotler, 96].

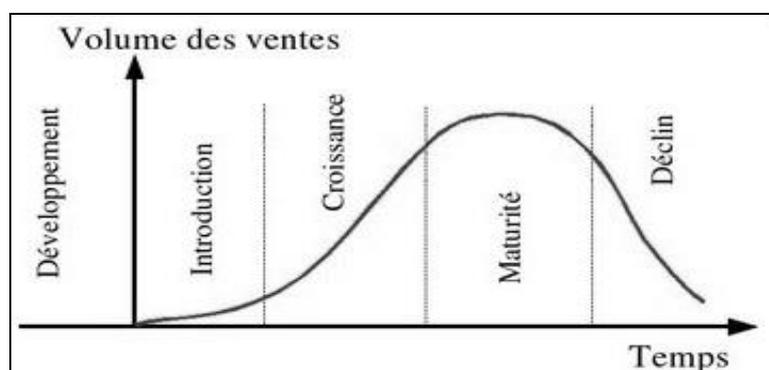


Figure 1.1 : Cycle de vie commerciale du produit [Kotler, 96]

La maturité technique correspond quant à elle au cycle de vie suivant (cf. figure 1.2). Ce cycle est structuré en trois phases ; chaque phase englobe une ou plusieurs étapes [Terzi et al., 07].

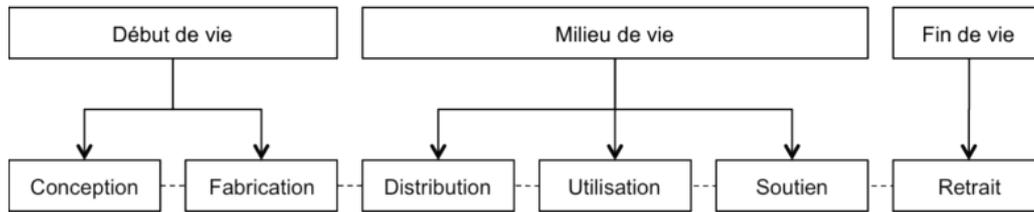


Figure 1.2 : Cycle de vie du produit (adapté de [Terzi et al., 07])

Le début de la vie correspond à la phase pendant laquelle le produit est conçu et réalisé. Il est divisé en deux étapes : l'étape de conception et l'étape de fabrication. L'étape de conception regroupe la conception du produit, la conception du procédé associé, voire la conception de l'usine de fabrication. Chaque étape de conception inclut des tâches relatives à l'élaboration du cahier des charges, la conception et les tests. L'étape de fabrication inclut les tâches de production et de logistique interne.

Le milieu de la vie correspond à la phase pendant laquelle le produit est distribué, utilisé et maintenu. Il est divisé en trois étapes : l'étape de distribution, d'utilisation et de support. La distribution correspond à la logistique externe. Le support correspond à la maintenance et à la réparation.

La fin de la vie correspond à la phase pendant laquelle le produit est collecté, désassemblé, recyclé, réutilisé ou détruit.

Toutes les étapes pouvant être vécues par un produit, qu'il soit tangible ou intangible (logiciel), sont décrites dans ce cycle. Par contre, un produit ne suit pas forcément chaque étape identifiée. D'une part, des produits ne sont pas concernés par certaines étapes, par exemple la maintenance et le démontage. D'autre part, ces étapes sont des étapes théoriques. Ainsi, dans la réalité, un produit peut s'avérer irréalisable et donc ne jamais suivre les étapes de fabrication, de distribution, d'utilisation et de retrait prévues initialement.

3.2 Les boucles retour dans le cycle de vie du produit

Lorsqu'un produit est fabriqué en série, certaines étapes de son cycle de vie sont naturellement non séquentielles. Imaginons un produit à réaliser en neuf exemplaires. Les conceptions du produit et du procédé de fabrication associé sont réalisées. Dès la fin de fabrication du premier produit « physique », les étapes suivantes se déclenchent alors que les huit autres exemplaires du produit ne sont pas encore fabriqués.

Dans le schéma suivant, un chiffre est attribué à chacun des neuf exemplaires du produit physique. A un instant t , certains exemplaires sont fabriqués, d'autres en distribution, d'autres utilisés et d'autres retirés.

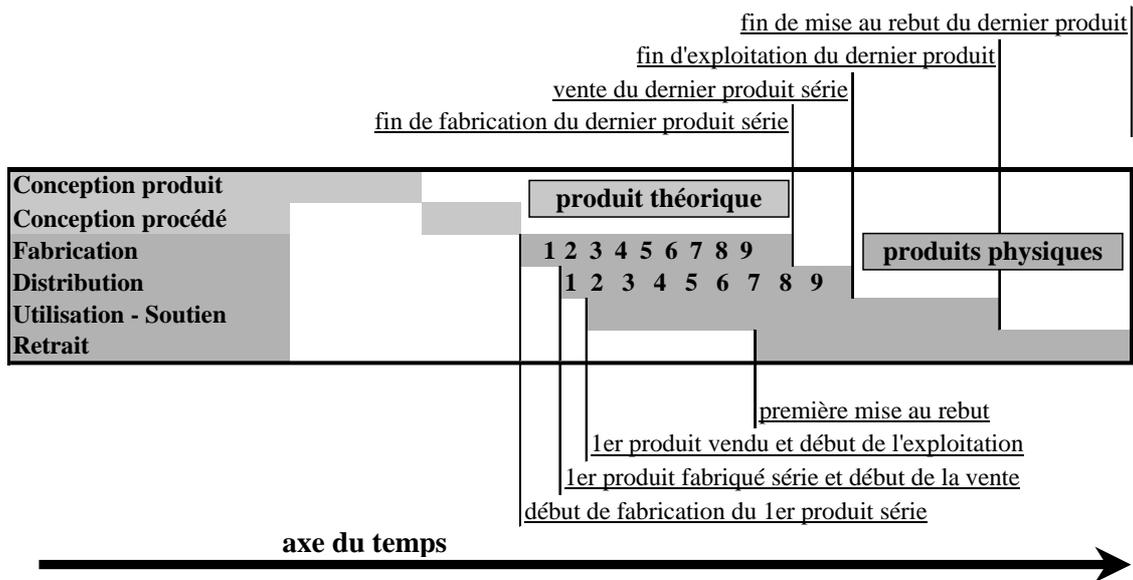


Figure 1.3 : Exemple du cycle de vie d'un produit

Durant la vie du produit, le produit et le procédé de fabrication associé peuvent évoluer. Les causes peuvent être externes, par exemple une évolution du cahier des charges demandée par le client ou des normes applicables. Les causes peuvent aussi être internes. Par exemple, un problème détecté en fabrication sur le 2^{ème} produit peut induire une amélioration qui sera mise en œuvre sur les produits suivants (3^{ème}, 4^{ème}, 5^{ème}, etc.). On parle alors de boucles de retour [Kiritsis et al., 03]. Ces évolutions impliquent des modifications des définitions initialement validées. Ainsi, les étapes de conception du produit et du procédé de fabrication associé ne se terminent pas au commencement de l'étape de fabrication mais à la fin de la fabrication voire à la fin de l'exploitation du dernier produit série.

Aujourd'hui, le coût et le délai de fabrication d'un produit sont aussi importants que son adéquation fonctionnelle à répondre au besoin. Les méthodes « Design to Cost » (conception en vue d'un coût objectif) et d'ingénierie concourante sont mises en œuvre pour répondre à ces impératifs. Elles impliquent une conception intégrée et simultanée des produits et des procédés associés. L'objectif est de réduire le délai de mise sur le marché et de prendre en compte les contraintes de la production dès la conception du produit. Concernant l'enchaînement des étapes, l'étape de conception du procédé de fabrication commence dès que des éléments de définition du produit sont validés. Ceci permet d'anticiper la fabrication d'outillages ou l'approvisionnement de matières alors que la conception du produit n'est pas totalement finalisée.

Voici le schéma précédent du cycle de vie mis à jour avec ces notions d'évolutions et d'ingénierie concourante :

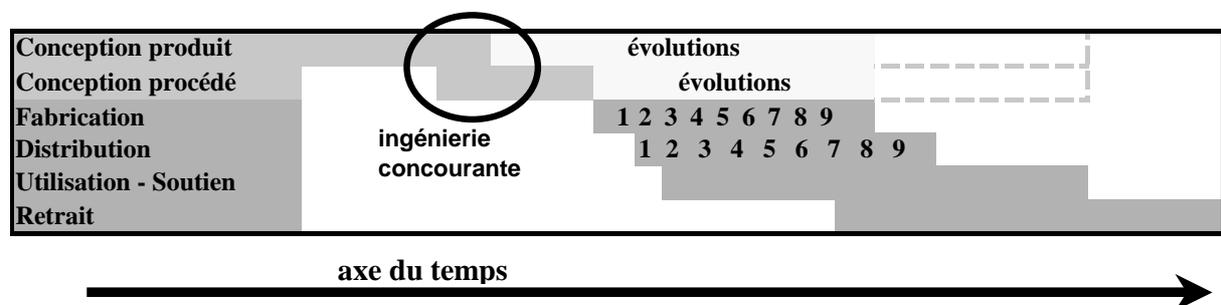


Figure 1.4 : Exemple du cycle de vie d'un produit - suite

3.3 Les processus transversaux du cycle de vie

Pour être concurrentielle, l'entreprise doit parfaitement maîtriser la qualité, les coûts et les délais associés aux étapes du cycle de vie du produit. Cette maîtrise nécessite d'identifier des activités qui contribuent à ajouter de la valeur au produit. Elle nécessite également de piloter les processus se déroulant tout au long du cycle de vie du produit. Parmi ces processus, trois en particulier ont été identifiés dans le paragraphe 2.2.1 (cf. figure 1.5).

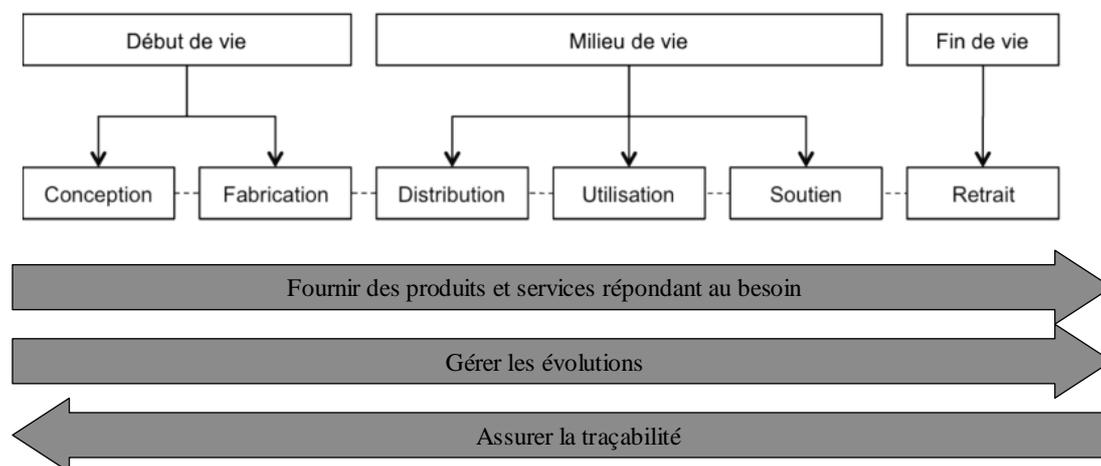


Figure 1.5 : Les processus globaux liés au cycle de vie du produit (adapté de [Terzi et al., 07])

Le premier processus concerne la capacité à fournir un produit physique répondant au cahier des charges du client. Le client exprime son besoin au début du cycle de vie. Seule une maîtrise de toutes les étapes (conception du produit, conception du procédé, fabrication...) rend possible l'adéquation entre les caractéristiques réelles du produit physique et des services de soutien associés et les caractéristiques décrites dans le cahier des charges initial. Le deuxième processus concerne la capacité à analyser les conséquences d'une évolution. Nous avons vu que pour renouveler sans cesse sa gamme de produits, l'entreprise est amenée à gérer plusieurs évolutions de son produit. Il est alors nécessaire d'identifier clairement quelles sont les informations qui concernent chacune des évolutions du produit. Il faut également être capable, lorsqu'une exigence évolue, d'analyser les conséquences de cette évolution sur toutes les étapes du cycle de vie. Le troisième processus concerne la capacité à assurer la traçabilité tout au long des étapes du cycle de vie. Pour répondre aux exigences réglementaires, il est nécessaire d'être capable de retrouver pour un produit physique les informations relatives à ce produit quelle que soit l'étape du cycle de vie concernée.

3.4 Les acteurs du cycle de vie : de l'entreprise à l'entreprise étendue

Les produits devenant de plus en plus complexes, de plus en plus d'acteurs sont impliqués dans leur conception et réalisation. Des relations donneur d'ordre / sous-traitant s'instaurent alors entre les différents acteurs. La norme AFNOR X 50-300 [AFNOR, 87] définit comme activités de sous-traitance « toutes les opérations concourant pour un cycle de production déterminé à l'une ou plusieurs des opérations de conception, d'élaboration, de fabrication, de mise en œuvre ou de maintenance du produit en cause, dont une entreprise, dite donneuse d'ordres, confie la réalisation à une entreprise dite sous-traitante ou preneur d'ordres, tenue de se conformer exactement aux directives ou spécifications techniques arrêtées en dernier ressort par le donneur d'ordres. »

Dans cette relation donneur d'ordres / sous-traitant, l'entreprise sous-traitante réalise un produit en fonction du cahier des charges de son donneur d'ordres. Si le produit est complexe, le sous-traitant d'un sous-ensemble peut lui-même sous-traiter une partie de sa mission à une autre entreprise. Il devient ainsi lui-même donneur d'ordres. Dans l'industrie automobile, les différents niveaux de découpage du produit sont définis par des rangs (cf. figure 1.6). Par exemple, un constructeur de

voitures délègue à un sous-traitant de rang 1 les planches de bord. Ce sous-traitant délègue à son tour à un sous-traitant de rang 2 les éléments de fixation.

Pour chaque niveau, le choix de faire ou de faire-faire la conception et la fabrication résulte d'une analyse des compétences internes et d'une confrontation avec les buts stratégiques de l'entreprise [Perrotin et al., 99]. Chaque « découpage » du produit concernant la conception et la fabrication implique un partage des responsabilités légales des choix et des validations et un partage des droits de propriété intellectuelle [Le Dain et al., 07].

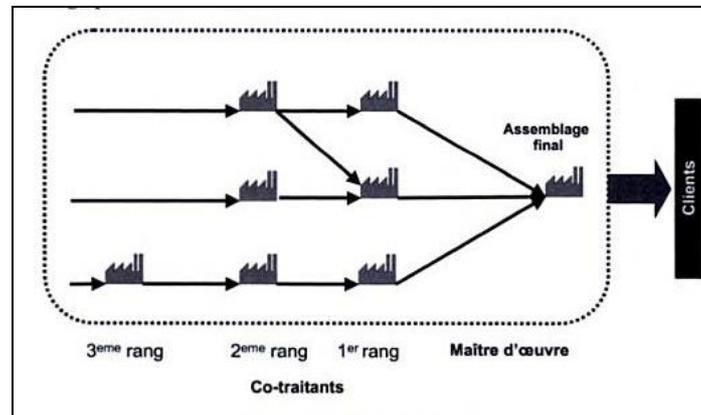


Figure 1.6 : L'entreprise étendue [Capraro et al., 02]

Si le donneur d'ordre confie la réalisation d'opérations à un groupe d'entreprises, ces sous-traitants sont alors cotraitants entre eux. Selon la norme AFNOR X 50-300 [AFNOR, 87], la cotraitance est la réalisation par un groupe d'entreprises d'un programme étudié. Cependant, le terme de cotraitance implique une parité entre les acteurs ce qui n'est plus le cas dès lors qu'il y a un donneur d'ordres dans le groupe [Altersohn, 97]. Concernant l'exemple de l'industrie automobile, deux sous-traitants de rang 1 peuvent avoir des relations de cotraitance entre eux si la conception des sous-ensembles dont ils sont responsables est simultanée. Par exemple, le sous-traitant responsable du rétroviseur s'accorde avec le sous-traitant responsable du pare-brise sans qu'aucun des deux ne soit le donneur d'ordres l'un de l'autre.

3.5 Conclusion

Ce paragraphe a identifié deux cycles de vie : le cycle commercial et le cycle technique. Il a également établi que les étapes de la vie d'un produit ne sont pas nécessairement séquentielles et que des boucles retour d'informations peuvent exister. Enfin, les acteurs nombreux et variés travaillant autour d'un produit dans le cadre de l'entreprise étendue ont été présentés.

L'identification des étapes de la vie du produit a permis de positionner trois processus globaux. Ces processus définissent la capacité à fournir un produit physique répondant au cahier des charges du client, la capacité à analyser les conséquences d'une évolution et la capacité à assurer la traçabilité. Elle a aussi mis en lumière la diversité des métiers concernés tout au long du cycle de vie : conception, fabrication, distribution, maintenance, etc. Ces métiers étant très différents les uns des autres, nous pouvons considérer que la perception organisationnelle (cf. § 2.3.2) inclut plusieurs perceptions.

Suite à cette mise en contexte préliminaire, nous pouvons maintenant dresser un état de l'art de ce qu'est l'approche PLM ou la gestion du cycle de vie produit.

4 Etat de l'art : l'approche Product Lifecycle Management (PLM) ou la gestion du cycle de vie produit

Un des chapitres du livre de John Stark [Stark, 05] s'intitule « *You start here – in the details of the PLM swamp* »⁵. La formulation est certes bien trouvée au vu de la difficulté ne serait-ce que de définir ce qu'est le PLM. Nous proposons dans ce paragraphe de poser des balises dans ce « marais » afin de pouvoir y naviguer par la suite de manière efficace. Tout d'abord, nous faisons un état de l'art des origines, des définitions et des buts de ce que nous appellerons « l'approche PLM ». Puis, nous étudions les principes servant de base pour la mise en œuvre du projet PLM mais aussi pour le développement d'une culture PLM au sein de l'entreprise.

4.1 L'approche PLM

Pour comprendre au mieux ce qu'est l'approche PLM, nous commençons tout d'abord par identifier ses multiples origines. Puis, nous étudions les termes servant à qualifier le PLM et les différents buts qui sont associés à cette approche. Ces différentes descriptions nous permettent de proposer la définition que nous utilisons pour la suite de nos travaux.

4.1.1 Les origines de l'approche PLM

L'approche PLM est souvent considérée comme le résultat de la conjonction de plusieurs méthodes opérationnelles. Ces méthodes répondent à des besoins distincts mais elles sont en accord avec une finalité commune : raisonner et agir sur le produit pour faire en sorte que toutes les étapes de la vie du produit contribue à fabriquer le bon produit sur le bon marché au bon moment et avec le bon coût. Ces méthodes proviennent de deux domaines : celui des métiers travaillant autour du produit (perception organisationnelle de l'entreprise) et celui de la gestion des informations (perception informationnelle de l'entreprise). Nous les présentons sous trois catégories : les méthodes liées à l'étape de conception du produit, celles liées au couple (étape de conception du produit, étape de conception du procédé) et celle liées au cycle de vie complet. Notre objectif n'est pas de faire une liste exhaustive mais de montrer la diversité des courants de pensée qui ont conduit à l'émergence de l'approche PLM.

Depuis les années 70, les produits connaissent une forte évolution technologique. Pour illustrer ce fait, prenons l'exemple d'une voiture. Ce produit était à l'origine essentiellement mécanique. Il est devenu aujourd'hui un pur produit mécatronique. Ainsi, la conception du produit est à présent le résultat d'un travail collaboratif entre des acteurs de métiers différents et pouvant appartenir à diverses entreprises [Devalan, 10]. Cette conception collaborative implique d'avoir une perception globale du produit afin de « découper » et coordonner son développement. La conception doit répondre à plusieurs besoins et contraintes. Ainsi, la conception en vue de X (ou « design for X ») est une méthode ayant pour but d'optimiser l'aspect X d'un produit. Le X peut représenter une caractéristique du produit (conception en vue « d'un coût objectif », « de la Qualité du produit », etc.). Il peut également représenter une caractéristique du projet de développement (conception en vue « d'un délai court de mise sur le marché »). Enfin, il peut représenter des besoins et des contraintes issus des étapes de la vie du produit (conception en vue de la « fabrication », de « l'assemblage », du « confort d'utilisation ou de réparation », du « désassemblage », ...).

Les Systèmes de Gestion de Données Techniques, ou SGDT, gèrent de manière intégrée et structurée les différents éléments de la définition du produit : fichiers de conception (ou fichiers CAO – Conception Assistée par Ordinateur), documents et nomenclatures. Ils gèrent les évolutions en utilisant des versions et ils identifient les états de maturité grâce à des statuts (« en création », « valide », « obsolète », par exemple). Ils constituent au sein de l'entreprise un référentiel commun des informations de définition du produit.

L'ingénierie concurrente (ou concurrent engineering) est une évolution de la conception en vue de la fabrication. Elle a pour but d'améliorer le délai de mise sur le marché et la qualité du produit [Maurino, 94]. Elle implique de concevoir de façon intégrée et simultanée le produit et les moyens de

⁵ « *Vous commencez ici – dans les détails du marais du PLM* »

production, de mise en œuvre et de maintenance associés. Pour cela, il est nécessaire de découper la définition du produit en différentes parties qui sont fournies au fil du développement aux concepteurs du procédé de fabrication. Ces derniers commencent alors la définition et la mise en œuvre des moyens de production alors que la définition du produit n'est que partiellement validée.

Concernant la gestion intégrée des informations relatives aux conceptions du produit et du procédé associé, l'usage de la maquette numérique est aujourd'hui devenu une réalité. « La maquette numérique est *« une représentation numérique étendue du produit utilisée comme plate-forme de développement produit / processus, de communication et de validation durant toutes les phases de la vie du produit »* » [Morenton, 06]. Elle sert entre autres à la mise en œuvre du concept d'usine numérique. L'usine numérique est une représentation numérique du procédé de fabrication qui permet une visualisation et une expérimentation virtuelle. Parmi les offres logicielles, nous pouvons citer le Manufacturing Process Management⁶ (MPM) proposé par les éditeurs de logiciels PLM. Les modules MPM ont des fonctionnalités variables parmi lesquelles la création des nomenclatures du produit pour fabrication à partir des nomenclatures issues de la conception produit.

Plusieurs actions et raisonnements concernent toutes les étapes de la vie du produit. Dans les paragraphes précédents, nous avons ainsi identifié la nécessité de prendre en compte la traçabilité des produits dans les méthodes de travail. Nous pouvons également citer la mise en œuvre d'une méthode de raisonnement transversale : l'analyse du cycle de vie du produit. Cette méthode a pour objectif d'évaluer l'impact environnemental du produit en considérant toutes les étapes de sa vie [Jolliet et al., 05].

Concernant la gestion transversale des informations, la chaîne numérique traite la géométrie du produit tout au long des étapes de conception du produit, de conception du procédé et de fabrication [Polizzi, 04]. En s'appuyant sur les géométries réalisées grâce au logiciel de CAO, un programme est créé à l'aide du logiciel de FAO (Fabrication Assistée par Ordinateur). Ce programme décrit les trajectoires des outils que la machine-outil doit exécuter pour réaliser le produit conçu. Un logiciel « post-processeur » transforme ensuite ce programme en code compréhensible par la machine-outil à commande numérique. Les géométries issues de CAO sont également utilisées comme données d'entrée de certaines machines de contrôle à mesure tridimensionnelle. La chaîne numérique permet de gagner du temps de programmation et d'éviter les erreurs relatives à des interventions manuelles. Cependant, la diversité des formats échangés et les évolutions désynchronisées des différents logiciels sont des freins à sa mise en œuvre.

Ce paragraphe a mis en lumière diverses méthodes apparues au fil du temps qui sont à l'origine de l'approche PLM.

4.1.2 PLM, un concept encore jeune

La figure 1.7 illustre le nombre de résultats à la requête : « product lifecycle management » sur le site Science Direct (réalisée avec les guillemets le 20 février 2013). Depuis 2002, les articles scientifiques relatifs au PLM sont de plus en plus nombreux chaque année. Entre 2004 et 2006, plusieurs ouvrages spécifiquement dédiés au PLM sont édités. Parmi les auteurs, peu ont un lien avec le monde académique : John Stark, Antti Sääksvuori et Denis Debaecker sont consultants, Anselmi Immonen est Vice President de l'entreprise Rocla Oyj et Michael Grieves a une carrière académique et industrielle.

⁶ Module MPM du produit Teamcenter de Siemens (06/2011)

http://www.plm.automation.siemens.com/en_us/products/teamcenter/solutions_by_product/mfg_process.shtml

Module MPM du produit Windchill de PTC (06/2011)

http://www.ptc.com/WCMS/files/82862/fr/3799_WC_MPM_DS_FR.pdf

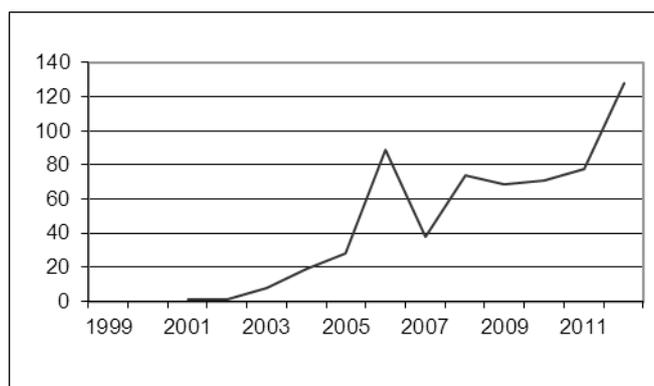


Figure 1.7: Nombre de résultats à la requête « product lifecycle management » sur le site Science Direct

Dans le milieu industriel, le terme PLM est apparu un peu avant 2000 sur le marché logiciel. En 2007, Jean-François Prevéraud, journaliste à Industrie & Technologies et l'Usine Nouvelle affirmait : « *Huit ans après le lancement du vocable PLM par l'éditeur SAP et sa récupération rapide par Dassault Systèmes, ce concept marketing est devenu une offre tangible chez les éditeurs et, surtout, une réalité chez les industriels* »⁷. Pourtant, il concluait la journée du PLM 2010 en constatant que chacun des 225 participants avait probablement une définition du PLM qui lui était propre. Ainsi, force est de constater que le concept « PLM » est difficile à caractériser.

4.1.3 Les qualificatifs du PLM

Afin de déterminer ce qu'est le PLM, un état de l'art des définitions existant dans les travaux suivants a été réalisé : [Ameri et al., 05] [Dutta et al., 05] [Stark, 05] [Sudarsan et al., 05] [Grieves, 06] [Jun et al., 07] [Terzi et al., 07] [Saaksvuori et al., 08].

A partir de ces définitions, une liste des mots choisis pour qualifier le PLM se profile : paradigme, concept holistique, concept d'entreprise, stratégie d'entreprise, approche, solution de gestion des connaissances, processus et activité de gestion d'entreprise. Dans ces définitions, le périmètre du PLM est délimité par toutes les étapes de la vie du produit et par l'entreprise elle-même, voire par un réseau d'entreprises. Pour John Stark, le PLM est une manière d'appréhender l'entreprise centrée sur les produits et dans laquelle toute décision et toute action de l'entreprise s'inscrit. Pour les autres auteurs, le PLM est une pratique de management qui contribue à l'atteinte de buts stratégiques relatifs aux produits de l'entreprise.

Dans le cadre de nos travaux, nous choisissons d'utiliser le mot « approche » pour qualifier le PLM. Nous considérons une **approche*** comme un cadre de référence qui répond à un projet en s'appuyant sur des buts, des principes, des méthodes et des outils pour développer une culture. Afin d'illustrer l'emploi de ce mot, nous reprenons le parallèle entre l'approche Lean et l'approche PLM [Ameri et al., 05] [Pinel et al., 11c]. L'approche Lean a pour but l'amélioration de la performance fondée sur l'élimination des gaspillages [Lyonnet, 10]. Cependant, force est de constater que l'entreprise faisait déjà de l'élimination de gaspillage avant que l'approche Lean n'existe. L'intérêt de l'approche Lean est qu'elle définit des principes (l'élimination des gaspillages, le juste à temps, la qualité, l'amélioration continue, le management visuel et le management des hommes), des méthodes et des outils [Lyonnet, 10]. Elle constitue ainsi un cadre de référence qui permet à l'entreprise d'acquérir une « culture Lean ». L'identification des buts associés à un projet d'entreprise relatif à l'élimination des gaspillages se fait alors en ayant un raisonnement structuré selon les principes de l'approche Lean. La mise en œuvre de ce projet est supportée par l'utilisation des méthodes et des outils relatifs à cette approche. De façon similaire, l'approche PLM a des buts, des principes, des méthodes et des outils qui constituent un cadre de référence relatif à un management centré sur les produits, sur la collaboration et sur les savoirs faire. La culture PLM est basée sur la confiance, l'ouverture, la réutilisation des connaissances, la collaboration et l'innovation [Ameri et al., 05].

⁷ http://www.knowllence.com/fr/publications/tdc_et_plm2-0_jf_preveraud.php

L'approche PLM est ainsi un cadre de référence que l'entreprise utilise selon ses besoins. Elle considère l'entreprise selon un pôle d'observation donné, à savoir les produits, la collaboration et les savoirs faire. Il convient donc pour chaque entreprise d'appréhender l'approche PLM en fonction de son contexte, de son existant, de ses priorités, etc. Pour juger de la pertinence de la mise en œuvre de ce qu'il est d'usage d'appeler un projet PLM, l'entreprise doit étudier la corrélation entre les buts identifiés dans l'approche PLM et les buts définis comme stratégiques pour l'entreprise. Si une adéquation est mise en lumière, un projet d'entreprise s'appuyant sur les principes, les méthodes et les outils de l'approche PLM peut alors être mis en œuvre.

La notion de culture PLM induite par le terme approche est particulièrement intéressante. En effet, elle montre l'importance de la gestion du changement dans le cadre d'un projet PLM. Elle illustre également le fait que les différents travaux autour du PLM n'ont pas vocation à imposer une méthode à appliquer à la lettre. Leur objectif est de proposer un cadre de référence sur lequel l'entreprise peut s'appuyer.

Afin de définir ce qu'est l'approche PLM, nous allons maintenant nous intéresser aux buts qui lui sont attribués.

4.1.4 Les buts de l'approche PLM

Voici ci-dessous les buts de l'approche PLM identifiés dans différents articles :

“The objectives of PLM are to increase product revenues, reduce product-related costs, and maximize the value of the product portfolio.” [Stark, 05]

“to produce the desirable outcomes of reduced time to market, improved profitability over the lifecycle and a stronger, more competitive market position.” [Dutta et al., 05]

“such that the right information in the right context at the right time can be made available.”

“for creating a product-centric environment” [Ameri et al., 05]

“for creating and sustaining such a product centric knowledge environment.” [Terzi et al., 07]

Ces différents buts montrent que l'approche PLM contribue à la maîtrise du triplet « qualité-coût-délai » pour le produit (cf. §2.1.1). Ces buts peuvent être structurés selon des niveaux stratégique, tactique et opérationnel. Le but stratégique vise à augmenter la profitabilité et à obtenir une meilleure place sur les marchés. Pour cela, l'entreprise doit être capable de fournir le bon produit, au bon moment et au bon coût. Ce but stratégique se décline en un but tactique qui consiste à créer et maintenir un environnement centré sur le produit. Ce dernier est lui-même décliné en un but opérationnel relatif à la perception informationnelle de l'entreprise. Il consiste à avoir la capacité de fournir une information pertinente au bon moment et à la bonne personne. A l'exception de certains auteurs, comme John Stark, qui prennent en compte les deux perceptions de l'entreprise de manière équivalente, la majorité des auteurs focalise leurs travaux sur la perception informationnelle.

4.1.5 Des écarts entre le nom et les définitions associées

Suite à cet état de l'art, force est de constater qu'il n'existe pas aujourd'hui de définition consensuelle du PLM. Une des explications peut être l'existence d'écarts entre le nom « Product Lifecycle Management » et les concepts qui lui sont associés. La traduction française de PLM est « gestion du cycle de vie du produit ». Le premier écart identifiable concerne le mot « cycle ». La définition du mot « cycle » est « *une suite de phénomènes se renouvelant dans un ordre immuable* » [Larousse, 13]. Si les étapes théoriques de la vie d'un produit sont toujours identiques, il n'en est pas de même des étapes réelles. Ainsi, un produit peut exister de l'expression du besoin au recyclage alors qu'un autre « mourra » à l'étape de conception du procédé si sa fabrication n'est pas rentable voire impossible. Dans les définitions du PLM ci-dessus, ce sont les étapes « réelles » du produit qui sont concernées. Ainsi, Product Life Management serait plus représentatif des approches décrites que Product Lifecycle Management. Le deuxième écart concerne le mot « product ». Le « Product Lifecycle Management » devrait assurer la gestion du produit. Ceci est uniquement le cas dans la définition de John Stark. Selon son point de vue, l'approche PLM remet en cause les produits, la structure organisationnelle, les

clients, les méthodes de travail, les informations, les systèmes d'information, etc. [Stark, 05]. L'approche PLM selon les autres auteurs concerne principalement le management des informations relatives au produit. Le terme approprié à ces définitions serait alors « Product Information System Management ». Enfin, le troisième écart concerne le singulier du mot « Product ». Ce mot désignant tous les produits de l'entreprise, le terme « Products Life Management » serait plus adapté.

4.1.6 La définition de l'approche PLM choisie dans le cadre de nos travaux

Pour la suite de nos travaux, nous définissons l'**approche PLM*** comme un cadre de référence identifiant des buts, des principes, des méthodes et d'outils adéquats pour mettre en œuvre un management centré sur les produits, sur la collaboration et sur les savoirs faire [Pinel et al., 11b]. Pour cette approche, nous déterminons un but stratégique qui est de maîtriser le produit en garantissant que les exigences sont respectées tout au long du cycle de vie ce qui rend robuste la perception qu'a l'entreprise de sa valeur ajoutée. Nous identifions également un but tactique qui est de capitaliser les connaissances relatives aux produits de l'entreprise. L'entreprise cherche ainsi à élever son expertise en déterminant, fiabilisant et exploitant ses compétences clés pour proposer des produits qui répondent aux attentes du marché. Enfin, un but opérationnel est identifié, celui de mettre en place un système d'information capable de coordonner les flux associés aux différentes étapes du cycle de vie. Cette coordination doit permettre de proposer les produits au bon moment (cf. figure 1.8). L'approche PLM est ainsi totalement en adéquation avec le but qui consiste à rendre l'entreprise capable de fournir le bon produit sur le bon marché au bon moment et avec le bon coût (cf. §2.2.2). Par contre, la robustesse recherchée pour assurer la maîtrise du produit semble être en opposition avec l'agilité opérationnelle que l'entreprise doit acquérir.

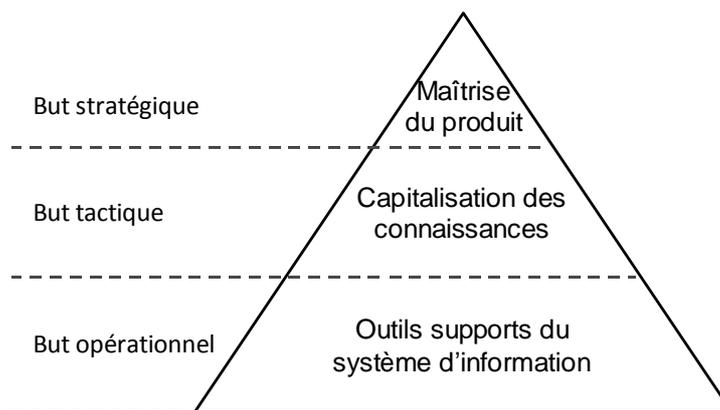


Figure 1.8 : Buts stratégique, tactique et opérationnel de l'approche PLM [Gautrot et al., 10]

4.1.7 Conclusion

Ce paragraphe a mis en lumière la difficulté de caractériser l'approche PLM et de choisir une définition consensuelle. Pour la suite de nos travaux, un état de l'art nous a permis de proposer une définition de l'approche PLM qui consiste à la percevoir comme un cadre sur lequel l'entreprise s'appuie pour maîtriser la gestion du cycle de vie de ses produits. La mise en œuvre de cette approche s'inscrit dans un projet que nous désignons projet PLM. Le paragraphe suivant présente les tenants et aboutissants liés à ce projet.

4.2 Les principes de l'approche PLM et la mise en œuvre du projet PLM

Définir de manière générique comment mettre en œuvre l'approche PLM au moyen d'un projet PLM s'avère délicat. En effet, les divergences entre les points de vue des auteurs et l'aspect holistique de l'approche PLM font que le périmètre à considérer est à la fois vaste et variable.

Ce paragraphe présente dans un premier temps un état de l'art qui détermine les principes attribués à l'approche PLM. Une analyse QQQCCP⁸ est ensuite réalisée pour déterminer les éléments à considérer dans le cadre du projet PLM. Enfin, les principales étapes du projet PLM sont identifiées.

4.2.1 Les principes de l'approche PLM

Après avoir défini les buts de l'approche PLM, nous nous intéressons ici à ses principes en partant d'un état de l'art des travaux suivants : [Ameri et al., 05] [Dutta et al., 05] [Stark, 05] [Sudarsan et al., 05] [Terzi et al., 07]. Ces principes peuvent être considérés comme des sous-but des buts identifiés dans le paragraphe précédent (cf. §4.1).

John Stark identifie plusieurs principes relatifs à la perception organisationnelle de l'entreprise pour atteindre les buts définis : se recentrer sur le produit, impliquer ses clients, écouter les commentaires des clients concernant les produits livrés, avoir une démarche écologique, construire une organisation transversale, créer des équipes de personnes hautement qualifiées, utiliser les technologies modernes, développer et maintenir une approche PLM cohérente, augmenter sans cesse les ventes et la qualité, réduire les cycles de temps et les coûts, avoir une bonne connaissance de son environnement et maintenir la sécurité.

Les autres auteurs identifient trois principes majeurs centrés sur la gestion des informations liées au produit (perception informationnelle). Le premier principe consiste à assurer un partage et une utilisation sécurisée des informations liées aux produits. Le deuxième principe consiste à maintenir dans le temps l'intégrité des informations relatives aux produits. Enfin, le troisième principe consiste à garantir la conformité entre les informations liées aux produits et la réalité évolutive qu'elles représentent.

Dans le détail de ces trois principes, la mise en œuvre d'« intégrations » revient de manière récurrente. L'intégration a pour objectif de « *combiner des parties pour faire un tout en maîtrisant les comportements émergents* » [Meinadier, 98]. Un système informatique intégré est quant à lui défini comme « *un système où une information n'est entrée qu'une seule fois et n'est stockée qu'à un endroit ou qui se comporte comme si elle n'était stockée qu'à un seul endroit. Cette information, une fois stockée, est accessible à tout moment par tous les programmes et tous les utilisateurs du système dotés des niveaux de sécurité adéquats* » [Tysebaert, 01]. Nous pouvons ainsi retenir comme définition que l'intégration est le fait de combiner différentes parties afin de créer un tout cohérent.

Ces différents principes illustrent l'aspect holistique de l'approche PLM. Ils mettent également en lumière le fait que les perceptions informationnelle et organisationnelle doivent être considérées de manière corrélée.

Une première catégorie d'intégration vise à créer une cohérence entre plusieurs éléments d'une même perception de l'entreprise. Pour la perception informationnelle, l'objectif est de créer un système d'information gérant de manière intégrée toutes les informations issues des étapes de la vie du produit. Pour la perception organisationnelle, l'objectif est de coordonner les différents métiers liés au cycle de vie. D'une part, la mise en œuvre de cette intégration implique de créer une entreprise étendue composée des acteurs amenant de la valeur ajoutée au produit. D'autre part, elle induit de rationaliser les processus mis en œuvre durant la vie du produit.

Une deuxième catégorie d'intégration vise à assurer la cohérence entre les éléments de la perception informationnelle et ceux de la perception organisationnelle de l'entreprise. Cette cohérence concerne des éléments stratégiques (gestion du portefeuille de produits), tactiques (gestion du projet de développement) et opérationnels (gestion des informations liées au produit) [Dutta et al., 05]. Cette catégorie concerne également différents éléments tactiques : les Technologies de l'Information, les processus métiers et les méthodes métiers [Terzi et al., 07]. Enfin, elle concerne différents éléments opérationnels : les personnes, les processus et les informations [Ameri et al., 05].

Les buts et les principes associés à l'approche PLM ayant été déterminés, leur mise en œuvre au moyen du projet PLM est maintenant présentée dans les deux paragraphes suivants.

⁸ Qui fait Quoi ?, Où ? Quand ? Comment ? Combien ? et Pourquoi ?

4.2.2 Le projet PLM : quels sont les éléments à considérer ?

John Stark attribue à l'approche PLM un périmètre très vaste [Stark, 05]. Pour lui, les éléments à considérer sont le produit, l'organisation de l'entreprise, les personnes, les méthodes métiers, les processus métiers, les informations relatives aux produits, les systèmes informatiques, les standards, les interfaces, etc. Le groupe de travail IWG-PLM [Terzi et al., 07] identifie pour l'approche PLM quatre piliers relatifs aux produits : les informations (pilier central), les processus métiers, les méthodes métiers et les technologies de l'information. D'autres auteurs [Ameri et al., 05] déterminent également quatre piliers relatifs aux produits : les connaissances (pilier central), les informations, les personnes et les processus métiers. Enfin, certains auteurs définissent que la mise en œuvre de l'approche PLM implique une grande panoplie de concepts, de technologies et d'outils [Saaksvuori et al., 08].

Pour structurer une synthèse de ces travaux, nous nous appuyons sur une analyse QQQCCP qui propose des réponses à chacune des questions (à l'exception des questions Pourquoi ? car elle a été traitée dans un paragraphe précédent (cf. §4.1.4) et de la question Combien ?) en se basant sur les buts et les principes de l'approche PLM⁹ déterminés précédemment.

Répondons aux questions **Où ?** et au **Quand ?**. L'approche PLM a pour périmètre l'entreprise étendue sans limite temporelle.

Puis vient la question **Quoi ?** Les « matières premières » de l'approche PLM sont les informations, les connaissances et les processus métiers relatifs aux produits.

La réponse à la question **Qui ?** se déduit des deux réponses précédentes : toute personne concernée par les informations ou les processus métiers relatifs aux produits dans l'entreprise étendue.

Intéressons-nous maintenant à la question **Comment ?** Pour créer un environnement centré sur le produit, il est nécessaire de s'appuyer sur les méthodes de mise en œuvre du système d'information (perception informationnelle). Ces méthodes impliquent une analyse des méthodes métiers relatives au produit utilisées dans l'entreprise (perception organisationnelle). Le groupe de travail IWG-PLM [Terzi et al., 07] classe ces dernières en quatre catégories. La première regroupe les méthodes utilisées pendant l'étape de développement du produit (l'analyse de la valeur, par exemple). La deuxième englobe les méthodes qui prennent en compte les besoins et contraintes issues des différentes étapes du cycle de vie du produit (Design for X, par exemple). La troisième regroupe les méthodes qui évaluent le produit selon des critères relatifs à une étape de la vie du produit (par exemple, l'Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité ou AMDEC). Enfin, la quatrième catégorie englobe les méthodes supportant l'amélioration continue (Six Sigma, par exemple).

Enfin, nous terminons par la question **Avec quoi ?** Pour supporter la gestion des informations, des connaissances et des processus métiers relatifs aux produits, l'approche PLM s'appuie sur plusieurs logiciels. Le groupe de travail IWG-PLM [Terzi et al., 07] définit ainsi le système informatisé PLM comme un système de systèmes informatisés. Le système PLM inclut aussi bien le logiciel PDM ou Product Data Management (« *a primary system component of PLM* ») que le logiciel ERP ou Enterprise Resource Planning (« *a fundamental component of PLM* »), soit toute application (définie comme « PLM agent ») qui traite des informations numériques relatives au produit. Ainsi, les « outils de l'approche PLM » sont les logiciels mais également les architectures et les standards utilisés par les logiciels pour communiquer entre eux et avec les systèmes extérieurs. Pour la suite de nos travaux, nous choisissons de nous appuyer, en l'adaptant, sur la définition du système d'information produit [Gzara, 00] pour définir le système PLM. Ainsi, nous considérons le **système PLM*** non pas comme un système de systèmes mais comme un système ayant des interfaces avec d'autres systèmes. Nous le définissons comme un système d'information supportant la création, la circulation, l'utilisation et l'évolution du patrimoine informationnel de représentation du produit. D'une part, ce patrimoine inclut les informations qui déterminent les exigences et les spécifications du produit et les informations qui définissent comment le produit doit être fabriqué, utilisé, maintenu et recyclé.

⁹ Les éléments concernés par l'approche PLM sont soulignés tout au long de l'analyse.

D'autre part, il inclut les informations qui décrivent le produit physique dans différents états (utilisé, fabriqué, maintenu, recyclé). Le support informatique de ce dispositif organisationnel est souvent réalisé au moyen d'un logiciel PLM.

Dans ce paragraphe, nous avons listé les éléments à considérer pour mettre en œuvre les principes de l'approche PLM. Les buts opérationnel, tactique et stratégique de l'entreprise, les produits, les connaissances, les méthodes métiers, les processus métiers et les personnes sont relatifs à la perception organisationnelle de l'entreprise. Le système d'information, les méthodes de mise en œuvre du système d'information, les logiciels, les architectures, les standards et les informations sont relatifs à la perception informationnelle.

La question qui se pose maintenant est : quels éléments doivent être remis en cause dans le cadre du projet PLM ? La réponse diffère selon les auteurs. John Stark considère que tous les éléments identifiés sont amenés à évoluer si cela est jugé nécessaire. Les autres auteurs focalisent sur le système d'information et les processus métier. D'une entreprise à l'autre, les changements impliqués par un projet PLM sont variables. Ils dépendent de la situation de départ de l'entreprise (sa culture, sa gestion des produits et des informations, son système d'information, etc.) et des buts qu'elle s'est fixée.

4.2.3 Le projet PLM : les différentes étapes

En identifiant les buts, les principes et les éléments relatifs à l'approche PLM, les paragraphes précédents ont présenté le cadre dans lequel s'inscrit le projet PLM. Dans ce paragraphe, l'objectif est de décrire les principales étapes « idéales » de ce projet qui découlent de ce cadre.

Tout d'abord, l'entreprise définit les buts attribués au projet. Ces buts doivent considérer les perceptions organisationnelle et informationnelle de l'entreprise de manière corrélée. Il convient ensuite de réaliser un état des lieux pour évaluer la situation de départ. Cet état des lieux met en lumière les séparations existant entre les deux perceptions de l'entreprise. Il identifie également les actions à mettre en œuvre pour atteindre les buts fixés. La réalisation de l'état des lieux peut s'appuyer sur des modèles de maturité spécifiques à l'approche PLM [Batenburg et al., 05] [Stark, 05] [Saaksvuori et al., 08]. Un modèle de maturité identifie des niveaux de maturité selon différents pôles d'observation jugés pertinents. Par exemple, le modèle peut déterminer quatre niveaux de maturité (traditionnel, en éveil, en adaptation, moderne) et différents pôles d'observation (stratégie produit, organisation, processus métiers, protection des données produit, transfert de données entre services, etc.) [Stark, 05].

Une fois les buts fixés et l'état des lieux réalisé, le projet PLM peut être mis en œuvre. Certains auteurs décrivent la gestion d'un projet PLM [Stark, 05] [Saaksvuori et al., 08] en précisant les étapes, les ressources, la rédaction du cahier des charges, le choix logiciel, la gestion du changement, etc. Par exemple, il est possible de structurer le projet PLM en quatre grandes étapes : le démarrage du projet, la définition du système PLM, la sélection du logiciel PLM et son déploiement et utilisation [Stark, 05].

4.2.4 Conclusion

Ce paragraphe recense les « principes » qui permettent de respecter la finalité de l'approche PLM. Cet état des lieux a mis en lumière qu'un produit est l'aboutissement d'un travail entre des acteurs, voire des organisations, s'appuyant sur des outils, réalisé selon des méthodes en exploitant des informations. Ces principes ont pour objectif d'assurer la cohérence entre ces différents éléments afin que le produit soit conforme au besoin du client et maintenu et recyclé en fonction des exigences fixées.

Ce paragraphe a également présenté la mise en œuvre de l'approche PLM via un projet PLM en décrivant les éléments à considérer et les étapes à suivre. Enfin, ce paragraphe a mis en lumière les spécificités du système PLM.

4.3 Bilan

Malgré de nombreux points communs entre les définitions existantes, aucune définition consensuelle n'a émergé jusqu'à présent. Au niveau des pratiques, les projets PLM industriels se multiplient. Après des années de gestation, le marché du PLM connaît ainsi une croissance qui devrait se poursuivre dans les années à venir¹⁰. Au niveau académique, l'existence de groupes de travail comme IWG-PLM et de conférences internationales sur le sujet prouve que le PLM peut être considéré comme un domaine scientifique pour lequel des chercheurs ont une volonté commune de définir des buts, des principes, des méthodes et des outils. John Stark considère¹¹ que la définition du PLM est une définition en évolution (« *the evolving definition of PLM* »), ce qui traduit bien le sentiment qui émane de l'état de l'art réalisé.

Dans ce paragraphe, une synthèse traduisant notre compréhension de ce qu'est l'approche PLM a été réalisée. Une définition de l'approche PLM a alors été proposée. Cette définition décrit l'approche PLM comme un cadre de référence pour un management centré sur les produits, sur la collaboration et sur les savoirs faire. Tout au long du paragraphe, les buts, les principes et les éléments relatifs à l'approche PLM ont été positionnés vis à vis des deux perceptions organisationnelle et informationnelle de l'entreprise. L'adoption de l'approche PLM a pour but de maîtriser la gestion du produit (but stratégique). Pour atteindre ce but, un but tactique est identifié : la capitalisation des connaissances relatives aux produits. Ce but tactique est lui-même décliné en un but opérationnel : la mise en œuvre d'un système PLM informatisé. Le système PLM est ainsi identifié comme la « clé de voûte » de l'approche PLM. La pertinence de la définition proposée au regard de nos travaux tient au fait que cette définition identifie clairement la hiérarchie de buts à laquelle est liée la mise en œuvre du système PLM.

Suite à cet état de l'art, l'approche PLM s'avère être en totale adéquation avec le but de fournir le bon produit sur le bon marché au bon moment et avec le bon coût identifié dans le paragraphe 2.2.2. Par contre, la robustesse recherchée pour assurer la maîtrise du produit semble être en opposition avec l'agilité opérationnelle que l'entreprise doit acquérir.

Après avoir défini les buts de l'approche PLM, c'est à dire les bénéfices que son adoption apporte, les questions qui se posent à présent sont les suivantes : quels sont les facteurs déclencheurs de la mise en œuvre d'un projet PLM ? La mise en œuvre d'un projet PLM est-elle devenue une nécessité pour toute entreprise ? Et plus particulièrement est-elle devenue une nécessité pour les PME sous-traitantes ? Pour répondre entre autres à ces questions, un groupe de travail constitué de chefs de projets PLM industriels et d'universitaires a été créé : le Club PLM. Ce groupe nous permet à la fois de faire un état des lieux des pratiques et d'approfondir des sujets relatifs à nos travaux.

5 La création du Club PLM

Dans le cadre de travaux de recherche relatifs au génie industriel, il nous semble indispensable d'avoir des échanges avec des chefs de projet PLM. En effet, ces échanges permettent d'acquérir une connaissance des pratiques du terrain. Ils servent également à identifier des questions en phase avec la réalité. Enfin, de leur synthèse découle l'élaboration de propositions génériques aux entreprises d'un domaine donné (et non pas à une entreprise). Ainsi, dans le cadre de nos travaux, nous avons pris la décision de créer un groupe de travail, le Club PLM regroupant des chefs de projet PLM.

A l'initiative de l'Université de Savoie et de Thésame (centre de ressources en mécatronique, gestion industrielle et management de l'innovation), le Club PLM a été créé en mars 2010 avec quinze entreprises fondatrices. L'animation et le pilotage de ce Club sont assurés par un parrain membre du club et coopté par les membres du Club. Lors de sa création, il a été souhaité que ce réseau n'intègre pas d'éditeurs ou d'intégrateurs. Ce choix est guidé par le désir de ne pas entraver la liberté d'expression mais également par le souhait de créer un nouvel angle de vue propre aux industriels et

¹⁰ Scherer M, *CAO: les nouvelles stratégies des utilisateurs de logiciels*, l'Usine Nouvelle, 2010.

<http://www.usinenouvelle.com/article/cao-les-nouvelles-strategies-des-utilisateurs-de-logiciels.N142756>

¹¹ <http://www.johnstark.com/> le 20 juin 2011

aux universitaires. La confidentialité au sein du Club est garantie par l'adhésion à une charte qui fixe les principes en la matière.

Le Club PLM est la réponse concrète à l'expression de deux besoins.

Le premier besoin est celui de chercheurs qui désirent avoir une recherche en phase avec les problématiques du terrain. Le premier objectif est de récolter des données via des réunions, des présentations, des questionnaires et des interviews. Le deuxième objectif est de construire des propositions qui soient génériques à un domaine. Aujourd'hui, le Club regroupe vingt-cinq entreprises, deux laboratoires de recherche et Thésame. Ces entreprises ont une grande diversité de tailles, de produits et de maturité dans le projet PLM. Cependant, force est de constater qu'au-delà de ces différences, des échanges autour de concepts communs sont réalisés de manière constructive.

Le deuxième besoin ayant mené à la création du Club est celui des chefs de projets PLM qui souhaitent élargir leur réflexion et compétences dans la conduite de projets PLM. La présence de multiples acteurs régionaux permettant de se comparer, d'échanger, de dialoguer au-delà du noyau dur incontournable (entreprise - intégrateur - éditeur) est une opportunité dont ils se sont saisis en créant un réseau d'entreprises. Les expériences vécues alimentent une base de réflexion autour du PLM. La richesse amenée par la diversité permet à chacun d'avoir une vision plus exhaustive des risques, des potentialités et des facteurs de succès des projets PLM.

Ainsi, le Club PLM n'a pas vocation à identifier de « bonnes pratiques » que toute entreprise doit appliquer. Ses objectifs sont d'une part de formuler des problématiques autour du PLM et, d'autre part, d'identifier des éléments de réponse. Pour illustrer que le projet PLM est une expérience humaine propre à chaque entreprise, la devise du Club PLM pourrait ainsi être « *Avance sur ton chemin, car il n'existe que par ta marche* (Saint-Augustin) ». L'entrée dans le Club était initialement réservée aux entreprises ayant un logiciel PLM. Cette restriction a aujourd'hui été levée pour deux raisons. La première raison est que, face à la complexité des logiciels existants, certaines entreprises font le choix de développer leur propre solution informatique. La seconde raison est le désir d'accueillir les entreprises en avant-projet afin qu'elles tirent profit au plus tôt des enseignements du Club et que le Club bénéficie des réflexions de personnes non influencées par l'utilisation des logiciels.

La première étude menée par les membres du Club a eu pour but d'identifier les facteurs déclencheurs et les objectifs associés aux projets PLM des différents membres. Cette étude nous a permis d'apporter des éléments de réponse à la question : La mise en œuvre d'un projet PLM est-elle devenue une nécessité pour les PME sous-traitantes ?

6 La mise en œuvre d'un projet PLM est-elle devenue une nécessité pour les PME sous-traitantes ?

La réponse à cette question est basée sur les témoignages de chefs de projet industriels et sur les résultats d'un questionnaire distribué au sein du Club PLM. A partir d'une liste synthétique de facteurs déclencheurs des projets PLM, nous avons identifié et développé ceux inhérents aux entreprises de sous-traitance de la vallée de l'Arve¹².

Parmi les facteurs déclencheurs possibles pour ces entreprises sous-traitantes, deux familles ont été définies.

La première famille regroupe les facteurs issus de la définition de nouveaux buts stratégiques de l'entreprise qui adopte un mode proactif pour faire face à la concurrence. Ainsi, l'entreprise se lance sur de nouveaux marchés, sur des produits plus complexes, elle passe d'un statut de sous-traitant de fabrication à un statut de sous-traitant participant à la conception du produit, etc. Cette famille de facteurs inclut ceux relatifs à la réorganisation de l'entreprise avec la mise en œuvre de filiales.

La deuxième famille regroupe les facteurs issus d'une adaptation de l'entreprise à son environnement. L'entreprise s'adapte aux demandes de ses clients (de plus en plus de versions de documents, des

¹² Ce travail a été réalisé dans le cadre d'une réunion avec le CTDec (Centre Technique de l'industrie du Décolletage – <http://www.ctdec.com/>) le 25/01/2011 .

documents plus complexes à fournir, des temps de développement plus courts, etc.), aux réglementations (réponse aux règlements Reach, etc.), aux besoins des utilisateurs (gestion de fichiers informatiques de plus en plus nombreux, gestion et sécurisation des informations contractuelles).

Tous les projets PLM industriels étudiés ont un but opérationnel commun : mettre en œuvre un logiciel PLM qui assure la centralisation et le partage des informations relatives aux produits. Cependant, face aux résultats de notre étude, force est de constater que la diversité des facteurs déclencheurs est grande. Cette diversité et la nature des facteurs identifiés laissent à penser que le panel d'entreprises concerné dans le futur par la mise en place d'un système PLM est important. Ce constat est en adéquation avec les chiffres en constante croissance du marché du PLM¹³.

Dans le cadre d'une analyse d'opportunité d'une approche PLM, il est à noter que l'entreprise peut s'appuyer sur le questionnaire¹⁴ créé par l'ENE (Espace Numérique Entreprises)¹⁵.

Pour conclure cette étude, il nous semble utile de préciser qu'en complément d'un facteur déclencheur, l'existence d'un « sponsor » est un des éléments indispensables à la mise en œuvre d'un projet PLM. Ce dernier est un membre du comité exécutif de l'entreprise qui promeut le projet auprès des instances de décision. Il choisit le chef de projet PLM et devient son allié face à toutes les difficultés. Il est l'avocat et le défenseur du projet PLM.

7 Conclusion : formulation du problème et identification de deux objectifs

Nos travaux ont pour sujet la définition et la construction du système PLM. Pour appréhender les tenants et les aboutissants de cette mise en œuvre, il était nécessaire d'identifier la hiérarchie de buts dans laquelle elle s'inscrit. Ce chapitre a donc présenté les buts de l'entreprise et ceux associés à l'approche PLM.

Tout d'abord, le contexte de l'entreprise a montré que son environnement est de plus en plus concurrentiel. Il devient alors impératif pour elle de se montrer proactive en identifiant des buts stratégiques pertinents et en mettant en œuvre des projets d'entreprise pour atteindre ces buts. Le paragraphe 2.2.2 a identifié deux buts majeurs et interdépendants de l'entreprise. Le premier but consiste à fournir le bon produit sur le bon marché au bon moment et avec le bon coût. Le deuxième but consiste à développer l'agilité opérationnelle qui permet de faire face à un environnement très changeant et de mettre en œuvre l'innovation. Pour rappel, l'agilité opérationnelle qualifie le délai de mise en œuvre d'une évolution opérationnelle, c'est à dire à la fois organisationnelle et informationnelle. Ces deux perceptions correspondent à deux pôles d'observation de la réalité opérationnelle de l'entreprise. Le pôle d'observation organisationnel considère les hommes et les processus qui fournissent les produits et les services. Le pôle d'observation informationnel considère les services automatisés qui manipulent les informations pour réaliser des fonctions. Lorsque ces deux perceptions ne sont pas prises en compte de manière conjointe dans les projets d'entreprise, force est de constater que ces derniers se soldent souvent par un échec.

Suite à cette mise en contexte, nous avons présenté l'approche PLM. Cette approche constitue un cadre de référence pour mettre en œuvre un management centré sur les produits, sur la collaboration et sur les savoirs faire. Elle définit des buts, des principes, des méthodes et des outils pour mettre en œuvre un projet d'entreprise PLM. Cette approche identifie trois buts dont un but stratégique qui vise à maîtriser le produit. Le but tactique consiste à capitaliser les connaissances relatives aux produits de l'entreprise. Le but opérationnel consiste quant à lui à construire le système PLM qui est nécessaire à l'atteinte des buts tactique et stratégique identifiés. La définition de l'approche PLM nous a permis de constater qu'elle répond au but de fournir le bon produit sur le bon marché au bon moment et avec le

¹³ Scherer M, CAO: les nouvelles stratégies des utilisateurs de logiciels, l'Usine Nouvelle, 2010.

<http://www.usinenouvelle.com/article/cao-les-nouvelles-strategies-des-utilisateurs-de-logiciels.N142756>

¹⁴ diagnostic flash PLM de l'ENE

<http://www.ene.fr/actualites/plm/boite-outils-plm-1-6-diagnostic-flash-plm,371.html>

¹⁵ L'ENE est une association de conseil et d'assistance sur les projets au numérique. Elle fait partie des organisateurs de la journée du PLM. <http://www.journeedulplm.fr/>

bon coût. Ainsi, le projet PLM, et donc la mise en œuvre d'un système PLM au sein de l'entreprise, font bien partie des moyens par lesquels l'entreprise peut atteindre ce but. Par contre, la robustesse recherchée pour assurer la maîtrise du produit semble être en opposition avec l'agilité opérationnelle que l'entreprise doit acquérir.

Le dernier paragraphe présente la création d'un groupe composé de chefs de projet PLM industriels et d'universitaires, le Club PLM. Ce groupe nous permet de faire des états des lieux des pratiques industrielles ; il constitue notre sujet d'étude pour la suite de ces travaux. Une première étude des facteurs déclencheurs des projets PLM a ainsi pu être réalisée. Cette étude avait pour objectif d'identifier les chances d'opportunité de mise en œuvre de ces projets par les entreprises PME sous-traitantes. Ses résultats mènent au constat que le panel d'entreprises concernées dans le futur par la mise en place d'un système PLM devrait être important.

Ce chapitre a montré que la construction d'un système PLM doit satisfaire aux deux buts majeurs identifiés de l'entreprise. Ces deux buts sont d'avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût et de développer l'agilité opérationnelle.

Le problème identifié est ainsi : comment réaliser cette construction ?

L'hypothèse que nous faisons dans ces travaux est qu'il est nécessaire de définir une méthode adaptée à la construction du système PLM. Le chapitre 2 présente ainsi les bases que nous avons utilisées pour bâtir la proposition présentée dans la suite de ce mémoire.

Chapitre 2

Les bases de notre proposition

Sommaire

- 1 Introduction
 - 2 Du besoin de modéliser au besoin de paradigmes
 - 3 Les paradigmes à la base du système de représentation
 - 4 Les cadres de modélisation
 - 5 Conclusion
-

1 Introduction

La conclusion formulée à la fin du chapitre 1 montre qu'il est nécessaire de définir une méthode adaptée à la construction du système PLM. Une méthode de construction d'un système d'information préconise un ensemble de modèles et un processus appropriés pour l'objectif fixé [Rieu, 99]. Ainsi, la modélisation est au centre de nos préoccupations. Afin d'établir les bases de notre proposition, ce chapitre présente les différentes définitions utilisées dans ces travaux.

Dans un premier temps, ce chapitre recense différentes définitions liées au processus de modélisation (§ 2). Il présente ensuite les paradigmes qui sont à la base du système de représentation utilisé pour construire des modèles (§ 3). Enfin, ce chapitre définit le concept de cadre de modélisation et il présente les différents axes d'étude qui lui sont associés dans les travaux existants mais également dans les travaux de ce mémoire (§ 4).

2 Du besoin de modéliser au besoin de paradigmes

Le chapitre 1 identifie les différentes phases du cycle de vie du produit. Chaque phase englobe une ou plusieurs étapes. Chaque étape implique différents acteurs qui ont tous leurs propres représentations du produit qu'il soit théorique ou physique. Ainsi, chaque acteur possède un « niveau de visibilité » du produit qui lui est propre. Le **niveau de visibilité*** d'un acteur détermine la représentation que ce dernier a du produit, il est défini en fonction des actions que l'acteur doit réaliser sur le produit. Le schéma suivant (cf. figure 2.1) illustre ces propos en identifiant d'une part quelques acteurs intervenant tout au long du cycle de vie du produit et, d'autre part, certaines des représentations du produit qu'ils utilisent. A travers ce schéma, nous pouvons constater que la notion de niveau de visibilité peut être étendue à une étape du cycle de vie. Ainsi, pour maîtriser le management du cycle de vie du produit, il est nécessaire de gérer les différentes représentations utilisées par les acteurs.

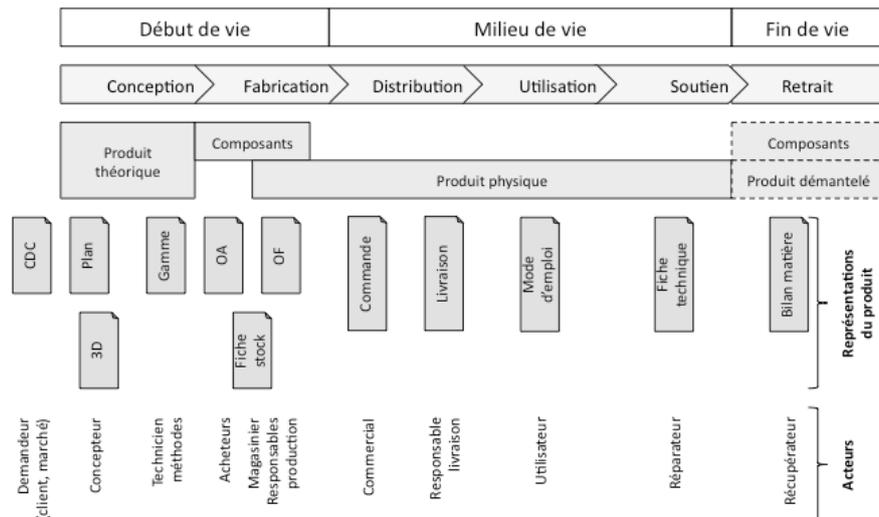


Figure 2.1 : Différentes représentations du produit en fonction des acteurs

2.1 La modélisation des produits

La définition du niveau de visibilité nous conduit à nous intéresser au processus qui permet de construire les différentes représentations du produit. La **modélisation*** est définie comme l'action de construire, de manière intentionnelle, des modèles susceptibles de représenter un objet ou une situation donnée pour le ou la rendre explicite ou pour rendre compte des effets d'actions potentielles sur cet objet ou cette situation [Le Moigne, 99]. La modélisation est donc définie comme un processus dont l'objectif est de produire un **modèle*** qui est une représentation intelligible, artificielle et partielle d'un objet ou d'une situation.

Ainsi défini, le modèle repose sur une double métaphore¹⁶, celle de la *représentation théâtrale* et celle de la *représentation diplomatique*. La première suggère l'idée d'exposer, sous une forme concrète, un objet ou une situation. L'objet ou la situation ainsi exposé peut être existant : modèle AS-IS, ou projeté : modèle TO-BE. La seconde métaphore suggère le transfert d'attribution en vertu duquel une personne peut agir en nom et place d'une autre. Le modèle est alors utilisé pour vérifier que l'objet ou la situation envisagé respecte les propriétés qui lui ont été assignées.

La production d'un modèle est toujours réalisée pour répondre à un objectif précis [AFIS, 12] [Genelot, 01]. Cet objectif dépend de l'étude que le modélisateur souhaite mener à partir d'un objet ou d'une situation donnée. Par exemple, le concepteur représente un produit par un plan pour définir ses dimensions, ses caractéristiques physiques et bien d'autres propriétés. Par contre, s'il souhaite vérifier que le produit conçu respecte certaines propriétés de flexion, il va représenter le produit par un modèle numérique de la structure.

Il est à noter que, suivant les cas, le modèle concerne soit un objet particulier ou une situation particulière, soit une classe d'objets ou de situations. Par classe d'objets ou de situations, il faut comprendre un ensemble d'objets ou de situations qui sont perçus comme ayant des caractéristiques et des propriétés communes. Dans ce cas, la représentation d'un objet ou d'une situation particulière est obtenue par un mécanisme d'instanciation. Dans ce mémoire, nous réservons le terme **modèle** pour désigner une représentation d'une classe d'objets ou d'une classe de situations et nous utilisons le terme **représentation**¹⁷ pour désigner un objet particulier ou une situation particulière.

¹⁶ Cette comparaison s'inspire de celle proposée par Jean Ladrière dans son article « Représentation et connaissance » (<http://www.universalis.fr/encyclopedie/connaissance/1-representation-et-connaissance/>).

¹⁷ Dans de nombreux travaux, la représentation d'un objet particulier ou d'une situation particulière est désignée « instance ». Ce terme a selon nous une connotation trop informatique.

2.2 La mission du système PLM

La première représentation du produit est un cahier des charges (CDC). Chaque étape de la vie du produit a pour objectif de participer à la transformation de ce CDC en un produit physique utilisé, maintenu et recyclé. La vie du produit peut être ainsi découpée en deux parties. Durant la première partie, le produit n'existe qu'à l'état théorique (conception du produit, conception du procédé de fabrication, etc.). Les exigences de la représentation fournie par une étape sont utilisées à l'étape suivante comme données d'entrée. Ainsi, elles servent de base de travail pour créer une représentation théorique du produit à un état plus mature. Par exemple, l'étape de conception du procédé crée la « nomenclature du produit pour fabrication » à partir de la « nomenclature du produit conçu ». Durant la deuxième partie de sa vie, le produit existe à l'état physique (fabrication, utilisation, maintenance, etc.). Chaque étape a pour objectif d'utiliser les exigences d'une représentation théorique pour fabriquer ou maintenir le produit physique. Pour répondre aux exigences de traçabilité et pour réaliser un suivi qualité, l'étape peut également avoir pour objectif d'enregistrer les propriétés du produit physique obtenu sous forme de représentation. Par exemple, l'étape de fabrication fabrique un produit physique à partir de la « nomenclature pour fabrication ». Elle peut également avoir pour but d'enregistrer la « nomenclature fabriquée » représentant le produit physique réalisé.

Dans le paragraphe 3.3 du chapitre 1, trois processus transversaux ont été identifiés. Le premier processus consiste à fournir un produit physique répondant au cahier des charges du client. Le deuxième processus consiste à analyser les conséquences d'une évolution d'une représentation du produit. Le troisième processus consiste à assurer la traçabilité tout au long des étapes du cycle de vie.

Ainsi, le système PLM doit assurer la cohérence et la traçabilité entre les différentes représentations du produit et il doit supporter l'analyse d'impact de la modification d'une des représentations sur les autres représentations. L'objectif recherché est de fournir à chaque acteur une représentation pertinente du produit ; cette représentation est construite en fonction du niveau de visibilité de l'acteur. Ces représentations concernent soit le produit théorique, soit le produit physique. Les représentations du produit sont désignées de la façon suivante dans la suite de ce mémoire :

Cycle de vie	Définition
Niveau de visibilité de l'étape	Ensemble des représentations du produit propres à une étape du cycle de vie.
Niveau de visibilité de l'acteur	Ensemble des représentations du produit propres à un acteur intervenant dans une étape.
Représentation du produit théorique	Représentation du produit théorique (modèle TO BE)
Représentation du produit physique	Représentation du produit physique (modèle AS IS)

Tableau 2.1 : Définitions des niveaux de visibilité et des représentations du produit

2.3 Les langages et la métamodélisation

L'élaboration d'un modèle nécessite un **langage*** qui est un ensemble de symboles (vocabulaire), de règles de composition des symboles (syntaxe) et de sens attribués aux symboles (sémantique).

Il est classique d'identifier trois types de langages :

- les langages naturels : un langage naturel autorise des représentations qui sont compréhensibles par une communauté de personnes (personnes parlant la langue française, par exemple). Le langage naturel offre souvent différentes solutions pour représenter une même situation. Toutefois, cette richesse de langage peut être à l'origine de redondances (pléonasmes, par exemple) ou d'ambiguïtés (polysémies, par exemple). Dans les travaux de ce mémoire, le langage naturel est utilisé pour exprimer des exigences associées au produit ou au système PLM (cf. chapitre 3).
- les langages semi-formels : un langage semi-formel s'appuie sur un ensemble fini de symboles (souvent graphiques) et une syntaxe bien définie qui peut être lisible par un ordinateur. Il est à

remarquer que les symboles étant dénotés par du texte, ce type de langage ne rend pas impossible les ambiguïtés. Dans les travaux de ce mémoire, le langage UML (Unified Modeling Language) est utilisé pour compléter la description des exigences et pour construire les modèles nécessaires d'une part à la gestion du produit et, d'autre part, au déploiement du système PLM (cf. chapitres 4 et 5).

- les langages formels : un langage formel est un langage doté d'une sémantique mathématique qui d'une part garantit l'absence d'ambiguïtés et, d'autre part, permet des raisonnements pour soit découvrir d'éventuelles incomplétudes ou inconsistances, soit prouver des propriétés [Chapurlat, 07]. Ces langages peuvent être utilisés pour construire des représentations du produit. Par exemple, la maquette numérique d'un produit est construite à partir d'un langage formel (cf. chapitre 1, § 4.1.1). Dans les travaux de ce mémoire, nous n'utilisons pas de langage formel pour la construction du système PLM.

Suite à ces définitions, il est possible de préciser celle du niveau de visibilité d'un acteur. Ce dernier est déterminé comme l'ensemble des représentations du produit construites à partir du langage propre à la famille d'acteurs à laquelle l'acteur concerné appartient. Dans le chapitre 1 (cf. chapitre 1, § 3.5), il a été montré que l'entreprise peut être perçue à travers plusieurs pôles d'observation, chaque pôle d'observation correspondant à une famille d'acteurs. Chaque famille d'acteurs s'appuie sur un langage spécifique. Afin de raisonner sur les interactions entre les familles d'acteurs des différentes étapes du cycle de vie, il convient ainsi d'étudier les correspondances entre les symboles des langages utilisés. L'étude de ces correspondances nécessite de représenter le langage de modélisation employé par chaque famille d'acteurs.

La **métamodélisation*** est définie comme l'action de construire le modèle d'un langage de modélisation ; ce modèle est désigné **métamodèle*** [Favre et al., 06]. Plus précisément, le métamodèle identifie les différents symboles du langage (vocabulaire) et les règles d'organisation de ces symboles (syntaxe). Le métamodèle ne fournit pas d'indication sur la sémantique [Blanc, 05]. Par définition, un modèle est le résultat de l'instanciation d'un métamodèle.

Ces différentes définitions introduisent trois niveaux de modélisation. La figure 2.2 illustre ces niveaux en prenant comme exemple de situation un produit théorique. Dans l'exemple suivant, le métamodèle permet de construire un modèle composé de classes et de relations entre ces classes. Le modèle définit que les produits (une classe) sont composés (une relation) de produits (une classe) et qu'ils sont normalisés (une relation) par des normes (une classe). A partir de ce modèle, des représentations sont créées pour décrire les produits théoriques de l'entreprise. Dans l'exemple suivant, la représentation de la table F7895 (produit théorique) est définie comme suit : la table F7895 est composée du plateau X2548 et de pieds P8975 et le plateau X2548 est normalisé selon la norme 7854576.

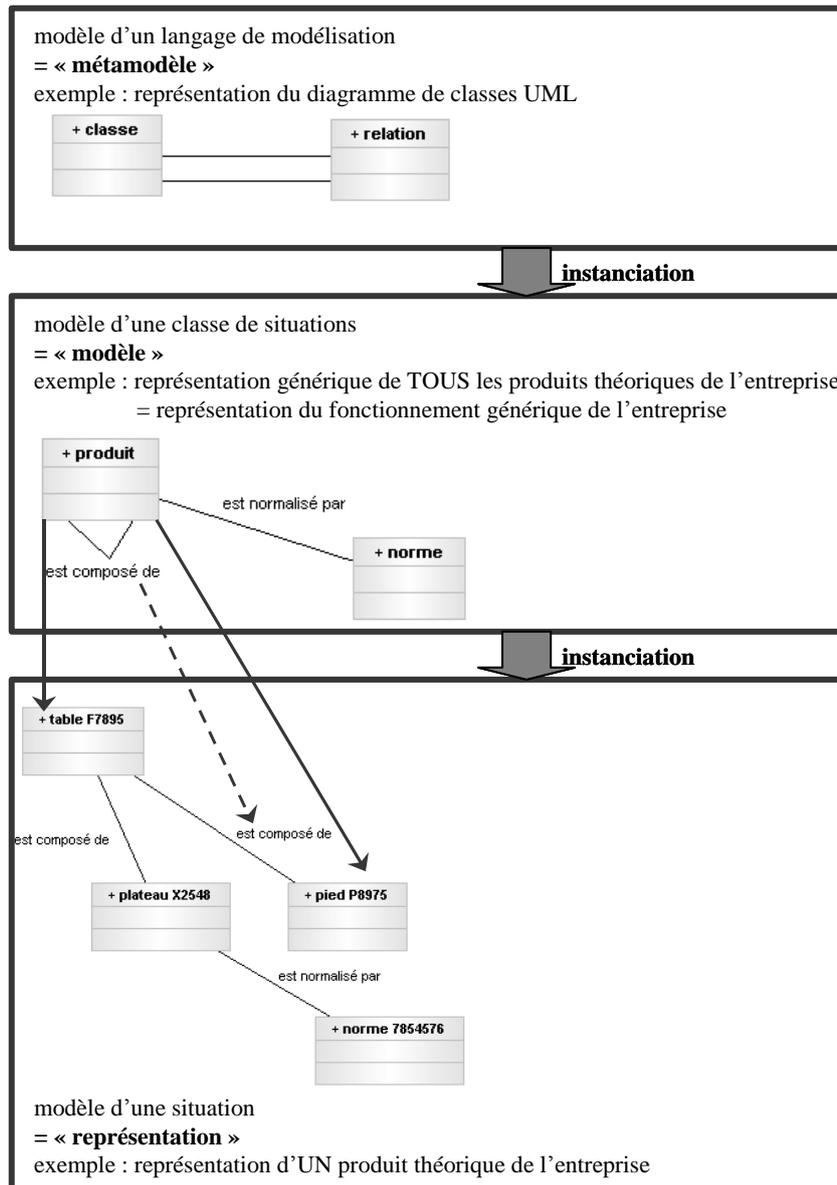


Figure 2.2 : Mécanisme d'instanciation entre métamodèle, modèle et représentation

Pour les travaux de ce mémoire, nous proposons ainsi de retenir les définitions suivantes :

Type de modèles	Définition
Modélisation	Action de construction intentionnelle d'un modèle d'une classe de produits ou d'un modèle du système PLM
Modèle	Représentation intelligible, artificielle et partielle d'une classe de produits ou du système PLM à un instant donné ; le modèle est construit à partir d'un langage de modélisation.
Représentation	Représentation intelligible, artificielle et partielle d'un produit particulier à un instant donné ; l'instance est construite à partir d'un modèle

Tableau 2.2 : Définitions de la modélisation, du modèle et de la représentation

Ces trois niveaux de modélisation peuvent être mis en parallèle de ceux préconisés par l'ingénierie dirigée par les modèles ou IDM. L'IDM identifie les quatre niveaux suivants (cf. figure 2.3) [Bezivin et al., 02] [Favre, 04]:

- **Le niveau M0** est composé des informations et des processus qui décrivent un objet ou une situation. Dans le cas du système PLM, ce niveau correspond aux représentations du produit et

aux processus relatifs à la gestion du cycle de vie du produit (processus déclenché pour réaliser un traitement dans une des étapes du cycle de vie du produit).

- **Le niveau M1** est composé de *modèles* qui décrivent une classe d'objets ou une classe de situations. Par exemple, les modèles peuvent être construits avec le langage UML [Rumbaugh et al., 04]. Dans le cas du système PLM, ce niveau correspond au modèle générique de tous les produits de l'entreprise et au modèle générique de tous les processus.
- **Le niveau M2** est composé des langages de définition des modèles, appelés aussi *métamodèles*. Ce niveau englobe, par exemple, les métamodèles définissant les différents diagrammes du langage UML (diagramme de classes, diagramme de séquences, diagramme de cas d'utilisation, etc.). Dans le cas du système PLM, ce niveau correspond aux métamodèles décrivant les langages de modélisation utilisés pour construire les différents modèles.
- **Le niveau M3** est composé d'une unique entité qui est le langage unique de définition des métamodèles, aussi appelé le *méta-métamodèle* : MOF (Meta-Object Facility). Ce méta-métamodèle est réflexif, le langage qu'il définit est utilisé pour représenter le méta-métamodèle¹⁸. Dans le cas du système PLM, le langage de ce niveau est utilisé pour créer les métamodèles.

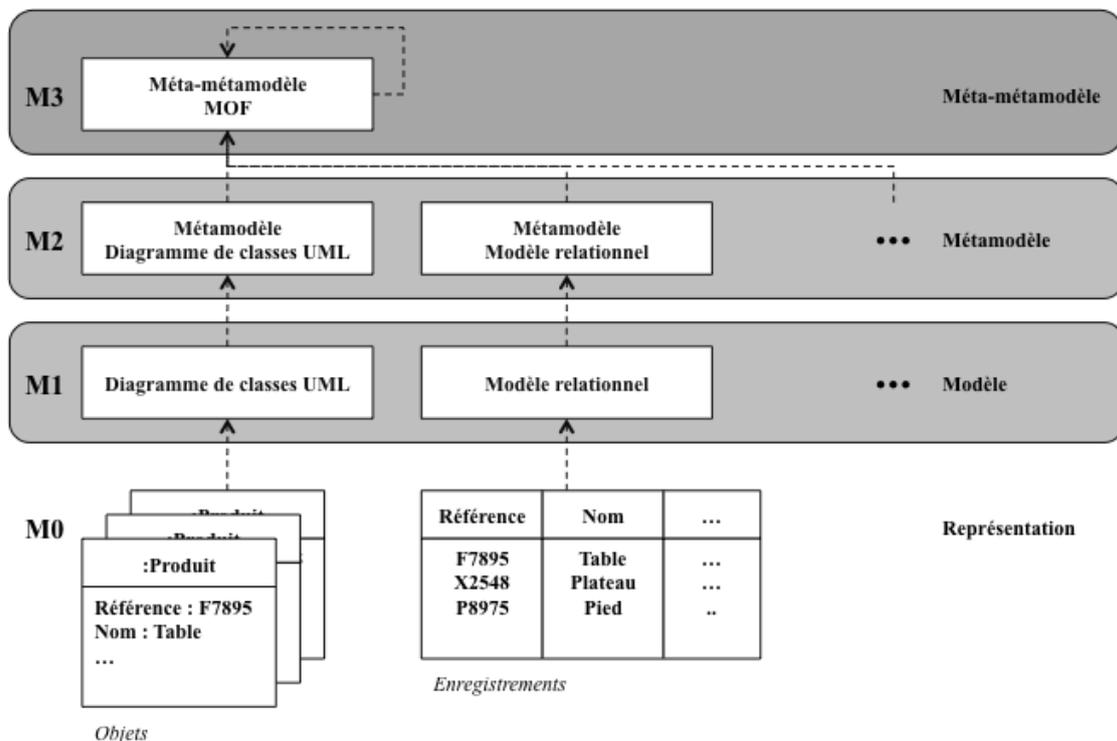


Figure 2.3 : Les quatre niveaux de modélisation préconisés par l'OMG (adapté de [Huget, 10])

2.4 Une nouvelle définition du système PLM

Ces différentes définitions permettent de compléter la définition du système PLM proposée dans le chapitre 1. Rappelons que le système PLM a été défini comme un système d'information supportant la création, la circulation, l'utilisation et l'évolution du patrimoine informationnel du produit (cf. chapitre 1, §4.2.2). Ce patrimoine englobe l'ensemble des connaissances, des informations et des données utilisées dans les différentes étapes du cycle de vie du produit.

Une **connaissance** est une croyance, une conviction personnelle justifiée qui accroît le potentiel d'un individu ou d'un groupe d'individus pour l'action. Une **information** est un stimulus qui produit un effet sur le comportement d'un individu ou sur son état cognitif [Reix, 11]. Une **donnée** est une

¹⁸ Cette réflexivité induit que le niveau M3 est le dernier niveau préconisé par l'OMG.

représentation d'une caractéristique d'un objet ou d'une situation décrite dans un code convenu (par exemple, valeur numérique, chaîne de caractères, valeur temporelle, etc.). Elle est définie pour répondre à un besoin et elle permet de conserver une trace pour une exploitation immédiate ou future [Melèse, 90].

Le système PLM ne gère ni les connaissances, ni les informations. Il gère leurs représentations. Dans les travaux de ce mémoire, nous nous focalisons sur la représentation des informations. Ainsi, la proposition du chapitre 4 ne prend pas en compte la modélisation des connaissances des produits. Les informations peuvent être modélisées par des données, des images, des sons, des textes, des plans, etc. La représentation obtenue est transformée en information par un processus, désigné processus interprétatif, qui a pour objectif de lui attribuer un sens. Le processus interprétatif est donc propre à un individu ou un groupe d'individus [Reix, 11].

Le système PLM peut donc être défini comme un système socio-technique qui a pour finalité de gérer les représentations du produit utiles aux acteurs intervenant tout au long du cycle de vie du produit. Ces acteurs peuvent être des acteurs internes à l'entreprise ou des acteurs externes dans un contexte d'entreprise étendue. Le système PLM doit donc fournir la bonne représentation au bon acteur au bon moment. En d'autres termes, le système PLM a pour finalité de construire le niveau de visibilité adapté à l'activité que l'acteur doit réaliser.

Le périmètre du système PLM est donc défini par l'ensemble des parties prenantes qui produisent ou utilisent des représentations du produit. Les acteurs sont des utilisateurs actifs qui bénéficient de manière « interactive » des services fournis par le système PLM.

La structure du système PLM comprend un système informatique qui contribue à la réalisation de sa finalité. Le système informatique, désigné logiciel PLM, collecte, mémorise, transforme, diffuse les représentations numériques des produits. Ce logiciel PLM est souvent une solution (libre ou privée) fournie par un éditeur ou une collectivité.

Il est important de distinguer les représentations manipulées par les différents acteurs de celles manipulées par un système informatique. Aussi, nous complétons comme suit les définitions proposées dans le paragraphe précédent :

Cycle de vie	Définition
Représentation du produit théorique	Représentation du produit théorique (modèle TO BE) manipulée par un acteur
Représentation du produit physique	Représentation du produit physique (modèle AS IS) manipulée par un acteur
Représentation numérique du produit théorique	Représentation du produit théorique (modèle TO BE) numérisée pour être manipulée par un système informatique
Représentation numérique du produit physique	Représentation du produit physique (modèle AS IS) numérisée pour être manipulée par un système informatique

Tableau 2.3 : Définitions des représentations du produit

2.5 La nécessité de paradigmes

Le paragraphe précédent montre que le système PLM doit gérer des représentations du produit. Ces représentations sont construites à partir d'un modèle décrivant la classe des produits de l'entreprise. Les questions qui se posent à présent sont : Quels modèles doit-on utiliser pour décrire une classe de produits ? Quels modèles doit-on utiliser pour construire le système PLM ? (cf. figure 2.4)

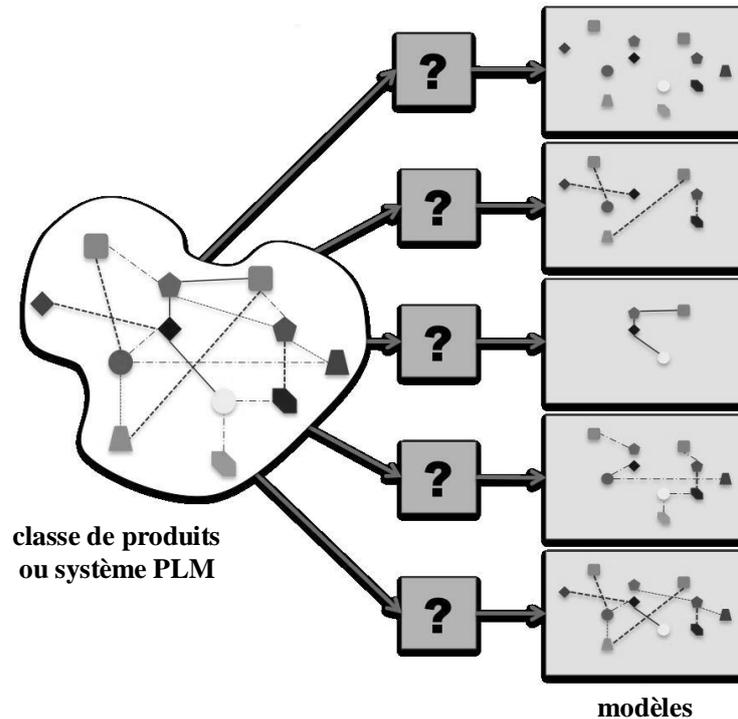


Figure 2.4 : Quels modèles pour représenter une classe de produits ou le système PLM adapté de [Genelot, 01]

Pour guider la création de ses modèles et faciliter leur interprétation par autrui, le modélisateur choisit un ou plusieurs paradigmes.

3 Les paradigmes à la base du système de représentation

Un **paradigme*** définit une conception théorique dominante ayant cours à une époque donnée dans une communauté scientifique donnée. Il supporte les types d'explications envisageables et les types de faits à découvrir dans une science donnée [CNRTL, 13]. Comme le souligne Edgar Morin [Morin, 80], un paradigme n'explique pas, mais il permet l'explication. Au minimum c'est un « pense-bête »; au maximum c'est un pense-intelligent qui nous aide à concevoir la complexité. Un nouveau paradigme ne remplace pas forcément le ou les paradigmes existants. Plusieurs paradigmes peuvent ainsi être utilisés conjointement s'ils ne sont pas en contradiction. Par extension, nous identifions un paradigme comme une manière de percevoir le monde.

Le choix d'un ou plusieurs paradigmes constitue une manière de percevoir la réalité à étudier. Ainsi, raisonner selon un paradigme donné induit l'utilisation de modèles en conformité avec ce paradigme (cf. figure 2.5).

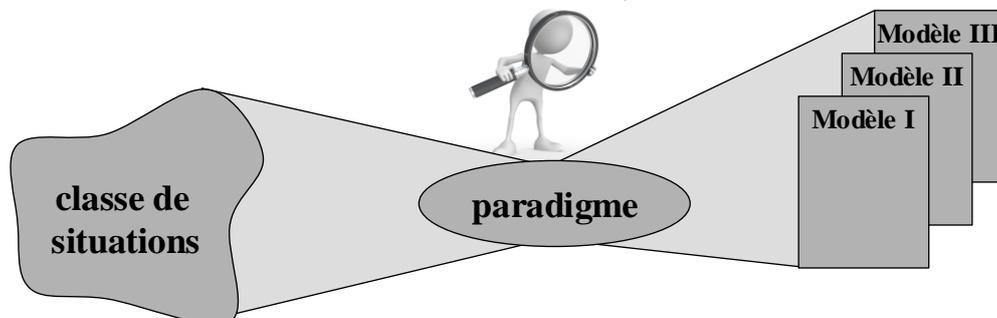


Figure 2.5 : Relations entre classe de situations, paradigme et modèles (adapté du systémo-graphie de Jean-Louis Le Moigne [Le Moigne, 90])

L'utilisation de paradigmes induit le processus de modélisation suivant (inspiré de [Genelot, 01] et [Le Moigne, 90]) :

- choix : identification du (ou des) paradigme(s) adapté(s) à l'étude envisagée sur la classe de situations.
- cadrage : définition des modèles adaptés à l'étude envisagée en identifiant leurs relations.
- développement : construction des modèles par identification des caractéristiques perçues sur la classe de situations étudiée.
- interprétation : utilisation des modèles pour expliquer une situation ou pour étudier les effets d'une intervention sur cette dernière.

Dans les travaux de ce mémoire, nous nous appuyons sur deux paradigmes : le paradigme systémique et le paradigme d'ambivalence.

3.1 Le paradigme systémique

Le **paradigme systémique*** conduit le modélisateur à percevoir la réalité comme étant un système.

Selon un pôle d'observation structurel, un système est défini par :

- des *éléments constitutifs* dont on peut évaluer le nombre et la nature (même si ce n'est qu'approximativement). Ces éléments sont plus ou moins homogènes (par exemple, pour une automobile : groupe motopropulseur, châssis, habitacle, liaison au sol, carrosserie) ou hétérogènes (par exemple, pour une entreprise : capitaux, bâtiments, personnel, etc.),
- une *limite (ou frontière) qui sépare la totalité des éléments de son environnement* : cette limite est plus ou moins perméable. Elle constitue une interface avec le milieu extérieur. Par exemple, la membrane d'une cellule, la peau du corps et la carrosserie d'une voiture constituent les frontières de la cellule, du corps et de la voiture. Dans certains cas (un groupe social, par exemple), la limite d'un système peut être floue ou particulièrement mouvante,
- des *réseaux de relations* : les éléments sont inter-reliés les uns avec les autres. Plus les interrelations sont nombreuses, plus le degré d'organisation est élevé et plus grande est la complexité. Les relations peuvent être des relations de transport ou des relations de communication. Ces deux types de relations peuvent se réduire à un seul type. En effet, communiquer revient à transporter de l'information et transporter revient à communiquer (faire circuler) de la matière, de l'énergie ou de l'information,
- des *stocks (ou réservoirs)* où sont entreposés la matière, l'énergie ou l'information constituant les ressources du système qui doivent être transmises ou réceptionnées.

Selon un pôle d'observation fonctionnel, un système est défini par :

- des *flux* de matière, d'énergie ou d'informations, qui empruntent les réseaux de relations et transitent par les stocks. Ils fonctionnent par entrées/sorties (ou inputs/outputs) avec l'environnement,
- des *centres de décision* qui organisent les réseaux de relations, c'est-à-dire qui coordonnent les flux et qui gèrent les stocks,
- des *boucles de rétroaction* qui servent à informer, à l'entrée des flux, sur leur sortie, de façon à permettre aux centres de décision de connaître l'état général du système,
- des *ajustements* réalisés par les centres de décisions en fonction des boucles de rétroaction et des délais de réponse (correspondant au temps que mettent les informations « montantes » pour être traitées et au temps supplémentaire que mettent les informations « descendantes » pour se transformer en actions).

En résumé, un **système*** est « une combinaison d'éléments en interaction organisés pour atteindre un ou plusieurs objectifs définis » [ISO, 08]. Une deuxième définition mnémotechnique peut être proposée [Le Moigne, 90] :

Un système est :

- Quelque chose (combinaison d'éléments)
- Qui *dans* quelque chose (environnement)
- *Pour* quelque chose (objectif)
- *Fait* quelque chose (activité = fonctionnement)

- Par quelque chose (organisation)
- Qui se *transforme* dans le temps (évolution)

Il est possible de distinguer plusieurs types de systèmes [AFIS, 12] :

- le système technologique : système dont TOUS les constituants sont mécaniques, électriques, magnétiques, électroniques, logiciels, etc.,
- le système sociologique : système dont TOUS les constituants sont des humains,
- le système socio-technique : combinaison d'un système technologique et d'un système sociologique,
- le système informatique : système technologique dont TOUS les constituants sont soit des matériels informatiques (hardware) soit des logiciels (software).

Le paradigme systémique (cf. figure 2.6) est la conjonction de deux paradigmes [Le Moigne, 90] : le paradigme cybernétique et le paradigme structuraliste.

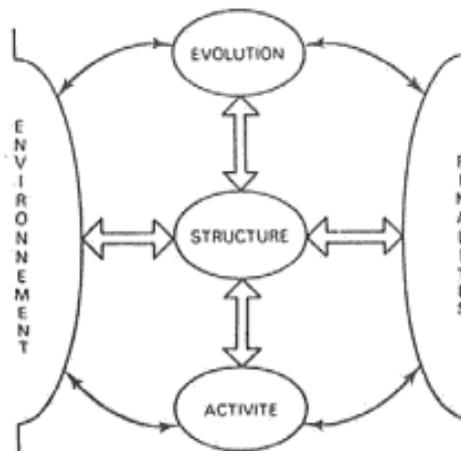


Figure 2.6 : Représentation d'un système [Le Moigne, 90]

Le **paradigme cybernétique*** perçoit le système comme une « boîte noire » dotée d'une finalité. La notion de « boîte » définit qu'il est possible de distinguer le système de son environnement. Ceci induit la définition d'une frontière entre le système et son environnement. Cette frontière détermine les interfaces, c'est-à-dire les « entrées » et les « sorties » du système. Le concept de « boîte noire » conduit à identifier la transformation des entrées en sorties sans se préoccuper de la structure interne qui va réaliser cette transformation.

Le **paradigme structuraliste*** perçoit le système comme une « boîte blanche ». Il s'agit, dans ce cas, de considérer le système comme une structure pourvue de fonctions qui la dotent d'un comportement. Ce paradigme définit donc trois « pôles d'observation » interdépendants :

- le *pôle ontologique* décrit la structure du système (ce qu'il est),
- le *pôle fonctionnel* décrit les fonctions que le système peut réaliser (ce qu'il fait),
- le *pôle génétique* décrit le comportement qui conduit le système à évoluer (ce qu'il devient).

En considérant que la structure du système peut être perçue comme un ensemble d'objets en interaction, le paradigme structuraliste peut être complété par le **paradigme objet***. Ce paradigme, issu du domaine informatique, perçoit les objets comme des structures dotées de propriétés qui lui confèrent un état et un comportement. Il est à noter que ce paradigme complète de façon pertinente le paradigme structuraliste. En effet, son application conduit à percevoir la réalité comme un ensemble d'objets en interaction. Elle produit ainsi des modèles qui peuvent être exprimés grâce à des langages informatiques. En d'autres termes, il s'agit d'identifier les « objets » du monde réel et leurs relations pour faciliter leur représentation numérique. Ce paradigme identifie trois espaces d'étude que nous utilisons dans la suite de ce mémoire : l'espace du besoin, l'espace du problème et l'espace de la solution [Booch et al., 00]. L'espace du besoin définit les modèles des exigences attendues, l'espace du problème s'appuie sur les modèles de gestion définis pour satisfaire les exigences tandis que l'espace de la solution identifie les modèles qui peuvent être implémentés sur un ordinateur. Ces

espaces correspondent aux modèles des trois pôles d'observation préconisés par l'architecture dirigée par les modèles (MDA – Model Driven Architecture) proposée par l'OMG (Object Management Group) [Blanc, 05] :

- le **modèle d'exigences** (CIM – Computer Independent Model) décrit les exigences du système,
- le **modèle métier** (PIM – Platform Independent Model) décrit le système indépendamment de la plate-forme technique utilisée pour sa réalisation,
- le **modèle plate-forme** (PSM – Platform Specific Model) décrit la solution implantée sur la plate-forme technique.

3.2 Le paradigme d'ambivalence

Le chapitre 1 a montré que l'entreprise peut être perçue à travers un pôle d'observation organisationnel et un pôle d'observation informationnel. Selon les besoins, des pôles sociologique, productif, financier, etc. peuvent être également utilisés. Chaque pôle d'observation fournit des modèles de l'entreprise adaptés à des thèmes d'étude. Il focalise sur les caractéristiques pertinentes pour l'étude à réaliser. Le **paradigme d'ambivalence*** considère que les perceptions correspondant à chaque pôle d'observation ne doivent pas être juxtaposées mais considérées de manière complémentaire et corrélée [Chelli, 03].

Comme cela a été défini dans le chapitre 1, la construction du système PLM doit s'appuyer sur le pôle organisationnel et sur le pôle informationnel (cf. chapitre 1, § 2.3.2). Rappelons que le pôle organisationnel fournit aux responsables métiers les modèles de l'entreprise leur permettant d'intervenir sur les processus et sur les produits qui sont la raison d'être de l'entreprise. Le pôle informationnel procure aux techniciens de l'information les modèles nécessaires pour « instrumenter » la gestion des processus et des produits. Cette perception de la réalité opérationnelle à travers deux pôles peut être rapprochée de la structuration proposée par l'ingénierie dirigée par les modèles :

- La perception du pôle organisationnel fournit une description du système PLM de l'entreprise indépendante de la plate-forme logicielle utilisée,
- La perception du pôle informationnel fournit une description de l'implémentation du système PLM sur la plate-forme logicielle de l'entreprise.

Le paradigme d'ambivalence considère l'entreprise comme une réalité ambivalente. Cette réalité doit être observée en identifiant les complémentarités et les correspondances entre les perceptions organisationnelle et informationnelle afin de développer une synergie entre les responsables métiers et les techniciens de l'information. Ce paradigme propose une rupture par rapport à la position actuelle qui considère que les techniciens de l'information doivent aligner leurs modèles sur ceux proposés par les responsables métiers. La figure 2.7 présente ainsi le raisonnement synergique promu par ce paradigme.

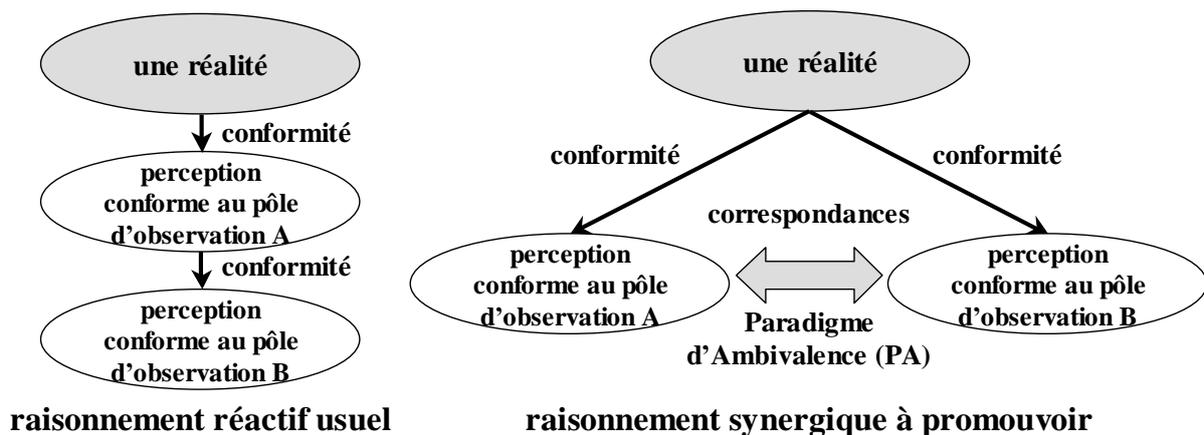


Figure 2.7 : Différences entre les raisonnements réactif et synergique (adapté de [Chelli, 03])

La définition des correspondances entre les deux pôles d'observation s'appuie sur trois principes [Chelli, 03]. Le premier principe considère que : « *Ce qui s'applique à la perception d'un pôle d'observation s'applique à la perception de l'autre pôle* ». Chaque pôle d'observation s'appuie sur des concepts qui lui sont propres. Le premier principe stipule donc qu'il doit exister une correspondance entre les concepts des différents pôles d'observation. Le deuxième principe précise que : « *La perception selon un pôle d'observation doit se faire à la fois à un niveau macroscopique (ou global) et à un niveau microscopique (ou détaillé)* ». Enfin le troisième principe stipule que : « *Au sein d'un pôle d'observation, tout concept de perception microscopique (ou détaillé) a pour vocation de décomposer le contenu d'un unique concept de perception macroscopique (ou globale)* ». Dans les travaux de ce mémoire, nous nous intéressons plus particulièrement à l'application du premier principe.

3.3 Modélisation du système PLM

Les paradigmes proposés dans les paragraphes précédents permettent de définir de façon précise les modèles suivants :

- **modèle d'exigences PLM** : ce modèle représente l'ensemble des exigences auquel doit satisfaire le système,
- **modèle métier** : ce modèle représente l'ensemble des éléments liés à la gestion de l'entreprise,
- **modèle métier PLM** : ce modèle représente l'ensemble des éléments liés à la gestion du cycle de vie du produit,
- **modèle plate-forme** : ce modèle représente l'ensemble des éléments liés à la gestion de l'entreprise qui sont pris en compte par des applications logicielles,
- **modèle plate-forme PLM¹⁹** : ce modèle représente l'ensemble des éléments liés à la gestion du cycle de vie du produit qui sont pris en compte par des applications logicielles,
- **modèle personnalisé du logiciel PLM** : ce modèle représente l'ensemble des éléments liés à la gestion du cycle de vie du produit qui sont pris en compte par un logiciel PLM.

¹⁹ Rappelons qu'au sein d'une entreprise il peut exister plusieurs applications qui ont en charge certaines étapes du cycle de vie du produit : ERP, CRM, CAO, FAO, PLM, etc.

Le tableau suivant présente les modèles utilisés dans les travaux de ce mémoire :

Système PLM	Définition
Modèle d'exigences PLM	Modèle représentant l'ensemble des exigences auquel doit satisfaire le système
Modèle métier PLM	Modèle représentant le pôle d'observation organisationnel lié à la gestion du cycle de vie du produit
Modèle plate-forme PLM	Modèle représentant l'ensemble des éléments liés à la gestion du cycle de vie du produit qui sont pris en compte par des applications logicielles
Modèle personnalisé du logiciel PLM	Modèle représentant le pôle d'observation informationnel lié à l'utilisation d'un logiciel PLM

Tableau 2.4 : Définitions du modèle métier PLM et du modèle personnalisé du logiciel PLM

4 Les cadres de modélisation

Les paragraphes précédents ont introduit de nombreux modèles. D'une part, le système PLM a pour objectif de gérer les représentations liées à la gestion du cycle de vie du produit. D'autre part, la construction du système PLM s'appuie sur des modèles représentant des perceptions de l'entreprise à travers différents pôles d'observation. Les questions qui se posent maintenant sont : Comment choisir le bon modèle au bon moment ? Comment assurer la cohérence entre les différents modèles ?

Les réponses à ces questions nécessitent de positionner les modèles les uns par rapport aux autres et de définir des relations entre les modèles. Il existe généralement deux types de relations entre les modèles²⁰ :

- La *relation de décomposition* (ou raffinement horizontal) définit la relation qui existe entre deux modèles construits à partir du même langage mais qui décrivent la réalité à des niveaux de détail différents. Par exemple, le langage SADT [IGL, 93] permet de construire un modèle (actigramme) décrivant les flux échangés entre les activités d'un système et chaque activité peut elle-même être décrite par un modèle (actigramme) en utilisant les mêmes symboles.
- La *relation de réécriture* (ou raffinement vertical) définit la relation qui existe entre deux modèles qui sont construits à partir de langages différents. La définition de cette relation s'appuie sur les métamodèles de chacun des langages.

L'objectif de ce paragraphe est d'identifier les différentes solutions existant pour répondre aux deux questions posées. Les solutions étudiées pour positionner les modèles les uns par rapport aux autres sont issues de méthodes développées pour modéliser le système d'information ou l'entreprise. Dans un premier temps, ce paragraphe décrit les cadres de modélisation qui proposent un positionnement entre les différents modèles utilisés pour percevoir une réalité (cf. § 4.1). Il expose ensuite les différentes dynamiques de passage pouvant exister entre les modèles (cf. 4.2). Enfin, ce paragraphe présente une ébauche du cadre de modélisation qui est à la base de la proposition du chapitre 4 de ce mémoire (cf. 4.3).

4.1 Les cadres de modélisation

Un **cadre de modélisation*** est une structure qui positionne les différents modèles proposés par une méthode suivant des axes d'étude définis, appelés aussi *dimensions*. Ses objectifs sont de guider le modélisateur dans le choix des modèles à utiliser et de lui permettre de cheminer afin de comprendre le tout [Pingaud, 05]. Un axe d'étude est déterminé par le paradigme sur lequel le cadre s'appuie pour percevoir la réalité. Par exemple, le paradigme structuraliste définit un axe identifiant les différents aspects pour observer un système : structure, fonction et comportement.

Chaque axe d'étude propose différents modèles pour permettre au modélisateur de se focaliser sur certains aspects du problème tout en gardant une perception globale. Les cadres de modélisation n'ont

²⁰ Ces relations sont souvent désignées *raffinement* dans le domaine du génie logiciel. [Bolusset et al., 02]

pas vocation à imposer des langages pour construire les différents modèles. Cependant, la plupart du temps, les méthodes proposent des langages pour chaque modèle identifié dans le cadre.

L'étude d'une réalité telle que l'entreprise nécessite d'introduire des niveaux d'abstraction. Par exemple, l'étude de l'entreprise implique de s'intéresser à son fonctionnement, à son organisation, à l'organisation logique et physique des moyens informatiques utilisés.

Un des premiers cadres de modélisation existants a été proposé par la méthode Merise [Nanci et al., 01]. Le schéma suivant illustre les axes d'étude de cette méthode :

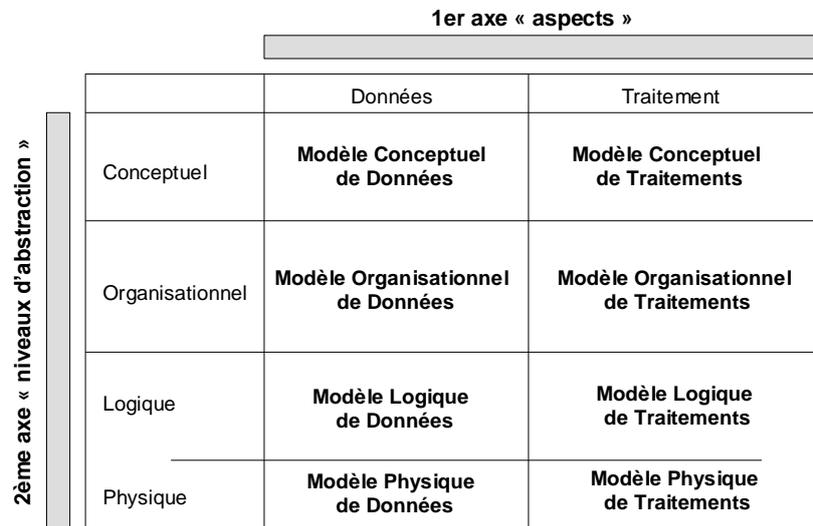


Figure 2.8 : Les deux axes du cadre de Merise (adapté de [Gabay, 98])

Comme le montre la figure précédente, ce cadre de modélisation est constitué de deux axes qui positionnent les modèles en fonction de deux points de vue et de quatre niveaux d'abstraction. Ainsi, le cadre Merise considère que le modélisateur doit s'appuyer sur huit modèles pour étudier le système d'information d'une organisation. Les modèles décrivent d'une part les données manipulées par le système et, d'autre part, les traitements associés à ces données. Les niveaux d'abstraction définissent quant à eux des points d'observation adaptés à une étude spécifique. Le niveau conceptuel (ou intentionnel selon les auteurs) permet de répondre aux questions : « Que fait le système ? » « Quelle est la finalité du système ? » « Quelles sont les informations manipulées par le système ? ». Le niveau organisationnel fournit des modèles pour répondre aux questions suivantes : « Quels sont les accès aux données des différents acteurs ? » « Qui réalise un traitement ? ». Les modèles des deux autres niveaux d'abstraction fournissent des réponses aux différentes questions traitant des aspects informatiques du système d'information. Il est à noter qu'un parallèle peut être fait avec les types de modèles introduits par MDA. Les niveaux conceptuel et organisationnel définissent le Système d'Information Organisationnel (SIO) qui correspond au modèle indépendant de la plate-forme (PIM – Platform Independent Model). Les niveaux logique et physique définissent quant à eux le système d'information informatisé (SII) qui correspond au modèle dépendant de la plate-forme (PSM – Platform Specific Model).

Suite à la complexité grandissante des systèmes d'information, plusieurs autres cadres ont été définis dans ce domaine. Le cadre ARIS (Architecture for Integrated Information Systems) est centré sur la description des processus métiers. Ce cadre a pour objectif de supporter la conception des systèmes d'information intégrés [Bernus et al., 05]. Le cadre Zachman propose deux axes d'étude. Le premier axe est constitué des six interrogations de base : Quoi, Comment, Où, Qui, Quand et Pourquoi. Le deuxième axe est composé de six catégories de parties prenantes : Visionnaire, Propriétaire, Concepteur, Réalisateur, Sous-traitant et Exécutant [Zachman, 08].

Dans le domaine de la modélisation d'entreprise, un des premiers cadres de modélisation proposé a été CIMOSA (Computer Integrated Manufacturing Open System Architecture) [Kosanke, 99]. Ce cadre

supporte la modélisation des systèmes intégrés de production. Il est composé de trois axes d'étude d'où le nom « cube CIMOSA » (cf. figure 2.9).

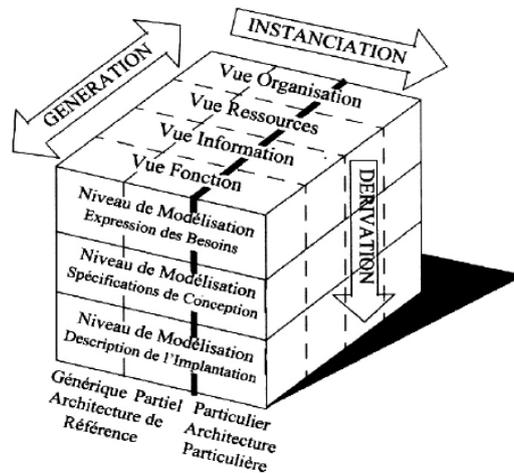


Figure 2.9 : Le cadre de modélisation CIMOSA [Kosanke, 99]

Dans ce cadre, l'axe d'instanciation identifie un niveau générique, un niveau partiel et un niveau particulier. Les modèles des deux premiers niveaux constituent une architecture de référence qui identifie les concepts communs à toutes les entreprises (niveau générique) et les concepts propres à un type d'entreprise (niveau partiel). Le niveau particulier décrit quant à lui les concepts d'une entreprise donnée. L'axe dérivation décrit le cycle de vie du système à travers trois étapes : expression des besoins, spécification et conception du système, description de l'implantation. Tout comme pour le cadre de modélisation Merise, un parallèle peut être fait entre cet axe et les modèles proposés par MDA : le niveau décrivant l'expression des besoins correspond au modèle d'exigences (CIM – Computer Independant Model), le niveau décrivant les spécifications et la conception du système correspond au modèle indépendant de la plate-forme (PIM – Platform Independant Model) et le niveau décrivant l'implantation correspond au modèle dépendant de la plate-forme (PSM – Platform Specific Model). L'axe de génération identifie quatre pôles d'observation du système intégré de production : fonction, information, ressources et organisation.

Depuis la création du cube CIMOSA, d'autres cadres de modélisation ont été proposés. Le cadre GRAI propose une grille fonctionnelle pour modéliser le système de décision de l'entreprise à travers deux axes d'étude [Doumeings, 85]. Le premier axe identifie les différents services de l'entreprise tandis que le second axe identifie différents niveaux de décisions. Le cadre PERA (Purdue Enterprise Reference Architecture) propose de modéliser les systèmes industriels à travers deux axes d'étude [Williams, 94]. Le premier axe décrit un cycle de vie basé sur sept étapes tandis que le second axe définit trois pôles d'observation : équipements, humains, contrôle/commande. Le cadre de modélisation GERAM (Generalized Enterprise Reference Architecture and Methodology) est une composition du cadre CIMOSA, du cadre PERA et du cadre GRAI [IFIP-IFAC Task Force, 99] [Chen et al., 97] .

Outre la définition des modèles à utiliser, le cadre de modélisation identifie le positionnement relatif des modèles, les zones de recouvrement et la dynamique de passage entre les modèles [Vallespir et al., 03]. Dans le paragraphe suivant, un état de l'art présente les différents mécanismes permettant d'assurer le passage entre les modèles.

4.2 La dynamique de passage entre les modèles

Chaque acteur de l'entreprise construit des modèles à partir d'un langage qui lui est propre. Lorsque deux acteurs sont amenés à collaborer au sein d'un même projet, il faut alors faire en sorte que ces derniers soient capables de communiquer. Lorsque deux langages différents sont utilisés pour construire des modèles, il faut définir un moyen pour que les deux acteurs comprennent les modèles

construits. Cette situation peut être décrite en s'appuyant sur une analogie de deux personnes ne parlant pas la même langue. Imaginons deux personnes parlant respectivement le russe et l'espagnol. Il existe deux possibilités pour faire en sorte que ces deux personnes se comprennent. La première consiste à utiliser une langue commune partagée par ces deux personnes (l'anglais, par exemple). La seconde possibilité consiste à faire appel à un interprète qui assure la traduction entre les deux langues. Ce paragraphe s'appuie sur cette analogie pour présenter les différents mécanismes de passage entre les modèles.

4.2.1 Utilisation d'un langage commun

Afin que deux acteurs puissent collaborer, la première dynamique de passage consiste à s'appuyer sur un langage commun. Il existe différentes possibilités pour définir ce langage commun :

- utilisation du même langage pour construire tous les modèles du cadre de modélisation,
- définition d'un langage unifié pour partager les concepts communs aux langages utilisés dans le cadre de modélisation,
- définition d'un langage englobant les différents langages utilisés dans le cadre de modélisation.

Utilisation d'un unique langage

L'utilisation du même langage de modélisation pour construire les différents modèles définis au sein d'un cadre de modélisation garantit une bonne compréhension de l'ensemble des modèles par tous les acteurs. La méthode GRAI préconise de décrire le fonctionnement de chaque centre de décision²¹ de la grille fonctionnelle par un réseau GRAI [Doumeingts, 85]. Lorsqu'une méthode ne propose pas de langage spécifique, il est possible de s'appuyer sur un langage partagé par l'ensemble d'une communauté. La notation BPMN (Business Process Model and Notation) proposé par l'OMG²² est couramment utilisée pour modéliser les processus. Suite à un accord entre les différents acteurs, cette notation peut être utilisée pour représenter l'ensemble des processus de l'entreprise.

Définition d'un langage unifié

Dans certains cas, il n'est pas possible d'utiliser un langage commun car les modèles construits introduisent des concepts qui sont spécifiques à un domaine. Dans ce cas, une solution peut être apportée en identifiant les concepts communs aux différents langages. Une fois les concepts identifiés il faut les unifier pour qu'ils soient partagés par les différents langages. Cette deuxième solution se limite donc à un partage de concepts communs. Le langage UML [Rumbaugh et al., 04] est une illustration de cette possibilité. Initialement, plusieurs langages existaient pour modéliser les différents aspects proposés par le paradigme objet. Afin de créer une homogénéité dans la communauté réalisant des travaux selon ce paradigme, un consensus a défini un langage unifié. Ce langage est composé de treize types de diagrammes qui sont utilisés pour construire les modèles selon différents pôles d'observation. Ces diagrammes se répartissent en deux catégories : les diagrammes structurels (ou statiques) et les diagrammes comportementaux (ou dynamiques). Certains concepts sont partagés entre plusieurs diagrammes (l'acteur, par exemple). Ces concepts partagés ont été définis de manière à pouvoir être utilisés de façon cohérente dans chacun des diagrammes.

Définition d'un « sur-langage »

Une dernière possibilité consiste à construire un langage capable d'englober tous les langages utilisés dans un cadre de modélisation. Dans ce cas, ce « sur-langage » est défini comme l'union de tous les langages préconisés par le cadre de modélisation. Chaque langage de modélisation est alors considéré comme une projection du « sur-langage ». Le projet UEML (Unified Enterprise Modelling Language) avait pour objectif initial de proposer un métamodèle englobant tous les métamodèles des langages couramment utilisés dans le domaine de la modélisation d'entreprise [Vallespir et al., 03].

²¹ Un centre de décision représente les différentes décisions prises au sein d'une fonction à un niveau donné.

²² BPMN 2.0, Business Process Model and Notation, <http://www.omg.org/spec/BPMN/2.0>

4.2.2 Transformation de modèles

La deuxième dynamique de passage identifiée par l'analogie présentée dans l'introduction de ce paragraphe consiste à définir des mécanismes de transformation entre les différents modèles préconisés par le cadre de modélisation. Ces mécanismes, définis dans le cadre de travaux issus de l'IDM, nécessitent de formaliser les règles de transformation entre les langages.

L'Ingénierie Dirigée par les Modèles (IDM), ou Model Driven Engineering (MDE) est née du besoin de pouvoir d'une part réutiliser des modèles ou des transformations entre modèles et, d'autre part, de transformer de manière automatique les modèles métiers (PIM – Platform Independent Model) en modèles exécutables sur une plate-forme (PSM – Platform Specific Model) [Kadima, 05] [Favre et al., 06]. La définition de ces mécanismes de transformation offre plusieurs avantages. Le principal intérêt est de rendre le modèle productif. En effet, la définition des transformations permet de « produire » un modèle à partir d'un autre modèle ce qui fait que les modèles ne sont plus de simples « outils de communication ». Ces transformations permettent donc de transformer le modèle construit à partir d'un langage donné en un modèle construit dans un autre langage de modélisation (M2M – Model 2 Model). D'autre part, elles permettent également un modèle décrit dans un langage donné en une représentation décrite dans un langage de programmation donné (M2T – Model 2 Texte). La figure 2.10 illustre ainsi ces deux types de transformations.

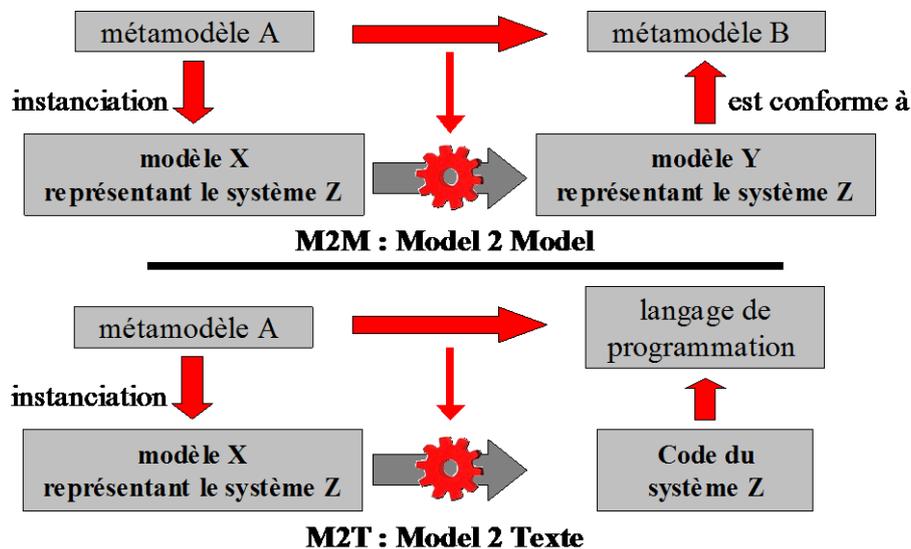


Figure 2.10: Transformations de modèles M2M et M2T

Dans le premier exemple de la figure 2.10, le modélisateur définit des règles de transformation entre les métamodèles A et B. A partir du métamodèle A, un modèle X représentant un système Z est construit. L'objectif consiste alors à construire une nouvelle représentation Y du système Z à partir des symboles et des structures du métamodèle B. Les règles de transformation doivent donc décrire les correspondances entre les symboles (vocabulaire) et les structures (syntaxes) des langages formalisés par les métamodèles A et B. Le second exemple de la figure 2.10 consiste à traduire un modèle dans un langage informatique. Le modélisateur crée un modèle à partir d'un métamodèle donné et le code de l'application est ensuite généré automatiquement. L'application de ces mécanismes offre de nombreux intérêts comme l'élaboration d'exigences dont la cohérence est contrôlée automatiquement, l'application automatique de patrons de conception, la génération de code et la génération de tests, etc. Les transformations mises en œuvre peuvent également formaliser dans les éléments des modèles générés automatiquement des traces identifiant les éléments des modèles à partir desquels ils ont été créés [Jézéquel et al., 06].

Dans certains cas, il est possible d'employer un langage « pivot » lors des transformations. Certains langages, comme le langage XML (eXtensible Markup Language), peuvent être utilisés pour transférer une représentation entre deux acteurs. L'utilisation d'un tel langage nécessite donc deux mécanismes de transformation. Dans le cas de la gestion du cycle de vie du produit, la norme STEP [ISO, 10] peut

servir de langage pivot pour transférer des représentations du produit entre un système PLM et d'autres systèmes.

Concernant les modèles du système PLM, l'objectif de nos travaux est de proposer des mécanismes de transformation pour, d'une part, construire le modèle métier PLM à partir du modèle d'exigence et, d'autre part, construire le modèle personnalisé du logiciel PLM à partir du modèle métier PLM.

4.3 Le cadre de modélisation de notre proposition

L'objectif de ce paragraphe est de construire un cadre de modélisation permettant de supporter la mise en oeuvre du système PLM. Le cadre proposé doit tenir compte des spécificités des systèmes d'information [Nanci et al., 01] [Blanc, 05]. Il doit également permettre d'appréhender l'entreprise dans sa globalité. De ce fait, il s'inspire des cadres de modélisation d'entreprise existants [Kosanke et al., 99] [IFIP-IFAC Task Force, 99] [Chen et al., 97] [Noran, 03]. Ainsi, dans ce paragraphe, la construction du cadre s'appuie sur les différents éléments présentés tout au long de ce chapitre.

Le système PLM doit gérer les représentations du produit. Le cadre de modélisation doit donc englober les modèles associés aux différentes étapes du cycle de vie du produit. Ces modèles sont utilisés pour construire le niveau de visibilité des acteurs intervenant dans la gestion du cycle de vie produit. Le premier axe d'étude de notre cadre de modélisation est donc basé sur le cycle de vie du produit. Il a pour objectif de positionner les représentations du produit, que le produit soit théorique ou physique, en fonction des étapes de son cycle de vie. Le schéma 2.11 présente cet axe d'étude :

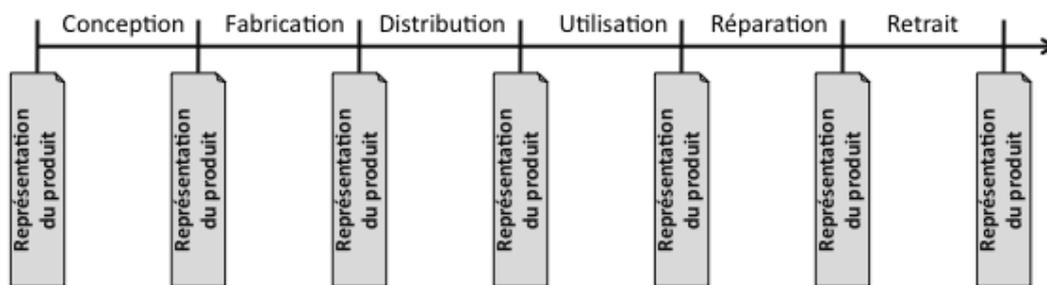


Figure 2.11: Axe d'étude basé sur le cycle de vie du produit

Le cadre de modélisation proposé devant supporter la construction du système PLM, le deuxième axe d'étude identifié concerne le cycle de vie du système PLM. Cet axe a pour objectif de positionner les différents modèles nécessaires à la réalisation du système PLM. Les trois étapes identifiées pour ce cycle sont identiques à celles proposées par le cadre CIMOSA. Le système PLM est un système d'information, il existe donc au sein de l'entreprise. Il est donc essentiel de définir les besoins en tenant compte du système existant. Le problème est formalisé dans le **modèle d'exigences PLM**, partie essentielle du cahier des charges du système. L'étape de spécification et de conception définit une solution (indépendante de toute plate-forme technique) au problème défini par l'étape précédente. Cette solution abstraite est formalisée dans le **modèle métier PLM**. Enfin, l'étape réalisant l'implémentation de la solution définit l'implémentation de la solution à partir de la solution définie par l'étape précédente. Cette solution concrète est formalisée dans le **modèle plate-forme PLM**.

Les différents espaces (problème et solution) pris en compte par ces modèles sont définis par le schéma suivant :

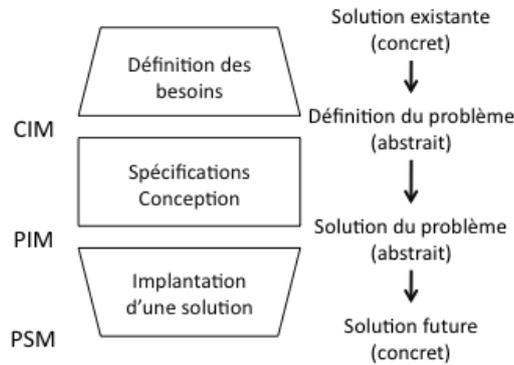


Figure 2.12: Le cycle de vie du système PLM

Le schéma 2.13 complète le cadre de modélisation en ajoutant cet axe d'étude.

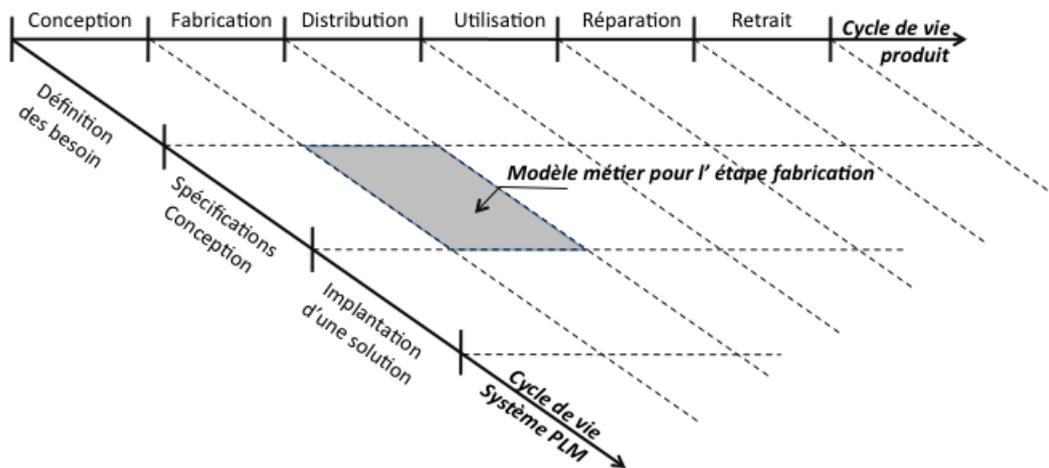


Figure 2.13 : Cadre combinant les axes « cycle de vie produit » et « cycle de vie système PLM »

Enfin, ce cadre doit fournir au modélisateur les mécanismes de transformation entre ces modèles. Il doit donc intégrer les différents niveaux de modélisation définis par l'ingénierie dirigée par les modèles.

Le schéma suivant présente le cadre de modélisation à la base de notre proposition.

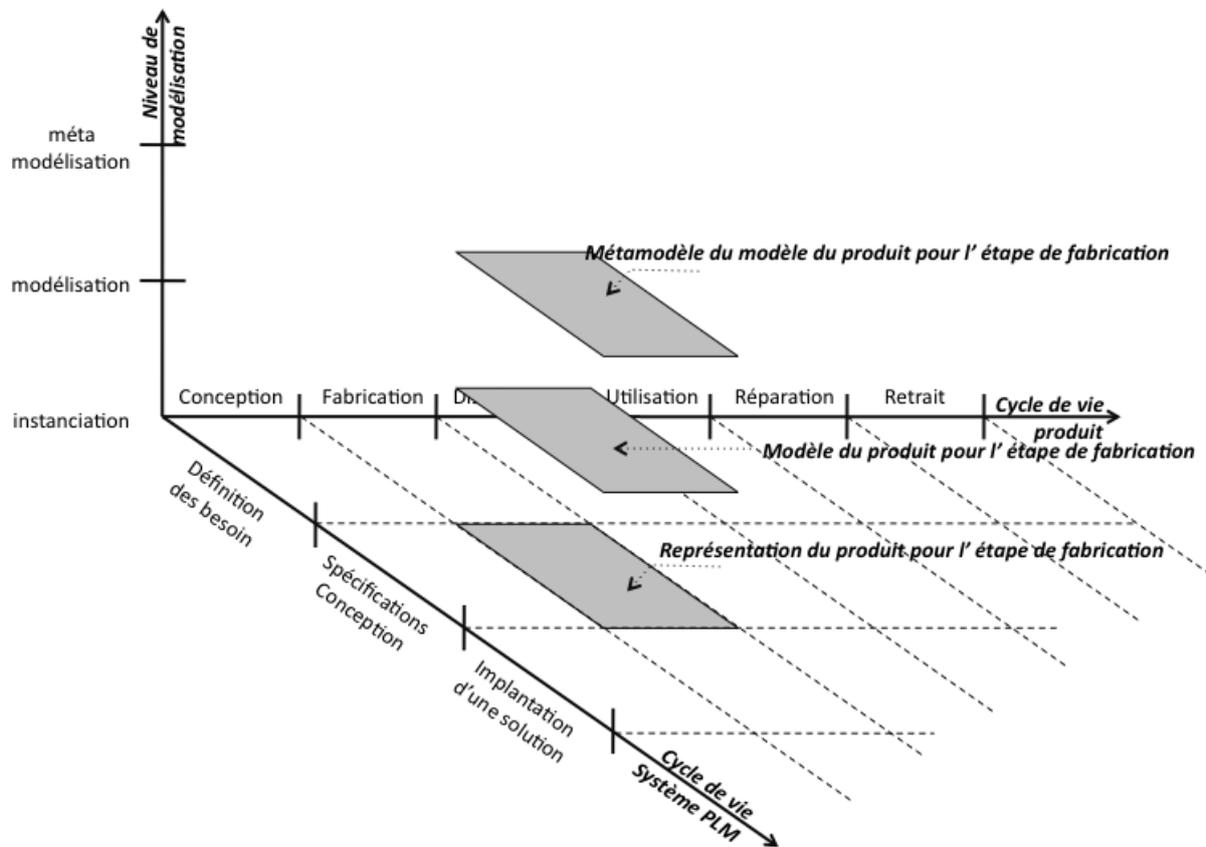


Figure 2.14 : Cadre de modélisation de notre proposition

5 Conclusion

Ce chapitre a présenté les concepts de modélisation utilisés dans les travaux de ce mémoire. Il a également défini les principales étapes du processus de modélisation en identifiant les paradigmes employés pour nos travaux. Après avoir déterminé les caractéristiques d'un cadre de modélisation, ce chapitre s'est terminé par la construction d'un cadre supportant la mise en œuvre du système PLM. Dans le chapitre 4, ce cadre, qui constitue la proposition des travaux de ce mémoire, est défini de manière plus détaillée et il est positionné par rapport aux travaux PLM existants.

Suite à l'énoncé des bases de notre proposition, le chapitre suivant a pour objectif de proposer une méthode pour élaborer le cahier des charges du système PLM en tenant compte des spécificités de ces systèmes.

Chapitre 3

Méthode d'élaboration du cahier des charges du système PLM

Sommaire

- 1 Introduction
 - 2 Etat de l'art : de l'identification des besoins à l'élaboration du CDC
 - 3 Les spécificités du système PLM et les difficultés rencontrées pour élaborer son CDC
 - 4 La définition de niveaux de maturité
 - 5 Vers une méthode d'élaboration du CDC du système PLM
 - 6 Conclusion
-

1 Introduction

La conclusion du chapitre 1 est qu'il est nécessaire de déterminer une méthode pour construire un système PLM en adéquation avec les deux buts de l'entreprise. Pour être en accord avec le cycle de vie du système PLM, le premier sujet traité par cette méthode est l'élaboration du cahier des charges du système PLM (cf. chapitre 2 § 4.3). Ainsi, ce chapitre a pour objectif de définir un processus d'élaboration du cahier des charges à partir des besoins et des contraintes identifiés. De nombreux travaux ont montré que cette élaboration est une étape essentielle pour construire un système d'information²³ [Rolland et al., 01]. En effet, une identification incomplète ou erronée des besoins aboutit généralement au développement ou au déploiement de logiciels qui ne répondent pas aux attentes des utilisateurs. Des études mettent en évidence la fréquence élevée de cette carence dans les projets informatiques et l'importance des coûts de correction induits [Bell et al., 76] [Standish, 95]. Face à ces constats, différents travaux ont été menés pour guider l'identification et la formalisation des buts (i.e. le « pourquoi » du système) et des besoins et contraintes (i.e. le « quoi » du système) pour les systèmes d'information [Rolland et al., 01] [Nuseibeh et al., 00]. Le résultat de cette ingénierie d'identification des besoins et des contraintes est le Cahier Des Charges (ou CDC) qui recense les exigences fonctionnelles et non fonctionnelles que le système d'information doit satisfaire.

Dans un premier temps, ce chapitre présente un état de l'art des concepts relatifs à l'identification et à la formalisation des buts, des besoins et des contraintes dans le domaine des systèmes d'information. Suite à cet état de l'art, les modèles et le processus retenus dans ces travaux sont choisis (§ 2). Un état des lieux définit alors les spécificités du système PLM et les difficultés rencontrées lors de l'élaboration de son CDC (§ 3). Cet état des lieux met en lumière la nécessité d'intégrer l'évaluation de la maturité PLM de l'entreprise lors du processus d'élaboration du CDC (§ 4). Enfin, une méthode adaptée est proposée (§ 5).

2 Etat de l'art : de l'identification des besoins à l'élaboration du CDC

L'objectif de ce paragraphe est de sélectionner parmi les travaux existants les modèles et le processus adaptés à l'élaboration du CDC du système PLM.

Tout d'abord, ce paragraphe décrit les différents modèles nécessaires à la formalisation des buts, des besoins et des contraintes d'un système (§ 2.1). Il présente ensuite les différents langages (§ 2.2) et un processus (§ 2.3) utilisés pour construire ces modèles. Ce paragraphe se conclut alors par la mise en lumière de difficultés rencontrées pour l'élaboration du CDC.

²³ Rappelons que le système PLM est défini comme un système d'information supportant la création, la circulation, l'utilisation et l'évolution du patrimoine informationnel de représentation du produit.

2.1 Quels modèles pour définir les buts, les besoins et les contraintes ?

La première étape de la construction du système²⁴ consiste à identifier et à formaliser les exigences que le système doit satisfaire. Ces exigences sont rassemblées au sein d'un référentiel, désigné *référentiel d'exigences*. Ce référentiel est défini à partir des besoins des utilisateurs du système et des contraintes à respecter. En d'autres termes, ce référentiel formalise le problème à résoudre. Il établit le contrat entre la personne physique ou morale pour le compte de laquelle le système est réalisé (maître d'ouvrage) et la personne physique ou morale qui le conçoit et en contrôle la réalisation (maître d'œuvre) [AFIS, 12].

Depuis les années 80, deux catégories de travaux traitent de la définition du référentiel d'exigences. Les travaux de la première catégorie concernent l'analyse fonctionnelle (ou analyse de la valeur) des produits manufacturés [AFNOR, 91]. Ceux de la deuxième catégorie concernent l'ingénierie de définition des besoins dans les domaines du génie logiciel [Sommerville, 92] et du système d'information [Rolland et al., 01] [Nuseibeh et al., 00]. Plus récemment, les travaux relatifs à l'ingénierie système proposent une approche globale qui s'applique à tout système, qu'il soit tangible (produits manufacturés) ou intangible (logiciel et système d'information) [AFIS, 12].

Parmi les différentes définitions existantes [IEEE, 90] [Sommerville et al., 97] [Wieggers, 00] nous retenons celle proposée par l'AFIS pour définir ce qu'est une exigence [AFIS, 12]. Une *exigence** est un énoncé qui spécifie une aptitude, une caractéristique ou une limitation d'un système, d'un produit ou d'un processus pour contribuer, dans des conditions d'environnement données, à la satisfaction d'un but donné [AFIS, 12].

Cette définition met en lumière la notion de *but*. Il est courant de s'appuyer sur des questions pour différencier les concepts de « but » et « d'exigence ». Le but permet de répondre à la question « Pourquoi le système existe ? » tandis que l'exigence répond à la question « Que doit faire le système ? ». Plusieurs travaux considèrent qu'il est nécessaire de comprendre les buts pour pouvoir exprimer correctement les exigences. Ils proposent donc des méthodes d'ingénierie de définition des besoins dirigées par les buts [Dardenne et al., 93] [Anton, 96] [Mylopoulos et al., 99] [Rolland, 03]. Ces méthodes préconisent une démarche pour construire une hiérarchie des buts attribués au système. Une fois cette hiérarchie déterminée, les exigences sont identifiées à partir des buts du plus bas niveau de la hiérarchie [Rolland, 03].

Comme exemples de buts, nous pouvons citer « diminuer le nombre d'utilisations de la mauvaise version d'un document », « diminuer le temps de recherche de documents », « sécuriser les données », « augmenter la réutilisation de modèles géométriques existants », etc.

Comme exemples d'exigences, nous pouvons citer « le système permet à l'utilisateur de créer des versions de documents », « le système permet à l'utilisateur de changer le statut d'un document », etc.

Pour être convenablement formalisée, une exigence doit répondre à plusieurs *caractéristiques de qualité* [AFIS, 12] qui sont :

- L'*unicité* : l'exigence ne traite que d'un sujet,
- La *précision* : l'exigence est rigoureuse dans son expression,
- La *non-ambiguïté* : l'exigence ne permet qu'une interprétation possible,
- La *pure prescription de résultat* : l'exigence répond à la question « Quoi ? » et non à la question « Comment ? »,
- La *vérifiabilité* : l'exigence est vérifiable,
- La *faisabilité* : l'exigence doit pouvoir être satisfaite dans le contexte de l'état de l'art technologique à l'horizon envisagé,
- Le *réalisme* : l'exigence doit pouvoir être satisfaite dans le contexte des contraintes du projet.

²⁴ Ce paragraphe reste volontairement général sur la formalisation des buts, des besoins et des contraintes. Même si un certain nombre de points sont issus de travaux orientés sur les systèmes d'information, les principes généraux énoncés s'appliquent généralement à tout système. Ils devront donc être pris en compte, par le système PLM, lors de la définition du référentiel d'exigences des produits. Les exemples proposés pour illustrer nos propos sont toujours liés au système PLM.

La non-ambiguïté nécessite de décomposer l'exigence jusqu'à un niveau qui garantisse la bonne compréhension entre les différentes parties prenantes liées à la construction du système. Prenons l'exemple de l'exigence « le système permet à l'utilisateur de créer des versions de documents ». La mise en œuvre de cette exigence n'implique pas forcément les deux aptitudes suivantes : « le système demande à l'utilisateur de rédiger un commentaire pour chaque nouvelle version d'un document existant » et « le système stocke les anciennes versions d'un document ». Si ces deux aptitudes sont requises, il est nécessaire de les formaliser dans le référentiel d'exigences. En effet, les juger implicites serait une erreur. Cela pourrait avoir pour conséquence une non-satisfaction du besoin.

La **vérification des exigences** garantit que les exigences sont écrites selon les règles de l'art, c'est à dire qu'elles respectent les caractéristiques de qualité définies.

Dans le cadre de l'ingénierie de définition des besoins, [Wiegers, 03] préconise de séparer les exigences relatives au système, des exigences relatives au projet de construction du système (coûts, délais, etc.). Pour construire le référentiel d'exigences, il est nécessaire de définir les buts, les besoins et les contraintes. Pour faciliter cette définition, il est possible d'utiliser la classification suivante [Wiegers, 03]²⁵ :

Les buts (*exigences d'affaires*) sont formulés sous forme de phrase avec un verbe à l'infinitif, par exemple « augmenter de 10% la ré-utilisation de modèle 3D existants ». Le verbe sous-entendu est le verbe *vouloir*. Ils répondent à la question *pourquoi* veut-on mettre le système en place ?

Les besoins utilisateurs (*exigences d'utilisation*) sont décrits sous forme de cas d'utilisation ou de scénarios. Ils sont formulés sous forme de phrase avec un verbe à l'infinitif, par exemple « Gérer des versions de documents ». Le verbe sous-entendu est le verbe *avoir besoin de*. Ils répondent à la question *quoi*, c'est à dire ce que le système doit faire.

Les règles d'affaires correspondent à des besoins et à des contraintes métiers ou réglementaires. Le verbe sous-entendu est le verbe *devoir*. Par exemple, « un utilisateur du service production n'a accès à un plan que si ce dernier est en état valide ».

Les exigences fonctionnelles expriment les comportements attendus du système. Elles dérivent des trois catégories précédentes.

Cette classification définit également trois catégories d'**exigences non fonctionnelles** :

Les *exigences de qualité* expriment des propriétés attendues pour des aptitudes telles que la rapidité, l'efficacité, la robustesse, l'ergonomie, etc.

Les *exigences d'interface* définissent les communications entre le système à l'étude et son environnement.

Les *exigences techniques* déterminent des contraintes relatives à l'utilisation d'un langage particulier, de technologies spécifiques. Par exemple, les exigences sur les formats de données identifient qu'un code postal est composé de cinq chiffres.

Les buts, les besoins utilisateurs et les exigences fonctionnelles concernent la perception organisationnelle de l'entreprise tandis que les exigences non-fonctionnelles sont plutôt liées à la perception informationnelle (exigences d'interface, exigences de qualité, exigences techniques, etc.). Toutes ces catégories expriment la perception « boîte noire » du système. Elles définissent en effet sa finalité et elles identifient les interfaces du système avec son environnement (cf. chapitre 2, § 3.1).

Il est à noter que la définition des exigences non fonctionnelles est aussi importante que la définition des exigences fonctionnelles. En effet, le non-respect d'exigences techniques peut être un frein majeur à l'adoption d'un nouveau système.

²⁵ Pour cet auteur, toute formulation (d'un but, d'un besoin, d'une contrainte, etc.) est considérée comme une exigence. Afin de ne pas amener de confusion dans ce mémoire et pour être en adéquation avec la définition de l'exigence que nous avons retenue, nous n'utilisons ce terme que pour l'énoncé d'aptitudes du système qui répondent aux caractéristiques de qualité définies. Dans un souci de compréhension les différentes classes ont été renommées (les noms proposés par l'auteur sont affichés entre parenthèses et en italique).

La figure 3.1 décrit les relations existant entre ces catégories. Elle identifie également trois modèles du futur système : le document de vision et de portée, le document de cas d'utilisation et le référentiel d'exigences. Chacun de ces modèles constitue une perception répondant à un objectif précis. Le *document de vision et de portée* décrit les buts qui sont la raison d'être du système. Le *document de cas d'utilisation* formalise les comportements du système attendus par les utilisateurs. Les exigences fonctionnelles sont dérivées de ces cas d'utilisation. Le *référentiel d'exigences* décrit les exigences fonctionnelles et non fonctionnelles. Il constitue le contrat entre le maître d'ouvrage et le maître d'œuvre et il sert à vérifier que le système livré possède les aptitudes attendues.

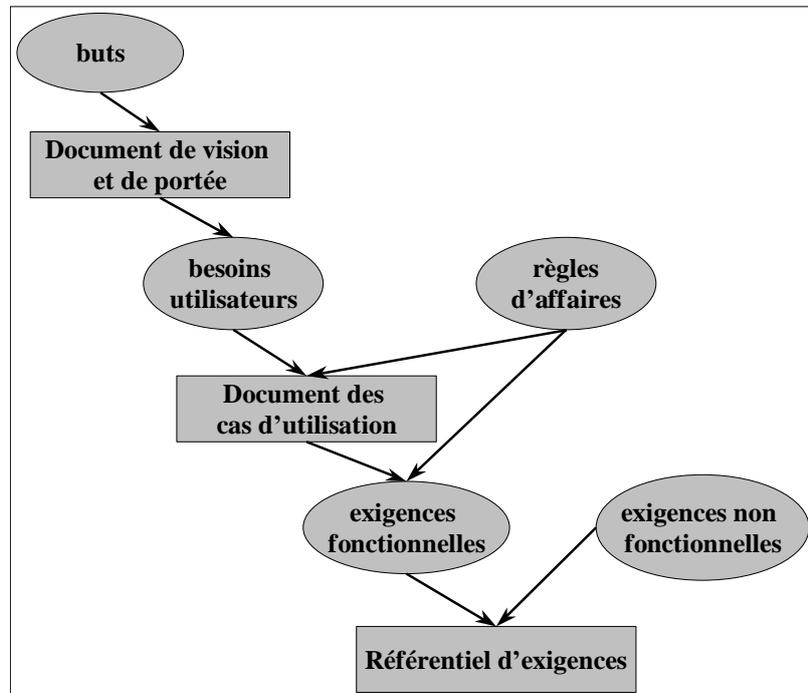


Figure 3.1 : Formalisation des buts, des besoins et des contraintes (adapté de [Wiegiers, 03])

Cette figure détermine les relations entre les buts, les besoins et les exigences. Un « besoin utilisateur » répond à un « but ». Cependant, aucune antériorité n'est préétablie. En effet, le processus d'identification et de formalisation des exigences est itératif. Un « but » peut être identifié suite à la définition d'un « besoin utilisateur » pour lequel on se sera posé la question suivante : « Pourquoi ce besoin utilisateur a-t-il lieu d'exister ? ». Ainsi, la détermination des buts, des besoins et des contraintes a une chronologie variable. Cependant, il est nécessaire d'identifier un minimum de buts pour initialiser le projet d'élaboration du futur système.

Les exigences peuvent être décomposées en plusieurs niveaux. Par exemple, l'exigence « le système gère des versions de documents » peut être décomposée en « le système permet à l'utilisateur de créer des versions de documents », « le système demande à l'utilisateur de rédiger un commentaire pour chaque nouvelle version d'un document existant » et « le système stocke les anciennes versions d'un document ».

Les caractéristiques suivantes permettent de vérifier que le référentiel d'exigences est formulé selon les règles de l'art [Constantinidis, 10] :

- *Complétude* : l'ensemble des exigences couvre la totalité du problème,
- *Cohérence* : il n'y a ni conflit ni de redondance entre deux exigences,
- *Traçabilité* : toute exigence peut être tracée vis à vis d'une exigence de plus haut niveau ou d'un but,
- *Maintenabilité* : un historique des modifications permet de réécrire une exigence,
- *Priorisation* : il est possible d'indiquer un ordre de priorité entre les exigences,
- *Monosémie* : tout terme utilisé a la même signification pour tous les lecteurs dans tout le document.

Pour être cohérent, le référentiel d'exigences ne doit pas contenir d'exigences redondantes. Il doit ainsi être composé d'exigences d'un seul niveau de décomposition. Le référentiel d'exigences ne doit pas contenir d'exigences en conflit. Il est ainsi le résultat d'un consensus entre les différentes parties prenantes qui peuvent avoir des besoins contradictoires.

La caractéristique de traçabilité exprime la capacité du référentiel à être validé. Dans le processus d'ingénierie de définition des besoins, la *validation d'une exigence* consiste à s'assurer qu'elle est en adéquation avec le but dont elle est dérivée. Au regard des trois modèles présentés précédemment [Wieggers, 03], le document de vision et de portée sert à valider le document des cas d'utilisation et le référentiel d'exigences. Ainsi, chaque cas d'utilisation doit contribuer à l'atteinte d'un but [Cockburn, 00].

Le référentiel d'exigences constitue une partie du *cahier des charges* (CDC) du futur système. Le CDC formalise les buts et les besoins des utilisateurs représentés par le maître d'ouvrage ainsi que les contraintes à respecter. Ce document a plusieurs fonctions [AFNOR, 91]. Il constitue un contrat entre le maître d'ouvrage et le maître d'œuvre. Il est également un document de communication. En effet, en supportant le dialogue entre les différentes parties prenantes, il favorise chez le maître d'œuvre une conception et une réalisation efficaces. Enfin, le CDC est la formalisation pérenne qui identifie les buts, les besoins et les contraintes définis pendant le projet. Cette formalisation s'avère utile lorsque des modifications sont envisagées sur le système ou lorsqu'un changement de ce dernier est étudié au vu de nouvelles opportunités sur le marché. La construction du CDC doit ainsi permettre une compréhension du besoin et des contraintes la plus juste possible entre les parties prenantes. Pour cela, le CDC est composé des trois modèles du futur système (buts, cas d'utilisation, exigences), de glossaires, etc. Il inclut le référentiel d'exigences du système mais également les exigences liées au projet (coûts, délais, formation...). Il est à noter que, dans ce document, chaque exigence fonctionnelle doit être justifiée par l'identification du but et des cas d'utilisation dont elle est dérivée.

2.2 Quels langages pour modéliser les buts, les cas d'utilisation et les contraintes ?

Ce paragraphe présente les différents langages utilisés pour formaliser les buts, les cas d'utilisation et les exigences.

2.2.1 Langage usuel de formalisation des buts

Les *buts* sont usuellement formalisés sous forme textuelle. Ils sont structurés sous forme de graphes de réduction ET/OU. Une réduction ET associe un but B à un ensemble de sous-buts B1, B2...Bn qui doivent tous être satisfaits pour que B le soit. Une réduction OU associe un but B à un ensemble B1, B2,..., Bn de sous-buts tels que la satisfaction de l'un d'entre eux assure la satisfaction de B [Rolland, 03].

Le tableau 3.1 présente un exemple de décomposition d'un but en sous-buts [Zancul, 09].

4- gérer les structures des produits
4.1- gérer des fiches articles
4.2- classifier les produits
4.3- construire et visualiser les structures du produit
4.3.1- définir le contexte de la structure du produit
4.3.2- définir le type de la structure du produit
4.3.3- définir la vue métier de la structure produit
4.3.4- stocker les structures du produit de la proposition du client
4.3.5- construire et visualiser les structures du produit
4.3.5.1- gérer des structures du produit avec l'aide d'un navigateur graphique
4.3.5.2- accéder à tous les objets de la structure du produit à partir du navigateur graphique
4.3.5.3- accéder à la structure du produit à partir des modèles 3D CAO
4.3.6- ...
4.4- gérer des variantes dans la structure du produit
4.5- gérer des configurations de la structure du produit

Tableau 3.1 - Exemple de décomposition d'un but [Zancul, 09]

2.2.2 Langage usuel de formalisation des cas d'utilisation

Les *cas d'utilisation* sont usuellement formalisés sous forme textuelle. Les éléments habituels constituant un cas d'utilisation sont : acteur principal, autres acteurs, déclencheur, description, préconditions, post-conditions, scénario nominal, scénario alternatif. Les cas d'utilisation peuvent être illustrés graphiquement par des diagrammes de cas d'utilisation, des diagrammes d'activité, des diagrammes de séquence ou des "interaction overview diagrams" d'UML [Roques, 09][[OMG, 11].

L'exemple suivant a été construit à partir d'un cas d'utilisation issu du document des services PLM définis par l'OMG [OMG, 11].

Cas d'utilisation : « Parcourir en descendant les données relatives à la structure d'un produit »

Description du cas d'utilisation :

En partant de la structure d'un produit, ce processus permet à l'utilisateur d'avoir une visualisation des données relatives au produit (incluant les données documentaires) correspondant au profil type de l'utilisateur.

Acteurs :

rôle	description	type
utilisateur	demande des données	personne / système
système PLM	fournit les données	système

Tableau 3.2 : Acteurs du cas d'utilisation
"Parcourir en descendant les données relatives à la structure d'un produit"

Scénario nominal :

- 1- L'utilisateur envoie une requête au système PLM pour une structure et un niveau de profondeur donnés.
Les actions suivantes sont répétées dans le système PLM pour chaque nœud de la structure du produit dans le périmètre de la requête :
- 2- Les autorisations sont vérifiées en regard des données relatives à la requête. Exception : accès refusé.
- 3- Les données sont collectées dans le système PLM.
Fin des actions répétées.
- 4- Le système PLM envoie les données à l'utilisateur sous forme de structure déroulée.

Diagramme de cas d'utilisation :

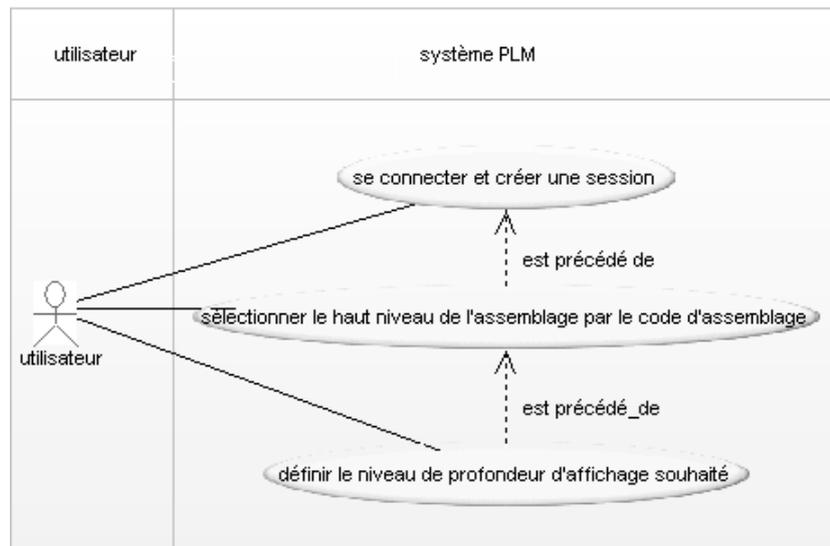


Figure 3.2 : Diagramme de cas d'utilisation
 “ Parcourir en descendant les données relatives à la structure d’un produit ”

Préconditions :

L'utilisateur est connecté et il a les autorisations pour accéder aux données de la requête.
 Le niveau de profondeur de la requête dans la structure est défini.

Post-condition (succès) :

Une structure de produit composée des articles et des données relatives aux articles et du niveau de profondeur souhaité est affichée.

Post-conditions (erreur) :

Un message d'erreur est affiché. L'erreur peut arriver pour les deux raisons suivantes :
 L'utilisateur n'est pas autorisé à accéder aux données.
 Les données ne sont pas disponibles dans le système PLM.

2.2.3 Langage usuel de formalisation des exigences

Les exigences sont usuellement formalisées sous forme textuelle. Pour être correctement formulée, une exigence doit répondre à plusieurs critères. Elle doit être grammaticalement correcte et rédigée à la forme active : sujet, verbe, complément. Le sujet est nécessairement un utilisateur, le système ou une caractéristique du système. Un terme a la même signification pour tout lecteur potentiel. Pour cela, les termes ambigus sont définis dans un glossaire [Constantinidis, 10].

Le langage naturel est commun aux responsables métiers et aux techniciens de l'information. Son utilisation permet de définir les exigences sous une forme aisément compréhensible par tous. Cette perception commune donne la possibilité de construire un consensus entre les parties prenantes lorsque des besoins contradictoires apparaissent. Cependant, son utilisation implique des inconvénients majeurs. Elle peut être source de problèmes dus à des interprétations différentes entre le rédacteur du référentiel d'exigences, le maître d'œuvre et le maître d'ouvrage. Elle peut également conduire à un manque de rigueur du document. Enfin, elle rend difficile l'exploitation des données pour des traitements automatiques.

Pour pallier ces inconvénients, certains travaux [Mallek et al., 11a] [Mallek et al., 11b] [Kamsu Foguem et al., 06] [Fougeres et al., 97] proposent des méthodes permettant de transformer des exigences exprimées en langage naturel en exigences exprimées en langage formel. D'autres travaux proposent des méthodes de formalisation progressive des exigences basées sur un modèle simulable [Nebut et al., 05]. Le principal objectif de la formalisation des exigences est de supporter la communication entre les parties prenantes. Cependant, l'utilisation de langages semi-formels ou

formels permet la mise en œuvre de techniques d'analyse comme la vérification au travers de règles ou de patrons pré-établis ce qui peut s'avérer nécessaire lorsque le volume d'exigences à traiter est important.

Il est à noter que la vérification de la cohérence du référentiel d'exigences implique que toutes les exigences (fonctionnelles et non fonctionnelles) soient exprimées à partir d'un même langage. Dans le cas contraire, il est en effet difficile de détecter les conflits et les redondances entre exigences.

Lorsque le volume d'exigences est important ou que les exigences sont soumises à de fréquentes évolutions, il peut être pertinent d'utiliser un logiciel pour gérer le référentiel d'exigences. Ce logiciel structure les exigences en les reliant de plusieurs façons. La traçabilité verticale identifie la décomposition des exigences et relie ces dernières aux buts. La traçabilité horizontale relie les versions successives d'une même exigence [Constantinidis, 10]. Les principaux logiciels du marché font l'objet d'une étude comparative dans [Sud et al., 03]. Il est à noter qu'il existe également plusieurs prototypes, fruit de travaux de recherche (PRO-ART, MENTOR, PRIME-SCREWS, etc.) [Rolland et al., 01].

2.2.4 Formalisation du cahier des charges

Le *cahier des charges* est usuellement formalisé sous forme textuelle. Il contient toutes les rubriques qui sont nécessaires à la compréhension des buts, des besoins et des contraintes. La plupart des patrons de cahier des charges existants comportent cinq à dix parties découpées en dix à trente paragraphes : motivation du projet, contraintes du projet, exigences fonctionnelles, exigences non fonctionnelles, ... Parmi ces patrons, nous pouvons citer celui d'IEEE 830 [IEEE, 93], celui de l'AFNOR X50-151 [AFNOR, 91], celui de Volere [Atlantic Systems Guild Inc., 05] et celui de Wiegers [Wiegers, 03]. Ces différents patrons font l'objet d'une étude comparative dans [Constantinidis, 10].

2.2.5 Autres langages existants

Dans le cadre de travaux scientifiques, plusieurs langages spécifiques à la formalisation des buts, des besoins et des contraintes ont été développés. Parmi ces langages, nous pouvons citer le langage SYSML [OMG, 12], le langage proposé par [Gnahou et al., 10] qui constitue une extension à SYSML, le langage défini par [Rolland, 03] qui permet de coupler la modélisation de buts à l'écriture de scénarios et le Langage de Description des Exigences (LDE) [Nebut et al., 05] qui supporte la construction de modèles d'exigences avec une sémantique non ambiguë. Nous pouvons citer également l'utilisation d'une grille GRAI R&D qui permet de formaliser les décisions à partir desquelles peuvent être dérivés des cas d'utilisation [Merlo et al., 04] [Vicien et al., 11].

2.3 Quel processus pour élaborer le cahier des charges ?

Quelle que soit la nature du système considéré, le processus d'élaboration du cahier des charges est composé de plusieurs étapes. La première étape consiste à faire des études préliminaires de faisabilité et d'opportunité pour évaluer la pertinence de la réalisation du futur système. La dernière étape consiste à valider le cahier des charges. Pour les systèmes d'information, le but attribué au nouveau système est souvent d'améliorer le système existant. Cette amélioration se fait au moyen d'automatisations de tâches manuelles, d'introductions de nouvelles fonctionnalités, etc. Dans ce cas, un accent tout particulier est mis sur l'étude de l'existant.

La figure 3.3 propose un processus d'élaboration du cahier des charges d'un système d'information [Constantinidis, 10]. Ce processus détermine les principales étapes de l'élaboration et il identifie les documents clés dont l'utilisation est préconisée²⁶.

Ce processus commence par l'étape *cadrer* qui a pour donnée d'entrée une lettre de mission du maître d'ouvrage. Cette étape consiste à identifier les buts, les parties prenantes et le périmètre relatifs au futur système. Elle a également pour objectif de lister les cas d'utilisation concernés par ce futur

²⁶ Nous adaptons ce processus au vocabulaire utilisé précédemment à savoir : document de vision et portée, document des cas d'utilisation et référentiel d'exigences (cf. § 2.1).

système. Ces différentes informations sont formalisées dans le *document de vision et portée*. L'étape **planifier** consiste à choisir le processus d'élaboration du cahier des charges, le patron qui servira à le construire et les techniques et les outils qui seront utilisés. Elle consiste également à planifier les jalons importants et les plages de travail. L'étape **analyser l'existant** consiste à construire le modèle du système existant (modèle « AS IS ») et à identifier les améliorations nécessaires. Cette analyse peut remettre en cause les buts ou le périmètre définis initialement. Une boucle retour d'itération est alors faite vers l'étape *cadrer*.

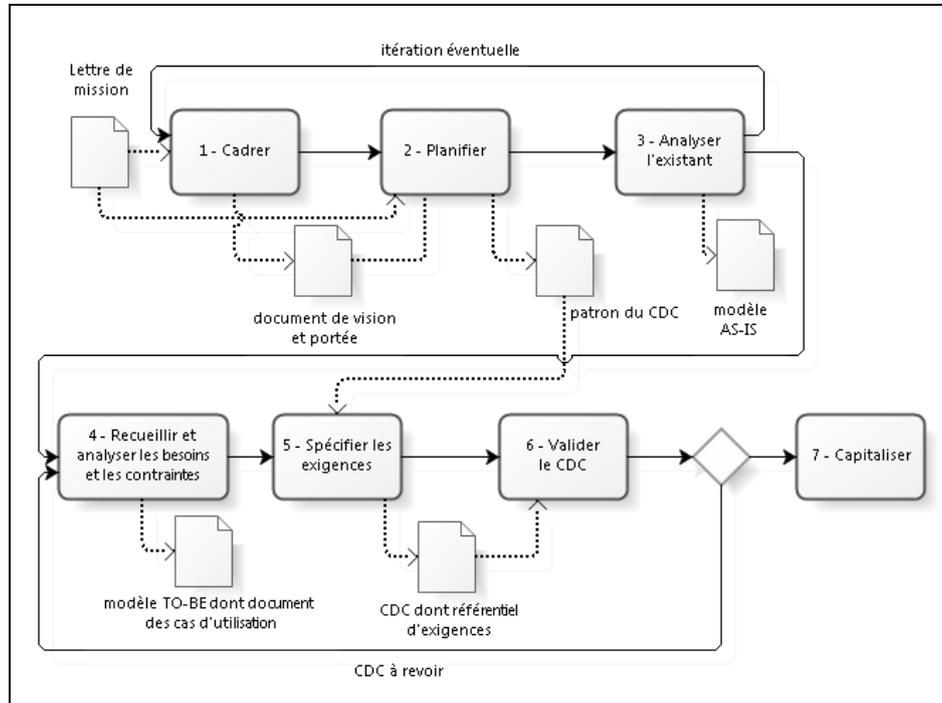


Figure 3.3 : Processus d'élaboration du Cahier des Charges (CDC) d'un système d'information [Constantinidis, 10]

L'étape **recueillir et analyser les besoins et les contraintes** a pour objectif de définir le modèle du futur système (modèle « TO BE ») ; ce modèle englobe le *document des cas d'utilisation*. Tout d'abord, les besoins et les contraintes sont recueillis grâce à l'interview des différents acteurs impliqués dans le fonctionnement du système. Une synthèse et une analyse sont alors réalisées par des groupes de travail pour construire le modèle « TO BE ». L'étape **spécifier les exigences** consiste dans un premier temps à formaliser et à vérifier le *référentiel d'exigences* et, dans un deuxième temps, à élaborer le *cahier des charges*. L'étape **valider le CDC** s'assure que les exigences définies répondent aux buts identifiés dans le document de vision et portée. Durant cette validation, s'il s'avère que des éléments sont à revoir, un retour est fait vers l'étape *recueillir et analyser les besoins et les contraintes*. L'étape **capitaliser** formalise les enseignements tirés du projet : mise à jour de la procédure, des documents standards à utiliser, etc. Ces étapes sont séquentielles mais il existe des boucles de rétroaction. En effet, il est difficile d'avoir une perception correcte du futur système dès le début du processus. Ainsi, au fil des étapes, les buts, le périmètre, les exigences... s'affinent et les incohérences et les imprécisions sont identifiées.

Ce processus met en lumière les différentes tâches liées à *l'ingénierie de définition des besoins* à savoir l'identification des exigences, leur formalisation, leur communication, leur validation et la gestion de leurs évolutions [Nuseibeh et al., 00]. La suite de nos travaux se focalise sur les tâches d'identification et de formalisation.

2.4 Les difficultés rencontrées pour élaborer le CDC

Dans le paragraphe précédent, un processus d'élaboration du CDC d'un système d'information a été défini. Cependant, force est de constater que plusieurs difficultés peuvent être rencontrées lors de sa

mise en application. De nombreuses études ont ainsi identifié certaines de ces difficultés [Standish, 95] [van Lamsweerde, 09] [Constantinidis, 10].

Quel que soit le projet défini autour du système d'information, ce dernier doit prendre en compte le fonctionnement actuel de la structure concernée. En effet, l'expérience et le savoir-faire sont sans intérêt s'ils ne peuvent s'appuyer sur une base réaliste pour construire le futur système. Ainsi, une des difficultés rencontrées consiste à analyser le système d'information existant.

En effet, l'étude de l'existant et l'élaboration du CDC de manière générale nécessitent des analyses longues et difficiles. Ces études impliquent de multiples parties prenantes et des informations nombreuses et variées. Ainsi, le manque d'implication des utilisateurs, l'inconstance de la motivation, la difficulté à trouver un langage commun et la complexité de gestion et de structuration des informations sont des freins à l'élaboration du CDC. La rentabilité de l'ingénierie de définition des besoins est également difficile à évaluer ce qui peut amener la direction de l'entreprise à sous-estimer son intérêt et à lui attribuer trop peu de ressources (en temps dégagé et en financement).

La formulation d'un ensemble complet et cohérent d'exigences s'avère également difficile à réaliser pour plusieurs raisons. Tout d'abord, il est nécessaire d'identifier les besoins relatifs à un système immatériel en réussissant à s'abstraire des solutions potentielles. Ceci oblige à penser en termes de problèmes et non en termes de solutions techniques. Les différentes parties prenantes ayant des besoins contradictoires, il faut également arriver à établir des consensus. Enfin, la perception du futur système n'est pas immédiate ; elle s'affine au fil du temps (maturation des idées avec le temps ou compréhension suite à l'élaboration de prototypes). Ainsi, certaines exigences ne sont identifiées qu'une fois l'élaboration du CDC terminée.

Parmi les risques encourus suite à ces difficultés, nous pouvons citer les sur-spécifications, les sous-spécifications, la négligence de certains profils d'utilisateurs et les spécifications rampantes. Ces dernières sont des spécifications qui remettent indéfiniment en cause ce qui a été fait [Meinadier, 98].

2.5 Conclusion

Cet état de l'art a montré comment identifier et formaliser les buts, les besoins et les contraintes qu'un système d'information doit satisfaire ou respecter. Il a identifié trois modèles qu'il est nécessaire de considérer pour l'élaboration du CDC : le document de vision et portée, le document des cas d'utilisation et le référentiel d'exigences. Le CDC est un document incluant le référentiel d'exigences et différentes autres informations. Suite à cet état de l'art, plusieurs langages ont été identifiés pour construire ces modèles. Un processus dédié à l'élaboration du cahier des charges d'un système d'information a alors été présenté et choisi comme base de travail pour la suite de nos travaux. Pour conclure, les principales difficultés rencontrées lors de l'élaboration du CDC ont été identifiées.

3 Les spécificités du système PLM et les difficultés rencontrées pour élaborer son CDC

Nos travaux ayant pour sujet le système PLM, nous identifions dans ce paragraphe les spécificités de ce système. Dans un premier temps, ceci nous permet de mettre en lumière les difficultés propres à l'élaboration du CDC de ce système. Dans un deuxième temps, l'identification de ces spécificités nous permet d'adapter les éléments définis précédemment (modèles et processus) au cas particulier du système PLM.

Les spécificités du système PLM sont abordées en s'appuyant sur les deux pôles d'observation identifiés dans le chapitre 1 (cf. chapitre 1, § 2.3.2). Ainsi, ce paragraphe décrit les spécificités liées à la prise en compte des besoins et des contraintes des responsables métiers (pôle d'observation organisationnel). Il expose ensuite les spécificités liées à la prise en compte des besoins et des contraintes des techniciens de l'information (pôle d'observation informationnel). Enfin, des retours d'expérience des membres du Club PLM confirment les spécificités définies et permettent d'identifier des difficultés propres à l'élaboration du CDC du système PLM.

3.1 Spécificités liées au pôle d'observation organisationnel

Ce paragraphe a pour objectif d'identifier les différentes caractéristiques propres au système PLM afin de définir par la suite leur influence sur l'élaboration du cahier des charges.

3.1.1 Spécificités relatives au périmètre du système PLM

Le but du système PLM est de fédérer le partage d'informations afin de créer un environnement centré sur le produit. Pour atteindre ce but, ce système doit être en interaction avec toutes les parties prenantes produisant ou utilisant une représentation du produit dans une étape de son cycle de vie. Il doit rassembler et structurer les informations relatives au produit.

Pour cela, d'une part, le système PLM doit être capable de contrôler la création, la modification et la suppression des représentations utilisées dans les différentes étapes du cycle de vie. Ainsi, il doit fournir à chaque responsable métier un accès sécurisé aux modèles adaptés à son niveau de visibilité (cf. chapitre 2 § 2.4).

D'autre part, les interfaces entre le système PLM et les autres systèmes manipulant des représentations du produit doivent être clairement identifiées. Ainsi, les exigences d'interface du système PLM doivent décrire de façon très précise les formats des différentes représentations échangées.

3.1.2 Spécificités relatives à la définition du modèle métier

Le système PLM doit garantir la cohérence entre les différentes représentations du produit en s'appuyant sur la définition d'interactions pouvant exister entre les informations liées au produit. Le **modèle métier PLM*** identifie les informations liées à la gestion des diverses représentations du produit. Ce modèle doit donc d'une part identifier et structurer l'ensemble des informations utilisées dans les différentes étapes du cycle de vie des produits et, d'autre part, définir les processus qui sont associés à leur gestion. Comme illustration de ces propos, la figure 3.4 montre un extrait du modèle métier du système PLM d'une entreprise.

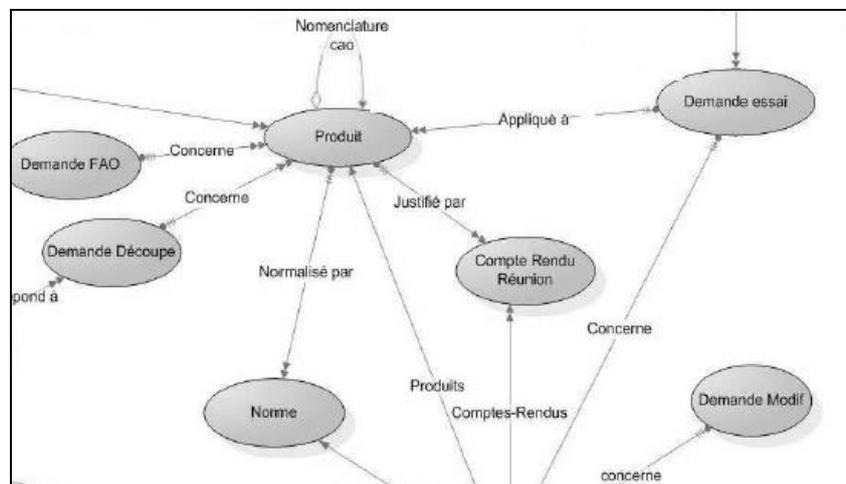


Figure 3.4 : Extrait du modèle métier de la société Duqueine mis en oeuvre au moyen de la personnalisation de leur logiciel PLM (Témoignage Journée du PLM 2008)²⁷

Suite à ce constat, le système PLM doit satisfaire d'une part des exigences liées à la nécessité de disposer d'un langage de définition du modèle métier (première catégorie d'exigences) et d'autre part des exigences liées à la nécessité de pouvoir instancier ce modèle métier pour créer des représentations des produits de l'entreprise (deuxième catégorie d'exigences).

Les exigences de la première catégorie d'exigences, désignées *exigences de modélisation*, précisent que le système PLM doit posséder les aptitudes pour construire le modèle métier de l'entreprise.

²⁷ <http://www.journeedulplm.fr/uploads/file/duqueine-composite.pdf>

Au niveau structurel, le langage de définition doit disposer des concepts nécessaires à la description de la structure des informations liées à la gestion du cycle de vie. Ce langage doit donc offrir la possibilité de définir :

- des types d'objets,
- des attributs pour les types d'objets,
- des relations entre les types d'objets,
- des contraintes structurelles.

En d'autres termes, le système PLM doit fournir un langage pour élaborer une structure de données répondant aux besoins spécifiques de l'entreprise. La figure précédente (cf. figure 3.4) permet d'illustrer ce type d'exigences :

- « le système gère des types d'objets Produit, Demande FAO, Norme, etc. » : exigence définissant les différents types d'objets manipulés par l'entreprise,
- « tout produit est défini par une référence, une version, une désignation, un statut, un auteur, une date de création » : exigence définissant les attributs des différents types d'objets,
- « un Compte Rendu peut être relié à un Produit à l'aide d'une relation « justifié par » : exigence définissant les relations possibles entre les types d'objets.

Au niveau fonctionnel et comportemental, le langage de définition doit disposer des concepts nécessaires à la description des différents processus associés à la gestion de la structure des informations définie précédemment. Le langage doit donc offrir la possibilité de définir :

- des cycles de vie pour les types d'objets,
- des processus aptes à manipuler, modifier, diffuser des objets de différents types,
- des règles métiers de création de version, de passage de statut, de validation ou de structuration,
- des contraintes fonctionnelles et comportementales.

Les exemples suivants illustrent deux exigences liées aux statuts (cf. figure 3.4) :

- « le produit passe d'un statut initial à un statut valide »,
- « pour passer à un statut valide, tous les éléments de nomenclature d'un produit doivent être valides ».

Les exigences de la deuxième catégorie d'exigences, désignées *exigences d'instanciation*, précisent que le système PLM doit posséder les aptitudes pour créer des représentations de produits demandées par les différents acteurs. Ces représentations ont pour objectif de décrire une situation ou un comportement particulier. Par exemple (cf. figure 3.4) :

- La table F7895 a été validée par le compte-rendu de la réunion de validation du 12/03/2012,
- La matière utilisée pour fabriquer le pied P8975 respecte la norme ISO xxxx.

3.1.3 Conclusion

Ce paragraphe a identifié les principales exigences liées au pôle d'observation organisationnel. Le système PLM s'appuie toujours sur différents logiciels pour automatiser certains traitements ou pour assister les responsables métiers dans leur prise de décision. De plus en plus d'entreprises s'appuient sur un logiciel PLM (Audros, Windchill, Teamcenter, etc.) pour instrumenter une partie de leur système PLM (cf. chapitre 1, § 4.2.2). A travers la définition des spécificités du pôle d'observation informationnel, le paragraphe suivant présente les exigences liées à l'utilisation de ce type de logiciel.

3.2 Spécificités liées au pôle d'observation informationnel

Avant de présenter les exigences liées à ce pôle d'observation, il nous semble important de rappeler certaines spécificités du logiciel PLM. Le déploiement d'un logiciel PLM du marché (Audros, Windchill, etc.) diffère par rapport à d'autres applications logicielles telles que les progiciels de gestion intégrée (ERP) et les progiciels de gestion de la relation client (CRM).

En effet, un ERP a pour objectif d'intégrer plusieurs fonctions de gestion de l'entreprise (gestion de production, gestion des achats, gestion comptable, etc.) afin que les différents traitements s'exécutent à partir d'un ensemble de données cohérent. La plupart du temps, l'ERP est défini comme une

structure modulaire où chaque module est associé à un type de gestion (module de gestion de production, module de gestion des achats, module de gestion comptable, etc.). Ces modules s'appuient sur des modèles structurants qui intègrent les principes de gestion à mettre en œuvre. Par exemple, le module de gestion de production d'un ERP s'appuie sur le modèle MRP II (Manufacturing Resource Planning) pour suggérer soit des ordres d'achat de composants ou de matières premières, soit des ordres de fabrication des produits à réaliser. De même, le module de gestion comptable s'appuie sur un modèle qui définit les principes de gestion définis par le plan comptable. De ce fait les structures de données et les processus sont prédéfinis par les éditeurs de logiciels et le déploiement de l'application logicielle au sein de l'entreprise s'effectue grâce à un ensemble plus ou moins important de paramètres.

Dans le cas du logiciel PLM, les éditeurs ne proposent pas de modèles structurants²⁸ pour gérer les informations liées aux différentes étapes du cycle de vie du produit. En effet, les solutions du marché sont des applications qui offrent un ensemble de services de base comme par exemple la gestion de documents grâce à un « coffre-fort » pour contrôler les différents accès. Les différentes fonctionnalités sont souvent regroupées au sein de modules dont le choix est défini lors de l'étape de configuration du logiciel (module de gestion de projet, module de gestion du portefeuille des produits, etc.). Tout comme le logiciel ERP, le logiciel PLM propose un ensemble plus ou moins important de paramètres pour adapter certaines fonctionnalités aux besoins de l'entreprise. Comme le logiciel PLM n'englobe pas de modèle de gestion particulier, l'éditeur offre la possibilité à l'entreprise de construire son propre modèle de gestion incluant d'une part la définition de la structure de données et, d'autre part, la définition des processus qui lui sont associés. Les mécanismes proposés pour construire ce modèle dépendent des solutions logicielles. Certains éditeurs proposent des langages pour que l'entreprise puisse définir son propre modèle de gestion des informations liées au produit (Audros, Teamcenter, par exemple). Dans d'autres cas, le modèle de gestion est élaboré à partir de « patrons » prédéfinis et de mécanismes spécifiques (Windchill, SmartTeam, par exemple).

La phase de déploiement d'un logiciel PLM s'effectue donc en trois étapes. Une première étape de configuration où l'entreprise définit les modules du logiciel PLM qu'elle souhaite utiliser. Une deuxième étape de personnalisation où l'entreprise construit son modèle de gestion, désigné **modèle personnalisé du logiciel PLM***, décrivant les données et les processus qu'elle souhaite gérer avec le logiciel PLM. Une troisième étape de paramétrage où l'entreprise définit ses caractéristiques d'usage des différents modules.

3.2.1 Spécificités liées à un logiciel

Ce paragraphe présente d'une part les exigences relatives à la qualité du logiciel (onze facteurs de Mc Call [Mc Call et al., 77]) et d'autre part les exigences relatives à la flexibilité du logiciel (quatre types de flexibilité identifiés par Henri Chelli [Chelli, 03]).

Facteurs de qualité d'un logiciel

Dans ce paragraphe, la description des onze facteurs est réalisée à l'aide d'une classification selon trois points de vue. Le premier point de vue concerne les aptitudes liées au fonctionnement du logiciel : la pertinence, l'intégrité, la facilité d'utilisation, la fiabilité et l'efficacité. Le deuxième point de vue concerne les aptitudes liées aux évolutions du logiciel : la maintenabilité, la testabilité et l'adaptabilité. Le troisième point de vue est lié à la transition vers un autre système et au couplage avec des systèmes existants : la portabilité, la réutilisabilité et la couplabilité. Grâce à une analyse des retours d'expérience du Club PLM, la description de chaque facteur précise le degré d'importance du facteur pour le logiciel PLM.

Facteurs liés au fonctionnement du logiciel :

La **pertinence** définit la capacité du logiciel à répondre aux buts fixés.

L'**intégrité** est l'aptitude d'un logiciel à être protégé contre tout accès par des personnes non autorisées. Pour le logiciel PLM, la sécurisation des données est un but important. En effet, les

²⁸ Certains éditeurs proposent des modèles « standards » déduits de leur expérience.

fonctionnalités mises en œuvre doivent garantir un sentiment de confiance des utilisateurs envers l'intégrité des informations stockées dans le système. Une donnée est ainsi accessible de différentes manières (lecture, écriture, destruction) selon son état de maturité (en création, valide, obsolète), selon le contexte d'un projet et selon le profil de l'utilisateur.

La maniabilité ou **facilité d'utilisation** est l'aptitude du logiciel à être convivial et simple d'emploi. Dans le logiciel PLM, la sécurisation des données peut rendre les traitements difficiles à l'usage. Ainsi, un des critères employés par les utilisateurs pour juger ce facteur est le nombre de « clics²⁹ » qu'il est nécessaire d'effectuer pour saisir, accéder, modifier ou supprimer une information. Un autre critère couramment employé pour évaluer ce facteur est le nombre d'Interfaces Homme-Machine (IHM) que les utilisateurs doivent manipuler pour réaliser leurs tâches. Ainsi, il est recommandé de proposer des IHM communes à plusieurs logiciels (logiciel PLM, logiciel ERP, etc.) et d'avoir une homogénéité dans l'esthétique et l'agencement des interfaces des différents logiciels du système d'information de l'entreprise.

La **fiabilité** est l'aptitude du logiciel à accomplir sans défaillance l'ensemble des fonctions pour une durée d'utilisation donnée. Dans un logiciel PLM, ce facteur est souvent jugé par la stabilité de l'application (peu d'incidents et de pannes) malgré un volume important de données à gérer.

L'**efficacité** est l'aptitude du logiciel à minimiser l'utilisation des ressources informatiques disponibles (processeurs, mémoires, etc.).

Facteurs liés à l'évolution du logiciel :

La **maintenabilité** est l'aptitude d'un logiciel à faciliter la localisation et la correction d'erreurs. Le regroupement des données et l'intégration des fonctions dans une même application répondent au besoin de faciliter la maintenance. Le logiciel PLM étant souvent un logiciel du marché, différentes versions du logiciel sont disponibles au fil du temps. Un des critères de maintenabilité est alors la capacité à évoluer vers une version supérieure de manière fiable et efficace en conservant d'une part le paramétrage, la configuration et la personnalisation en cours et, d'autre part, les développements spécifiques réalisés.

La **testabilité** est l'aptitude d'un logiciel à se prêter à une vérification d'adéquation aux exigences.

L'**adaptabilité** est l'aptitude d'un logiciel à faciliter l'adjonction de nouvelles fonctions ou la modification voire la suppression de fonctions existantes. Ce facteur est très important pour le logiciel PLM pour deux raisons. La première est la conséquence d'un déploiement incrémental du logiciel PLM. En effet, le périmètre des données et des fonctions gérées par le logiciel s'agrandit au fil du temps. Il est ainsi nécessaire de pouvoir incorporer de nouveaux types d'information, de nouveaux processus et de nouvelles fonctions alors que le logiciel est déjà en utilisation. La deuxième raison est liée à la couplabilité. En effet, il est nécessaire que les couplages entre le logiciel PLM et d'autres logiciels soient capables d'évoluer lors des changements de version du logiciel PLM ou du logiciel couplé.

Facteurs liés à l'interopérabilité :

La **couplabilité** est l'aptitude du logiciel à communiquer ou interagir avec d'autres applications logicielles. Ce facteur intervient à deux phases du projet de déploiement d'un logiciel PLM. La première phase concernée est celle de migration des données existantes. En effet, il est intéressant avant toute utilisation du logiciel PLM de pouvoir importer dans la base vierge de ce dernier des données relatives aux produits existants à partir d'autres applications logicielles (PDM, ERP, etc.). Cette migration permet au moment du déploiement du logiciel PLM de pouvoir dès le début de son utilisation effectuer des recherches sur les produits existants et de pouvoir faire des réutilisations lors de la conception de nouveaux produits. La deuxième phase concernée est liée à l'utilisation du logiciel PLM. Comme nous l'avons vu à plusieurs reprises, ce logiciel doit être capable d'échanger des données avec les différentes applications logicielles existant dans l'entreprise. Par exemple, le couplage avec le logiciel CAO assure des transferts de données et de représentations de conception (plans, modèles 3D, etc.). De même, le couplage avec l'ERP garantit une cohérence entre les données

²⁹ Un clic correspond généralement à l'exécution d'une action élémentaire.

mémorisées dans le logiciel PLM et celles mémorisées dans la base de données techniques de l'ERP. Enfin, le couplage avec le site Web de l'entreprise assure une cohérence entre les données affichées par le site (catalogue des produits, par exemple) et celles mémorisées au sein du logiciel PLM. Ce facteur est donc essentiel pour le logiciel PLM car ce dernier doit être en interaction avec l'ensemble des applications logicielles impliquées dans la gestion du cycle de vie du produit.

La *portabilité* est l'aptitude à transférer le logiciel dans un autre environnement.

La *réutilisabilité* est l'aptitude d'un composant logiciel à être réutilisé dans des applications différentes.

Les différents types de flexibilité

Un des buts de l'entreprise identifié dans ces travaux concerne le maintien, voire le développement, de l'agilité opérationnelle. Pour assurer cette agilité, il est nécessaire d'introduire la notion de flexibilité lors de la définition des exigences du système PLM. L'agilité opérationnelle dépend de quatre types de flexibilité [Chelli, 03]. La *flexibilité technologique* est délivrée par les langages, les outils et les méthodes des technologies de l'information. Pour illustrer les outils et les méthodes, nous pouvons citer l'utilisation d'ateliers de génie logiciel qui génèrent un programme à partir de la description précise d'un besoin métier. La *flexibilité structurelle* vise à mettre en place des composants qui facilitent l'intégration de nouvelles fonctionnalités. La *flexibilité potentielle* consiste à développer et introduire des fonctionnalités qui pourront servir potentiellement dans le futur. La *flexibilité topographique* consiste à créer une segmentation appropriée qui permet de focaliser les impacts des évolutions de la réalité opérationnelle sur un nombre réduit de composants informatisés.

3.2.2 Spécificités du logiciel PLM

Dans le logiciel PLM, les informations sont gérées sous forme d'objets structurés entre eux à l'aide de liens. L'objet manipulé peut être un document tangible (un plan encapsulé dans un objet, par exemple) ou un concept (un article, par exemple). Afin de définir un environnement centré autour du produit, la structure d'articles qui constitue le produit doit être utilisée comme « colonne vertébrale » pour organiser les différentes informations relatives au produit (plans, notices, devis, etc.).

Le pôle d'observation organisationnel a montré que le fonctionnement du système PLM s'appuie sur le modèle métier PLM. Un facteur essentiel pour le logiciel PLM est donc sa capacité à implémenter ce modèle métier grâce à la construction du modèle personnalisé dans le logiciel PLM.

3.2.3 Conclusion

Ce paragraphe a identifié les principales exigences du pôle d'observation informationnel. Pour ce faire, nous nous sommes appuyés, dans un premier temps, sur un ensemble de facteurs de portée générale avant d'identifier un facteur spécifique au logiciel PLM.

3.3 Retours du Club PLM : confirmation des spécificités identifiées et mise en lumière de difficultés propres à l'élaboration du CDC du système PLM

Au moyen de retours d'expérience issus du Club PLM³⁰, ce paragraphe a pour objectifs, d'une part, de valider certaines des spécificités identifiées dans les paragraphes précédents (cf. § 3.1 et § 3.2) et, d'autre part, de mettre en lumière les difficultés propres à l'élaboration du CDC du système PLM.

La première difficulté identifiée est relative à la mauvaise connaissance de l'existant. Connaître l'existant permet de définir clairement le périmètre du système PLM. Suite aux retours d'expérience du Club PLM, on constate qu'il est difficile d'avoir une perception correcte de la situation existante ce qui peut avoir des conséquences sur la définition du système PLM. Par exemple, lors d'une analyse de

³⁰ Ces retours sont le fruit d'échanges réalisés dans le cadre des neuf réunions ayant eu lieu depuis la création du Club PLM (cf. chapitre 1 § 5). Les sujets abordés concernaient différents aspects des projets PLM : gestion du changement, définition des modèles dans les logiciels PLM, qualité des données, etc.

l'existant, une entreprise a oublié d'interviewer certains acteurs ayant des activités spécifiques autour du produit : le niveau de visibilité de ces acteurs n'a donc pas été intégré au sein du système PLM. Une autre expérience a montré qu'il pouvait exister un décalage entre les processus formalisés au sein de l'entreprise et les processus réellement appliqués. De ce fait, le niveau de visibilité des acteurs défini au sein du système PLM ne reflète pas la réalité. De même, une expérience a montré que le volume et la qualité des données (exactitude) recensés en début de projet ne sont pas conformes à la réalité. Tout comme pour l'expérience précédente, ce problème induit que les informations fournies aux acteurs ne sont pas pertinentes s'il n'y a pas de remise à niveau préalable de la qualité des données. En effet, si ces données ont été importées telles quelles dans le logiciel PLM, les résultats proposés lors des requêtes sont invalides (principe du « garbage in – garbage out »).

Certains retours d'expérience ont montré la difficulté d'identifier tous les besoins et de rendre les exigences cohérentes. L'aspect transversal du système PLM a pour conséquence que les parties prenantes impliquées dans le projet PLM sont nombreuses et variées. La diversité des cultures, la diversité des besoins, la diversité d'implication et les jeux de pouvoir peuvent engendrer des tensions entre les différentes parties prenantes.

Nous avons vu que le fonctionnement des logiciels PLM est basé sur la création de structures d'objets (cf. paragraphe 3.2.2). Des retours d'expérience ont mis en lumière que l'utilisation de telles structures induit des changements qui sont difficiles à assimiler pour les futurs utilisateurs. Le premier changement est la structuration au moyen d'objets qui remplace la structuration « historique » au moyen de répertoires. Le deuxième changement est la structuration basée sur des concepts intangibles, structuration centrée « article », qui remplace souvent une structuration basée sur des objets tangibles, structuration centrée « plans ». La troisième est le changement de culture qui est lié au partage d'information. En effet, des expériences ont montré que certains utilisateurs avaient du mal à stocker leurs représentations dans une structure commune. Avant le système PLM, ces acteurs transmettaient souvent leurs représentations (plan, 3D, etc.) sous forme de copies et conservait la maîtrise complète des représentations « sources ». Le logiciel PLM permet également de gérer des représentations à différents statuts (en création, en cours de validation, validé, obsolète, etc.). Ceci impose aux utilisateurs de stocker leurs représentations dans la structure dès la création. Certains utilisateurs sont gênés par le fait de rendre public des représentations pouvant contenir des erreurs. D'autres acteurs considèrent que la connaissance des statuts des différentes représentations peut être utilisée comme un indicateur de « rendement d'activités ».

Le facteur de couplabilité a été identifié comme un facteur important pour le logiciel PLM. Dans certains cas, certaines fonctionnalités peuvent être réalisées par deux logiciels différents (logiciel PLM et logiciel ERP, en particulier). Le retour d'expériences a montré que ce constat rendait difficile le positionnement du logiciel PLM par rapport à d'autres logiciels existants au sein de l'entreprise (ERP, GED, CAO, etc.). Il peut être alors ardu de prouver la valeur ajoutée du logiciel PLM.

La faculté de s'abstraire des solutions (i.e. les logiciels PLM) pour définir le cahier des charges (i.e. la formalisation du problème) est également une difficulté importante pour deux raisons. La première est la difficulté de définir les fonctionnalités du système PLM indépendamment des fonctionnalités proposées par les logiciels PLM. Des retours d'expérience ont montré que les exigences fonctionnelles ne sont pas déduites d'un document de cas d'utilisation (comme préconisé dans le § 2.1). En effet, ces exigences sont identifiées à partir de connaissances des fonctionnalités proposées par les solutions du marché. De ce fait, il n'est pas possible de valider le référentiel d'exigences construit étant donné que les buts ne sont pas définis clairement (cf. § 2.1). La deuxième raison est la difficulté d'identifier le « retour sur investissement » du coût induit par l'ingénierie de définition des besoins. Ne sachant pas évaluer ce retour, les entreprises élaborent soit des cas d'utilisation qui ne décrivent que les exigences fonctionnelles qui seront automatisées par le logiciel PLM, soit des cas d'utilisation déduits des modèles « standards »³¹ proposés par les éditeurs. Ce constat induit les mêmes conséquences que celles identifiées précédemment (première raison).

³¹ Rappelons que ces modèles sont définis à partir de l'expérience de l'éditeur (« benchmark » des différentes installations effectuées auprès des clients).

3.4 Conclusion

Dans ce paragraphe, nous avons rappelé les spécificités du système PLM selon les pôles d'observation organisationnel et informationnel. Certaines de ces spécificités ont été confirmées par les retours d'expérience du Club PLM. Ces retours d'expérience ont également permis d'identifier différentes difficultés rencontrées lors de la définition du système PLM. Parmi ces difficultés, la corrélation entre le niveau de maturité de l'entreprise et l'approche adoptée a été mise en lumière.

4 La définition de niveaux de maturité

Les échanges avec le Club PLM ont clairement montré que des prérequis sont nécessaires pour pouvoir définir précisément le système PLM. Ces prérequis traduisent la maturité de l'entreprise selon deux points de vue. Le premier point de vue est lié à l'organisation de l'entreprise et à ses processus métiers. Le second concerne les compétences nécessaires pour définir le futur système PLM.

4.1 Maturité relative à l'organisation de l'entreprise et à ses processus métiers

La maturité relative à l'organisation de l'entreprise et à ses processus métiers concerne des groupes d'individus. Ainsi, dans une même entreprise, des différences de maturité peuvent exister au sein de l'organisation.

L'objectif du système PLM est de supporter le travail collaboratif autour du produit. Avant de mettre en œuvre ce système, il est ainsi nécessaire d'harmoniser les processus métiers afin que les acteurs aient des pratiques et un langage commun. Cette standardisation s'effectue de manière transversale entre les différents services de l'entreprise. Elle doit apporter une perception globale du produit. Ce travail de réingénierie des processus métiers a pour objectif de standardiser mais également de rationaliser pour améliorer les flux d'informations.

Pour que le travail collaboratif soit le plus efficace possible et que les connaissances autour du produit augmentent, un véritable changement de culture est nécessaire. Une prise de conscience du fonctionnement global de l'entreprise doit casser l'habituelle attitude de « client-fournisseur » entre services afin d'instaurer un « tous ensemble » autour du produit. Pour cela, la personne créant l'information doit la mettre au plus tôt à disposition des autres acteurs, même si cette dernière n'est pas finalisée. En effet, même incomplète, l'information peut déjà être utilisée pour faire des simulations en gestion d'approvisionnement et de production par exemple. Cette approche transversale et globale implique des structures d'information cohérentes entre les métiers.

En résumé, l'amélioration du travail collaboratif passe par les améliorations interdépendantes de l'organisation, de la culture d'entreprise et du système d'information. Pour implémenter un système PLM transversal et centré sur le produit, l'entreprise doit posséder une certaine maturité organisationnelle.

4.2 Maturité relative aux compétences nécessaires pour définir le futur système PLM

La maturité relative aux compétences nécessaires à la définition du futur système PLM concerne des individus et des groupes d'individus. Ainsi, dans une même entreprise, des différences de maturité peuvent exister entre services ou entre unités mais aussi entre des personnes d'un même service.

Le paragraphe 3.3 a listé plusieurs freins à la définition des exigences du système PLM : la difficulté d'appréhender une gestion centrée « articles » plutôt que « plans », par exemple. Ces difficultés sont dues au changement de mode de gestion mais également à la faculté des utilisateurs de formaliser leurs exigences. En effet, la formalisation d'exigences nécessite la création de modèles représentant le futur système en étant capable de s'abstraire des logiciels pouvant potentiellement être utilisés pour les implémenter. Cette capacité requiert des compétences spécifiques en analyse du système d'information.

Le paragraphe 3.3 a également mis en lumière le phénomène d'apprentissage ayant lieu lors du déploiement d'un logiciel PLM. Ce phénomène est illustré par l'adage « *c'est en forgeant qu'on devient forgeron* » utilisé régulièrement par les chefs de projet ayant déployé plusieurs logiciels PLM. Dans la définition d'un système PLM, il s'avère ainsi préférable de commencer par formaliser des fonctions simples. Une fois que des compétences dans le domaine ont été développées par la pratique, l'identification et la formalisation d'exigences relatives à des fonctions complexes telles que la gestion de connaissances peuvent être faites.

4.3 Etat de l'art des modèles de maturité PLM existants

A notre connaissance, seul un membre du Club PLM a identifié et mis en œuvre des indicateurs relatifs aux deux perceptions de l'entreprise. La figure 3.5 identifie les différents critères d'évaluation qui ont été identifiés et illustrent les évaluations qui ont été réalisées avant et après le projet PLM. Ces critères sont relatifs à la perception organisationnelle (augmenter la confiance des clients, réduire les délais, etc.) et à la perception informationnelle (accès aux données valides, faciliter l'accès aux 2D / 3D, etc.).

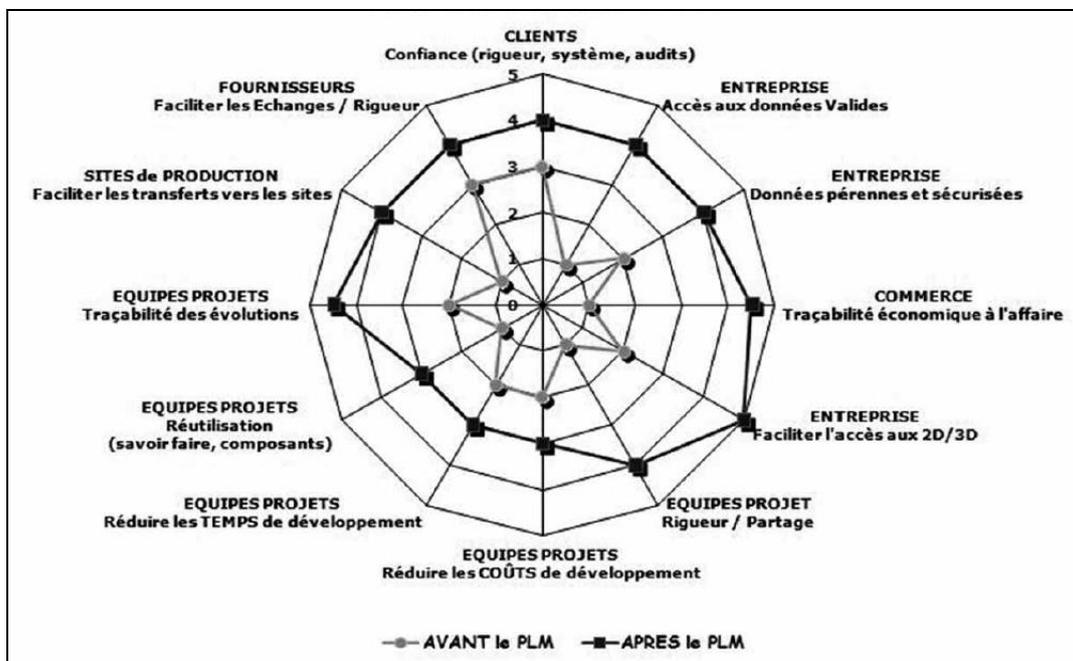


Figure 3.5 : Evaluation des améliorations amenées par le projet PLM dans une entreprise du Club PLM³²

Différents travaux académiques ont pour objet la définition de modèles de maturité PLM [Batenburg et al., 05], [Stark, 2005], [Saaksvuori et al., 08].

Le modèle le plus simple identifie cinq niveaux de maturité en décrivant en quelques lignes des caractéristiques à respecter tant au niveau processus métiers qu'informatisation de ces processus [Saaksvuori et al., 08].

Un autre modèle propose de mesurer quatre niveaux de maturité et cinq pôles d'observation : « stratégie et gouvernance », « management et contrôle », « organisation et processus », « personnes et culture » et « technologies de l'information » [Batenburg et al., 05]. Ce modèle de maturité met particulièrement l'accent sur la cohérence entre les différentes perceptions de l'entreprise. A partir de ce modèle générique, des entreprises correspondant à quatre types d'entreprises ont été évaluées : entreprises d'équipement et de transport, fournisseurs d'outils liés aux technologies de l'information, éditeurs de logiciels et services financiers. Suite à cette évaluation, des modèles de référence ont été

³² <http://www.jiteconline.com/sts/www-jiteconline-com/fch/ds/214.pdf>

construits pour chaque type d'entreprises en se basant sur les valeurs moyennes des évaluations des entreprises appartenant au type concerné.

Enfin, un des modèles les plus complets propose de mesurer quatre niveaux de maturité sur de nombreux pôles d'observation regroupés selon trois points de vue : le point de vue global de l'entreprise, le point de vue développement nouveaux produits et le point de vue gestion des informations liées au produit [Stark, 2005]. Ce modèle est accompagné d'un questionnaire permettant d'évaluer le niveau de maturité de l'entreprise.

Les deux derniers modèles présentés sont riches et complémentaires. Les modèles de référence liés au premier modèle montrent que l'alignement moyen mesuré entre les différentes perceptions varie selon la typologie et la taille de l'entreprise. Le second modèle reflète, à notre avis, le mieux la complexité de l'adoption d'une approche PLM. Les critères définis sont détaillés et très concrets, ce qui permet une bonne compréhension et une différenciation claire de chaque niveau de maturité. Par contre, le résultat de l'évaluation est à contextualiser au type et à la taille de l'entreprise.

Ces différents modèles permettent d'évaluer la maturité de l'organisation de l'entreprise mais pas les compétences nécessaires pour construire le système PLM. Ils identifient les modifications à apporter à l'organisation et au système d'information pour développer l'agilité opérationnelle.

Ces évaluations montrent l'importance de mettre en œuvre des processus transversaux ce qui nécessite de transformer une structure organisationnelle pyramidale et organisée par département en une structure plate et organisée par unité. La formation des collaborateurs est aussi identifiée comme un facteur clé d'évolution des organisations.

5 Vers une méthode d'élaboration du CDC du système PLM

Aujourd'hui, plusieurs travaux traitent de l'ingénierie de définition des besoins relatifs aux systèmes PLM. Certains travaux illustrent l'utilisation de cas d'utilisation sur un exemple de construction de système PLM [Merlo et al., 05] [Eynard et al., 04]. D'autres travaux proposent une définition générique des besoins du système PLM [Zancul, 09] [Le Duigou, 10] [OMG, 11]. Ces travaux formalisent des cas d'utilisation et des hiérarchies de buts mais pas d'exigences.

Après une étude de ces différents travaux, nous proposons dans ce paragraphe de définir un processus complet d'élaboration du cahier des charges d'un système PLM en prenant en compte ces différentes propositions.

5.1 Définition générique des besoins

L'OMG [OMG, 11] définit un ensemble de cas d'utilisation³³ pour identifier les besoins relatifs au système PLM. La définition de ces cas d'utilisation est basée sur les modèles du PDM Schema [Ungerer, 02] et de la norme STEP ISO 10303-214:2000 [ISO, 00]. Les définitions proposées par l'OMG ne contiennent que des cas d'utilisation, les exigences en découlant ne sont pas identifiées.

Une autre définition propose une liste hiérarchisée³⁴ de buts génériques fonctionnels et de buts génériques d'interface entre le système PLM et d'autres systèmes [Le Duigou, 10]. Cette liste a été définie à partir de l'analyse des besoins et des contraintes explicites et implicites de trois entreprises. Elle a été élaborée à partir d'interviews d'experts et d'observations effectuées sur le terrain (observation de l'expert dans l'exercice de ses fonctions et pratique du métier de l'expert).

Enfin la dernière définition propose également une liste³⁵ hiérarchisée de buts génériques fonctionnels et de buts génériques d'interface entre le système PLM et d'autres systèmes [Zancul, 09]. Cette liste a

³³ Plus précisément, l'OMG propose 27 cas d'utilisation.

³⁴ Cette liste est constituée de 7 buts et de 80 sous-buts.

³⁵ Cette liste est constituée de 13 buts décomposés en 1234 sous-buts. Parmi les 13 buts, 6 sont identifiés comme propres aux systèmes PLM et 7 comme des extensions possibles.

été définie à partir des fonctionnalités proposées par cinq éditeurs de logiciels PLM leaders du marché (Dassault, Oracle, PTC, SAP et Siemens).

Les deux listes proposées sont complémentaires. La première liste identifie clairement les buts relatifs à la cohérence entre les différentes représentations du produit (cf. chapitre 2 § 2.4). La seconde liste est plus exhaustive, elle identifie clairement les buts relatifs à la personnalisation du système PLM (cf. § 3.1.2). Dans les deux cas, aucune relation de dépendance (à part celle de composition) n'est définie entre les différents buts. Pour être en cohérence avec l'approche de maturité proposée, les antécédences préconisées entre les buts devraient être identifiées.

Les buts proposés par les deux dernières listes sont présentés dans le tableau 3.3.

Identifiés par Zancul	Identifiés par Le Duigou
1- gérer des idées, des exigences et le portefeuille des produits	
2- gérer des projets	gérer des projets
3- analyser le coût du produit	
4- structurer le produit	gérer le multi vues
5- gérer la configuration	& gérer la configuration
6- gérer les documents	
7- gérer la qualité	
8- répondre aux exigences environnementales	
9- gérer les services et la maintenance	
10- gérer les procédés	gérer les procédés
11- gérer les composants standards et les fournisseurs	gérer la collaboration fournisseur
12- gérer la collaboration	
13- assurer l'interopérabilité	assurer l'interopérabilité
	gérer la collaboration client

Tableau 3.3 : Buts identifiés par [Zancul, 09] et [Le Duigou, 10]

Nous proposons ci-dessous une description des principaux buts identifiés en jaune dans ce tableau.

Gérer des projets : ce but précise que le système PLM doit gérer les ressources, assurer la planification des tâches et fournir des indicateurs permettant d'évaluer l'avancée du projet de développement du produit.

Structurer le produit, gérer la configuration et le multi-vues : ce but précise que le système PLM doit disposer de structures du produit. Si les buts identifiés par les deux auteurs ont des similitudes, ils existent des différences concernant les activités liées à la gestion de la configuration. Par exemple, la gestion des variantes fait partie de la configuration [Le Duigou, 10] ou de la structuration du produit [Zancul, 09].

Gérer les documents : ce but précise que les documents doivent être stockés de manière structurée et qu'ils doivent être caractérisés par des attributs.

Gérer les procédés : ce but précise que les gammes de fabrication doivent décrire l'enchaînement des opérations et doivent identifier les ressources utilisées (atelier, machine, outillage, etc.).

Gérer les composants standards : ce but précise que le système PLM doit disposer de la liste des composants standards des fournisseurs.

Gérer la collaboration : ce but précise que le système PLM doit contrôler l'accès aux informations et aux connaissances, gérer la communication entre les collaborateurs et le travail en équipe.

Assurer l'interopérabilité : ce but précise que le système PLM doit disposer d'interfaces avec d'autres logiciels de l'entreprise : CAO, ERP, logiciel de gestion de projet, etc. Les données échangées peuvent être des fiches articles, des structures produit, des gammes de fabrication, etc.

Gérer la collaboration client : ce but précise que le système PLM doit supporter les échanges d'informations avec les clients.

Ayant décrit les principaux buts, nous allons maintenant identifier les caractéristiques d'une méthode permettant d'élaborer le CDC du système PLM.

5.2 Un processus adapté à l'élaboration du CDC du système PLM

Dans le paragraphe 4, il a été défini qu'il est nécessaire d'avoir une certaine maturité organisationnelle pour pouvoir mettre en œuvre un système PLM transversal et centré sur le produit. Cette maturité implique un changement de culture et une refonte des processus métiers. Il a également été déterminé que les personnes définissant le CDC du système PLM doivent posséder une certaine maturité en analyse du système d'information.

Le processus défini dans le paragraphe 2.3 identifie les étapes propres à l'élaboration du CDC. Nous proposons de déterminer les étapes préliminaires indispensables pour s'assurer que les pré-requis à la définition du système PLM existent bien. La figure 3.6 définit ainsi les étapes d'analyse et de mise à niveau éventuelle de l'organisation. Le processus global proposé prend ainsi en compte la perception organisationnelle et la perception informationnelle de l'entreprise.

Dans la première partie de ce processus global, l'étape **cadrer** consiste à construire le *document de vision et de portée global* qui décrit les buts organisationnels et informationnels relatifs au travail collaboratif et aux connaissances relatives aux produits. Ces buts remettent potentiellement en cause le système d'information mais également les processus métiers de l'entreprise. L'étape **évaluer la maturité** consiste à identifier les points faibles de la situation existante au regard des buts définis. L'étape **identifier les actions à mettre en œuvre** consiste alors à construire un *plan d'actions* listant les processus métiers à redéfinir et les évolutions nécessaires pour le système d'information. L'étape **évaluer la maturité des ressources internes** consiste à évaluer si les ressources internes possèdent les compétences jugées nécessaires (modélisation des processus, concepts de la gestion du cycle de vie produit, ...) pour atteindre les buts définis. Les résultats de cette évaluation sont pris en compte dans l'étape **planifier** pour définir les besoins en formation et les ressources externes nécessaires. L'étape **refonte des processus métiers** est réalisée si une mise à niveau de l'organisation est nécessaire pour atteindre les buts fixés. Cet état des lieux de l'organisation peut être faite au moyen de cadres de modélisation spécifiques à la collaboration relative au produit tels que la grille GRAI R&D par exemple [Pol et al., 05].

Le processus défini se reproduit de manière itérative. En effet, les niveaux de maturité doivent être franchis un par un et de manière cohérente entre les perceptions organisationnelle et informationnelle.

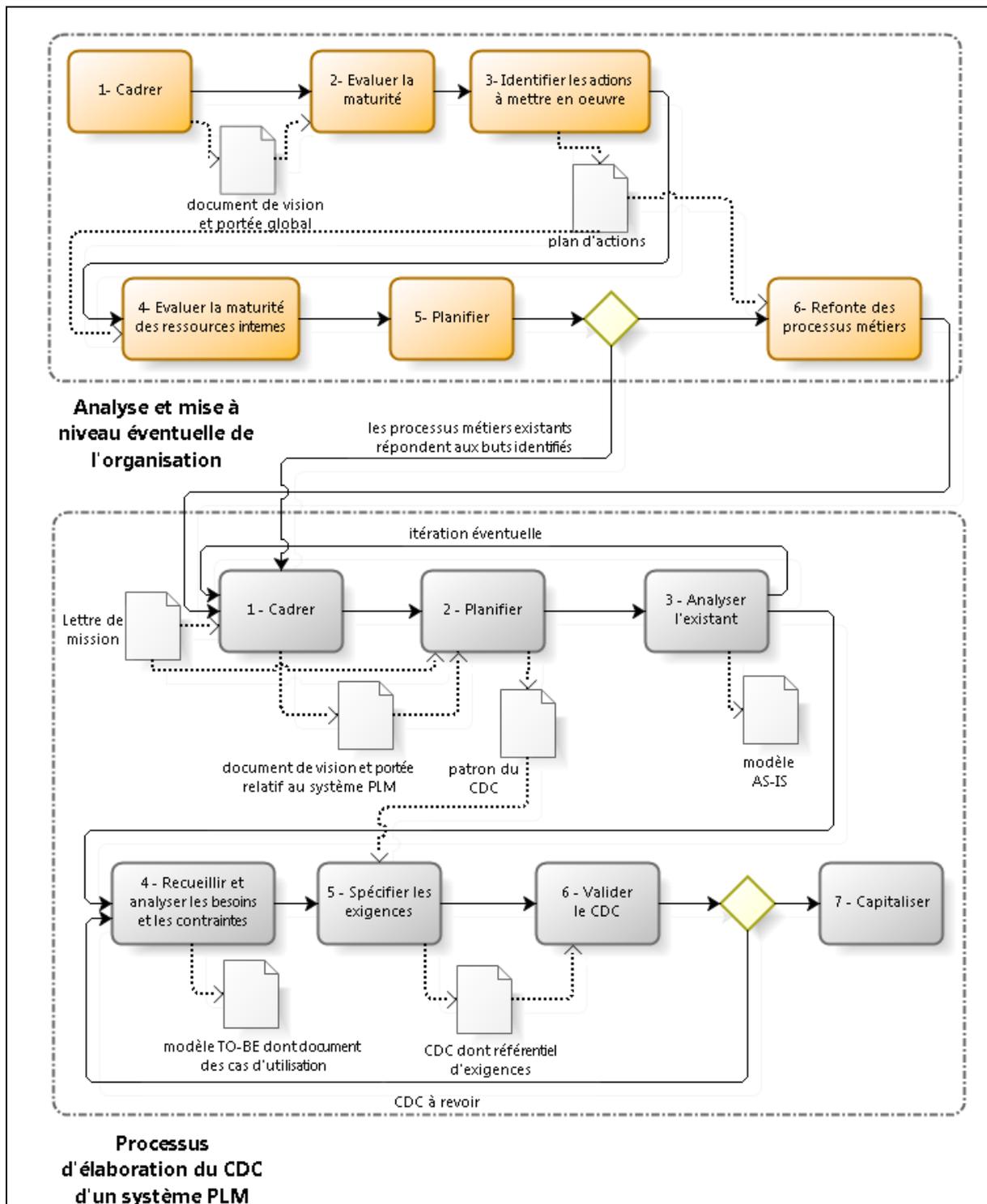


Figure 3.6 : Processus d'élaboration du CDC d'un système PLM et étapes préliminaires

Utilité des modèles de maturité et des définitions génériques des besoins

Ce paragraphe a pour objectif d'identifier comment des modèles de maturité et des définitions génériques peuvent supporter les tâches dans le processus que nous venons de définir (cf. figure 3.7).

Dans un premier temps, des *modèles de maturité* peuvent être utilisés pour évaluer la maturité. L'utilisation de ces modèles supporte :

- une évaluation de l'état existant en prenant en compte toutes les perceptions de l'entreprise pertinentes,

- une évaluation de la cohérence de maturité entre les différentes perceptions de l'entreprise (organisationnelle et informationnelle par exemple),
- une identification des actions à mettre en œuvre pour monter en maturité.

Dans notre processus, nous introduisons ainsi deux questionnaires basés sur des modèles de maturité (cf. figure 3.7). Ces questionnaires supportent l'évaluation de la maturité organisationnelle et l'évaluation de la maturité des ressources internes.

Les **définitions génériques des besoins** peuvent être utilisées comme support de l'élaboration du CDC. Elles identifient des buts, des besoins et des contraintes génériques relatifs à un système PLM. Par rapport aux difficultés identifiées pour définir les exigences, ils sont utilisables (cf. figure 3.7) pour :

- faciliter la définition du périmètre des besoins à prendre en compte ;
- faciliter la dérivation des buts en exigences ;
- faciliter la formalisation des exigences par l'utilisation, dans la mesure du possible, d'un vocabulaire neutre (c.-à-d. non spécifique à l'entreprise) facilitant les échanges entre les différentes parties prenantes (divers services de l'entreprise, maître d'œuvre, maître d'ouvrage, rédacteur du cahier des charges) ;
- faciliter la vérification de l'exhaustivité des analyses (utilisation des listes comme « check-list »).

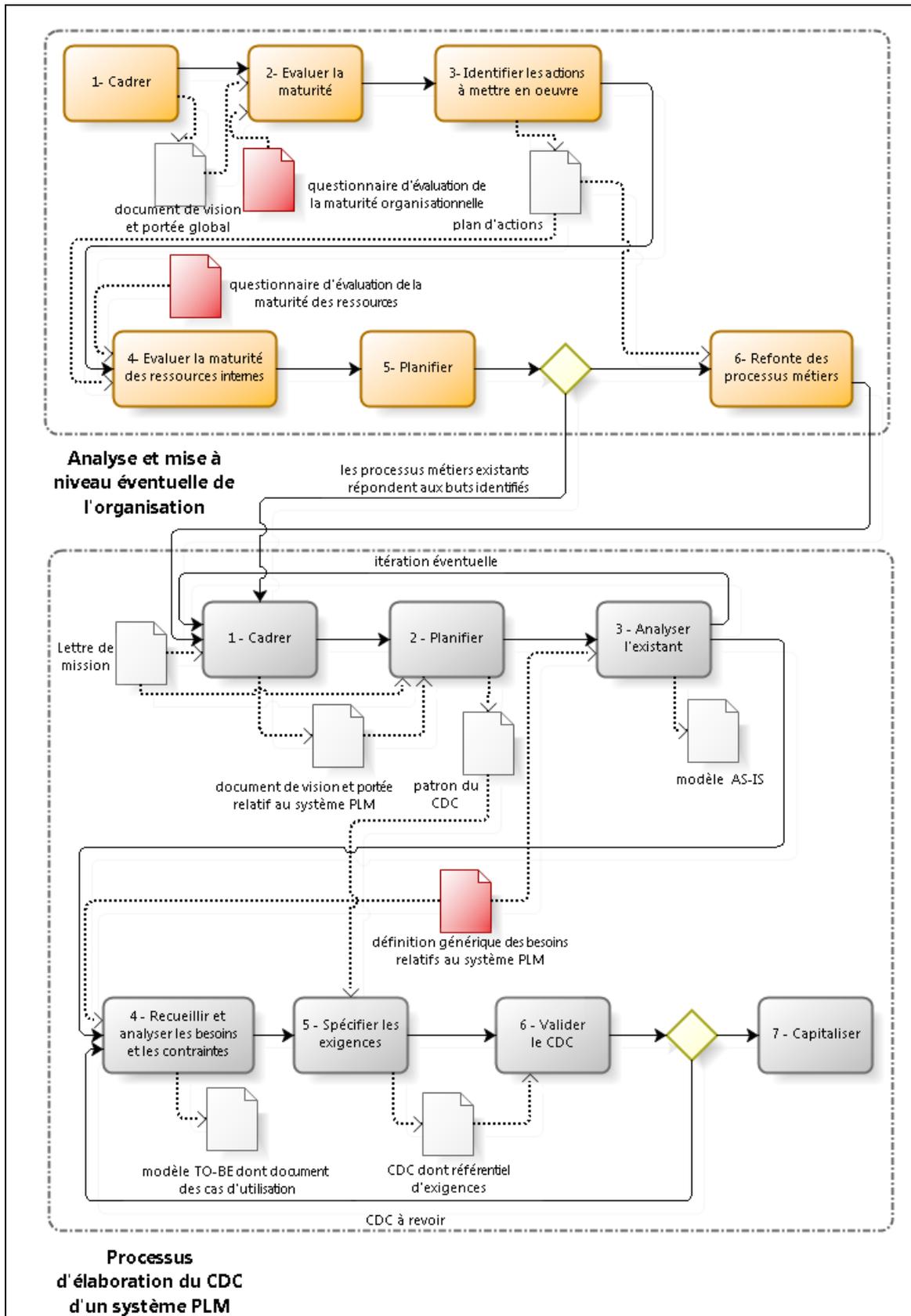


Figure 3.7 : Positionnement des questionnaires d'évaluation de la maturité et du modèle générique des besoins relatifs au système PLM dans le processus défini

5.3 Conclusion

Ce dernier paragraphe propose une adaptation du processus défini dans le paragraphe 2. Cette adaptation intègre des étapes préliminaires d'analyse et de mise à niveau éventuelle de l'organisation. Elle intègre également l'utilisation de modèles de maturité et de définitions génériques des besoins du système PLM.

6 Conclusion

Tout au long de ce chapitre, la nécessité de déterminer une méthode spécifiquement adaptée à l'identification et la formalisation des buts, des besoins et des contraintes relatifs au système PLM a été mise en lumière.

Tout d'abord (§ 2), un état de l'art a montré qu'il est nécessaire de considérer trois modèles du futur système : le document de vision et portée, le document des cas d'utilisation et le référentiel d'exigences. Le cahier des charges a été défini comme un document qui inclut le référentiel d'exigences et différentes autres informations. A la fin du paragraphe, un processus existant pour l'élaboration du cahier des charges d'un système d'information a été choisi.

Puis (§ 3), les spécificités du système PLM ont été rappelées : spécificités relatives au système existant et aux logiciels PLM du marché. Des retours d'expérience de chefs de projet PLM industriels ont alors permis de confirmer la réalité des spécificités identifiées et d'identifier des difficultés propres à l'élaboration du CDC du système PLM. Parmi ces difficultés, la corrélation entre le niveau de maturité de l'entreprise et l'approche adoptée a été mise en lumière. Ainsi, le paragraphe 4 a été consacré à identifier les caractéristiques permettant d'évaluer la maturité « PLM » d'une entreprise.

Dans le dernier paragraphe (§ 5), une adaptation du processus choisi dans le paragraphe 2 a été proposée. Cette adaptation prend en compte la perception informationnelle de l'entreprise mais également sa perception organisationnelle. En effet, elle intègre des étapes préliminaires d'analyse et de mise à niveau éventuelle de l'organisation. Elle intègre également l'utilisation de modèles de maturité et des définitions génériques des besoins PLM.

L'hypothèse formulée dans le chapitre 1 est qu'il faut une méthode de mise en œuvre du système PLM en adéquation avec les deux buts de l'entreprise. Une méthode d'analyse d'un système d'information identifie un ensemble de modèles et un processus appropriés pour l'analyse souhaitée [Rieu, 99]. Le chapitre 2 a montré l'intérêt de proposer un cadre de modélisation adapté pour la mise en œuvre du système PLM. L'objectif du chapitre suivant consiste ainsi à définir de manière détaillée le cadre proposé dans le chapitre 2 et à le positionner par rapport aux travaux PLM existants.

Chapitre 4

Cadre de modélisation pour la mise en œuvre du système PLM

Sommaire

- 1 Introduction
 - 2 Mise en œuvre du système PLM : formulation de deux problèmes
 - 3 Etat de l'art : les méthodes de mise en œuvre du système PLM
 - 4 Notre proposition : un cadre de modélisation basé sur le paradigme d'ambivalence
 - 5 Cadre de modélisation et mise en œuvre du système PLM (Problème 1)
 - 6 Cadre de modélisation et cohérence entre les différentes représentations du produit (Problème 2)
 - 7 Conclusion
-

1 Introduction

Le problème identifié dans le chapitre 1 est : comment réaliser une construction du système PLM en adéquation avec les deux buts de l'entreprise qui sont d'avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût en maintenant, voire développant, l'agilité opérationnelle ? L'hypothèse formulée dans le chapitre 2 était que l'utilisation d'un cadre de modélisation adapté constitue un élément de réponse à ce problème. Les objectifs de ce chapitre sont de détailler et d'illustrer l'utilisation du cadre introduit dans le chapitre 2 (cf. chapitre 2, § 4.3).

Tout d'abord (§ 2), au regard de la solution proposée, le problème initial est précisé au moyen de la formulation précise de deux problèmes. Ensuite, un état de l'art des méthodes supportant la construction du système PLM est présenté (§ 3). Cet état de l'art nous permet de positionner nos choix en regard des travaux existants et d'identifier les mécanismes proposés pour assurer les interfaces entre le système PLM et d'autres systèmes. La construction du cadre de modélisation est alors détaillée (§ 4). Enfin, l'utilisation du cadre proposé pour répondre aux problèmes identifiés est illustrée (§ 5 et 6).

2 Mise en œuvre du système PLM : formulation de deux problèmes

Dans le chapitre 1, le problème formulé est : comment construire le système PLM ? Dans le chapitre 2, trois axes d'étude sont identifiés : un axe représentant le cycle de vie du système PLM, un axe représentant le cycle de vie du produit et un axe représentant les niveaux de modélisation.

Selon l'axe du cycle de vie du système PLM, les problèmes que nous identifions sont : comment aider l'entreprise à construire successivement ses modèles d'exigences PLM, métier PLM et plate-forme du système PLM ? Comment passer du modèle d'exigences PLM au modèle personnalisé du logiciel PLM ?

Selon l'axe du cycle de vie du produit, le problème que nous pouvons identifier est : comment assurer la cohérence et la traçabilité entre les différentes représentations du produit utilisées tout au long de sa vie et comment supporter l'analyse d'impact d'une modification ?

Ce paragraphe a pour objectif de préciser et reformuler ces problèmes au vu du cadre proposé dans le chapitre 2.

En résumé, le métamodèle de personnalisation d'un logiciel PLM définit le langage pour construire le modèle personnalisé du logiciel PLM de l'entreprise. A partir de ce modèle personnalisé, l'entreprise construit les représentations de ses produits (cf. figure 4.2).

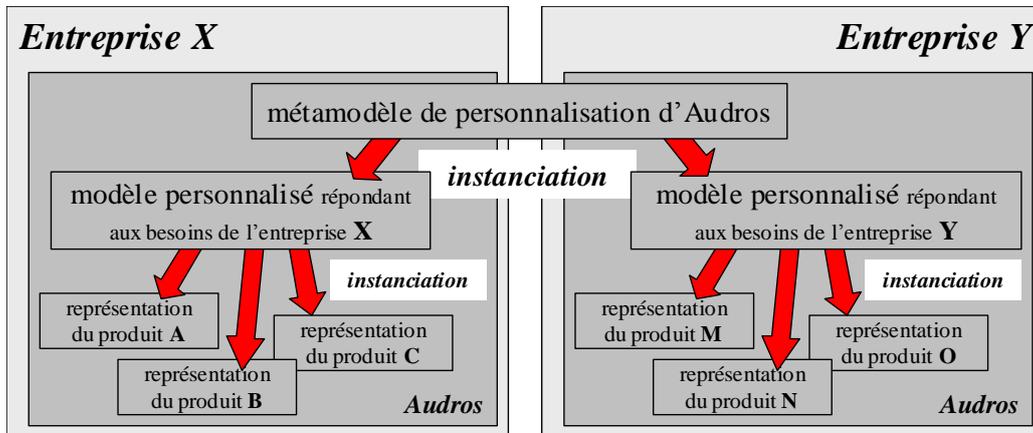


Figure 4.2 : Logiciel PLM : relations d'instanciation entre métamodèle, modèle et représentation du produit

Nous pouvons ainsi compléter les questions identifiées précédemment avec la question suivante : comment créer le modèle personnalisé à partir du métamodèle de personnalisation du logiciel PLM ?

2.1.2 Les modèles utilisés pour passer de l'expression du besoin à la personnalisation du logiciel PLM

Dans le paragraphe 4.3 du chapitre 2, trois modèles ont été définis : le modèle d'exigences PLM, le modèle métier PLM et le modèle plate-forme PLM.

Le chapitre précédent a montré que le modèle d'exigences du système PLM est composé d'exigences fonctionnelles et non fonctionnelles d'une part et d'exigences de modélisation et d'exigences d'instanciation d'autre part (cf. chapitre 3, § 3.1.2). Notre étude se focalise sur la personnalisation du logiciel PLM et sur la cohérence entre les différentes représentations du produit. Ainsi, la suite de nos travaux concerne la mise en œuvre des exigences fonctionnelles de modélisation et la mise en œuvre des exigences d'interface entre le système PLM et d'autres systèmes. Les travaux réalisés se concentrent sur le modèle personnalisé d'un logiciel PLM, ils n'appréhendent donc pas le modèle plate-forme PLM dans sa globalité.

Le modèle d'exigences PLM et le modèle métier PLM sont centrés sur la perception organisationnelle de l'entreprise. Le modèle personnalisé du logiciel PLM est centré sur la perception informationnelle. Cependant, les deux perceptions sont présentes dans chacun des trois modèles. Dans le modèle d'exigences PLM, des contraintes techniques peuvent être définies (perception informationnelle). Dans le modèle métier PLM, les différentes informations sont structurées (perception informationnelle). Dans le modèle personnalisé du logiciel PLM, la structure de l'organisation de l'entreprise est transcrite (perception organisationnelle).

2.1.3 Etude des usages dans le Club PLM

La figure 4.3 illustre les trois modèles identifiés ci-dessus pour le système PLM et les transformations permettant de passer d'un modèle à l'autre.

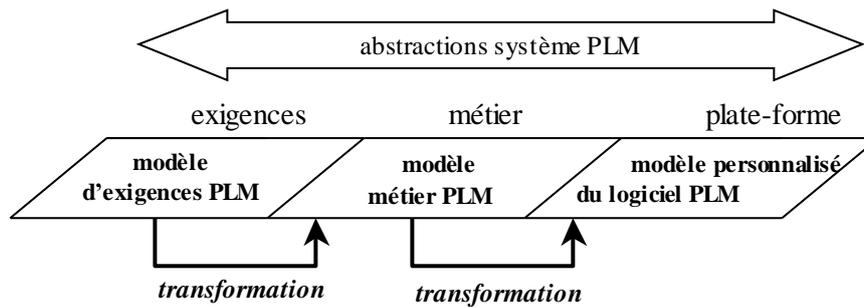


Figure 4.3 : Modèle d'exigences PLM, modèle métier PLM et modèle personnalisé du logiciel PLM

En confrontant les expériences des membres du Club PLM, nous faisons plusieurs constats concernant la construction de ces trois modèles.

Le premier constat est que les exigences sont bien formalisées dans un cahier des charges (cf. chapitre 3). Le second constat est que le modèle métier PLM est construit après la sélection du logiciel PLM. Ainsi, ce modèle n'intervient pas dans le choix du logiciel PLM et la modélisation est souvent influencée par les caractéristiques (structure de données, modes opératoires, etc.) du logiciel choisi. De ce fait, l'entreprise n'introduit pas dans le modèle métier PLM des spécificités qui ne peuvent pas être mises en œuvre dans le logiciel PLM sélectionné. Le troisième constat est que le modèle métier PLM et le modèle personnalisé du logiciel PLM sont construits par l'intégrateur. Tout d'abord, l'intégrateur construit le modèle métier PLM. Puis, il implémente, souvent manuellement, le modèle métier PLM obtenu dans le logiciel PLM en construisant le modèle personnalisé. Enfin, les derniers constats sont que, la plupart du temps, le modèle métier PLM est considéré comme un support de communication durant le projet PLM et il est construit à partir d'outils bureautiques. Ainsi, une fois le logiciel PLM mis en production, la mise à jour de ce modèle est considérée comme une tâche sans valeur ajoutée. Les modifications sont ainsi réalisées directement dans le logiciel PLM (modification du modèle personnalisé du logiciel PLM) sans qu'il n'y ait aucune formalisation des évolutions sur le modèle d'exigences PLM et le modèle métier PLM initiaux.

A partir de ces constats, plusieurs besoins relatifs à la construction du modèle métier PLM et du modèle personnalisé du logiciel PLM dans un contexte industriel sont identifiables. Le modèle métier PLM doit être construit en amont afin d'aider l'entreprise à choisir une solution logicielle capable de prendre en compte l'ensemble de ses spécificités. En cas de changement de plate-forme, le modèle métier PLM construit reste valide et donc réutilisable. Il est également nécessaire de formaliser les règles de transformation du modèle métier PLM en modèle personnalisé du logiciel PLM pour les trois raisons suivantes :

- la formalisation permet une capitalisation du savoir-faire pour l'intégrateur, l'administrateur du logiciel (famille des techniciens de l'information) et le responsable métier,
- l'instrumentation des transformations dans un environnement de modélisation rend le modèle métier PLM productif,
- la mise à jour du modèle métier PLM et du modèle personnalisé du logiciel PLM est facilitée en cas d'évolutions.

2.1.4 Reformulation des problèmes posés

Suite aux niveaux de modélisation concernant la mise en œuvre du logiciel PLM et aux besoins identifiés, nous pouvons reformuler les problèmes définis en introduction de ce paragraphe ainsi : Comment assurer la cohérence entre le modèle d'exigences PLM, le modèle métier PLM et le modèle personnalisé du logiciel PLM qui représentent une même réalité : l'entreprise ? Comment créer le modèle personnalisé du logiciel PLM à partir du métamodèle du logiciel PLM choisi ?

2.2 Formulation d'un second problème : cohérence et traçabilité entre les différentes représentations du produit

Dans le chapitre 2, nous avons identifié que plusieurs représentations du produit sont utilisées tout au long de sa vie : représentation fonctionnelle, représentation du produit conçu, représentation du produit pour fabrication, ... Chaque représentation correspond à une étape de la vie du produit. Chaque étape a des objectifs, un savoir-faire métier bien spécifique et un univers de discours en relation avec ce métier. Par exemple, l'univers de discours de l'étape de conception du produit est composé de plans de conception, de nomenclatures de conception, de normes, etc. L'univers de discours de l'étape de conception du procédé est composé de nomenclatures pour fabrication, de gammes de fabrication, d'outillages, ...

Pour illustrer ces différences d'univers de discours, prenons l'exemple de la description de la structure d'un produit pour l'étape de conception du produit et pour l'étape de conception du procédé associé. Le bureau d'études construit la nomenclature du produit, désignée « nomenclature du produit conçu ». Cette structure arborescente, composée d'*ensemble*, de *sous-ensembles fonctionnels* et de *composants*, est créée suivant une logique fonctionnelle. Elle doit répondre aux besoins et aux contraintes exprimées par le cahier des charges du produit. Les liens structurels expriment une relation « *est composé de* ». Les éléments et les liens composant cette structure sont propres à l'univers de discours du métier de conception du produit. A partir de cette nomenclature, le bureau des méthodes construit une autre nomenclature, désignée « nomenclature pour fabrication ». Cette structure définit, en fonction de la politique de gestion d'achat, de fabrication, des flux et des stocks de l'entreprise, des *éléments achetés* et des *éléments fabriqués*. Les liens structurels expriment une relation « *est fabriqué à partir de* ». Les éléments et les liens composant cette structure sont propres à l'univers de discours du métier de réalisation du produit.

Nous pouvons constater que, si les deux représentations sont des structures arborescentes, les nœuds et les liens ne représentent pas les mêmes concepts. De plus, les structures des deux nomenclatures sont différentes. En effet, des sous-ensembles fonctionnels de la nomenclature de conception peuvent « disparaître » dans la nomenclature pour fabrication. Des sous-ensembles de fabrication, des *bruts*³⁶ et des éléments en état intermédiaire de fabrication peuvent également « apparaître » dans la nomenclature de fabrication.

Comme l'illustre la figure 4.4, de véritables déformations des exigences décrivant le produit peuvent émerger tout au long des étapes de sa vie.



Figure 4.4 : Exemple de déformations des exigences décrivant un produit

La figure 4.5 nous permet d'illustrer ces déformations.

Le premier type de déformation est lié à la relation « Fournisseur-Client » qui existe entre deux étapes. Dans ce cas, nous considérons que la représentation transmise définit les exigences à satisfaire par le « Client ». La représentation du produit, fournie par le « Fournisseur », est construite à partir d'un modèle propre à son univers de discours. Si le « Fournisseur » et le « Client » appartiennent à deux étapes différentes, ils ont chacun un modèle qui est propre à l'univers de discours du métier de leur étape. Par conséquent, avant toute transformation, le « Client » doit traduire la représentation fournie dans son propre univers de discours. Nous pouvons ainsi identifier un premier problème (illustré à

³⁶ Élément acheté pour réaliser un composant ou produit.

gauche dans la figure 4.5) : Comment s'assurer que les exigences définies par la représentation construite dans une étape (une étape correspondant à un univers de discours propre à un métier : conception, fabrication, etc.) sont bien comprises par l'étape suivante dont le métier est différent ?

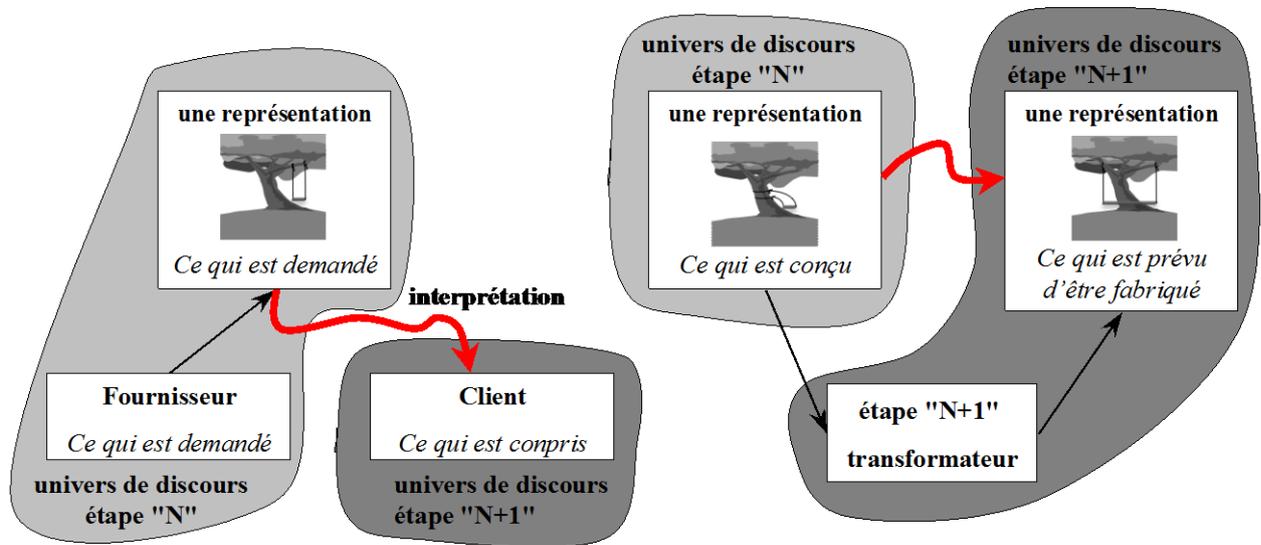


Figure 4.5 : Les déformations dues à des erreurs d'interprétation et à des erreurs de transformation

Le deuxième type de déformation est lié à la « transformation » d'une représentation. Une fois, qu'une étape du cycle de vie a construit sa représentation du produit, il faut s'assurer que les nouvelles exigences exprimées par cette dernière sont cohérentes avec celles exprimées dans la représentation fournie par l'étape précédente. Nous pouvons ainsi identifier un second problème (illustré à droite dans la figure 4.5) : Comment s'assurer que de nouvelles exigences définies par une représentation du produit propre à un métier ne sont pas en conflit avec les exigences définies par une représentation du produit fournie par un autre métier ?

Enfin, un troisième type de déformation identifiable est celui dû à une mauvaise analyse d'impact des évolutions d'une représentation. En effet, si une modification est réalisée sur la représentation de conception suite à une évolution du besoin par exemple, il est nécessaire de pouvoir mettre en œuvre les modifications nécessaires sur les autres représentations (pour fabrication, pour maintenance..) afin que le produit fabriqué corresponde au nouveau besoin identifié et que la maintenance puisse toujours être réalisée.

Ainsi, la traçabilité peut être définie comme la capacité à retrouver à tout instant toutes les représentations pertinentes d'un produit et ceci quelles que soient les évolutions.

La question initiale est donc complétée par les deux questions suivantes : Comment s'assurer que les exigences définies par la représentation construite dans une étape (une étape correspondant à un univers de discours propre à un métier : conception, fabrication, etc.) sont bien comprises par l'étape suivante dont le métier est différent ? Comment s'assurer que de nouvelles exigences définies par une représentation du produit propre à un métier ne sont pas en conflit avec les exigences définies par une représentation du produit fournie par un autre métier ?

2.3 Conclusion

Ces deux paragraphes ont permis d'identifier clairement deux problèmes selon l'axe d'étude du cycle de vie du produit du cadre proposé dans le chapitre 2. Afin de positionner le choix d'un cadre de modélisation (et des modèles constituant ce cadre) au regard des travaux existants, un état de l'art des méthodes définies pour la mise en œuvre du système PLM est réalisé dans le paragraphe suivant. Cet état de l'art nous permet également d'identifier les mécanismes proposés pour assurer les interfaces entre le système PLM et d'autres systèmes.

3 Etat de l'art : les méthodes de mise en œuvre du système PLM

Le chapitre 3 a défini différents types d'exigences. Au regard des problèmes formulés, la suite de ces travaux est focalisée sur la satisfaction des exigences fonctionnelles relatives à la personnalisation et sur la satisfaction des exigences d'interface.

Dans un premier temps, ce paragraphe fournit un état de l'art des méthodes existant pour définir les modèles du système PLM (cf. § 3.1). Il décrit ensuite les différents mécanismes proposés pour assurer les interfaces entre le système PLM et d'autres systèmes (cf. § 3.2).

A l'issue de ces deux états de l'art, nous examinons les réponses apportées aux deux problèmes définis.

3.1 Méthodes pour définir les modèles du système PLM

3.1.1 Les différentes méthodes

Dans les chapitres précédents, il a été défini qu'une méthode d'analyse d'un système d'information identifie un ensemble de modèles et un processus adaptés pour l'analyse souhaitée. Le chapitre 2 a également montré que la mise en œuvre du système PLM implique d'analyser de manière conjointe le système PLM et le produit. En effet, les représentations du produit sont créées à partir du modèle personnalisé dans le logiciel PLM.

Dans certains travaux, les *processus* décrivant la mise en œuvre d'un système PLM sont très macroscopiques [Gzara, 00] [Bissay, 10] [Le Duigou, 10]. Il existe toutefois une proposition qui décrit de manière détaillée les différentes étapes en incluant des étapes de rétro-ingénierie pour prendre en compte la situation existante [Bacha, 02].

Les travaux portant sur les *modèles* peuvent être regroupés en deux catégories.

Les travaux de la première catégorie illustrent comment créer les modèles à partir de langages et de cadres de modélisation existants. Parmi les travaux utilisant des langages existants, certains travaux proposent une ingénierie des PDM basée sur différents diagrammes du langage UML (Unified Modeling Language) [Eynard et al., 06] [Merlo et al., 05]. Il existe également une méthode globale d'implémentation d'un système PLM dans une PME/PMI avec des modèles construits à partir du diagramme de classes du langage UML et du langage BPMN (Business Process Model and Notation) [Bissay, 10]. Parmi les travaux utilisant un cadre de modélisation, une proposition définit une ingénierie des Systèmes d'Informations Techniques en s'appuyant sur le cadre de Zachman [Zachman, 08] [Bacha, 02].

Les travaux de la seconde catégorie proposent des modèles génériques pour supporter la construction du modèle d'exigences PLM et du modèle métier PLM. L'utilisation de ces modèles génériques a deux intérêts majeurs. Ils permettent au modélisateur de ne pas partir d'une feuille blanche et d'identifier dès le départ des pôles d'observation liés au problème. L'entreprise utilise ainsi ces modèles génériques comme base de travail pour construire un modèle (d'exigences PLM ou métier PLM) en adéquation avec ses besoins spécifiques. L'utilisation de modèles génériques facilite également les comparaisons de modèles et la réutilisation de « parties » de modèles. Le chapitre précédent a présenté trois modèles génériques relatifs à la formalisation des buts, des besoins et des contraintes [OMG, 11] [Zancul, 09] [Le Duigou, 10]. Concernant les modèles métiers, des travaux identifient les éléments usuels nécessaires à la définition du modèle métier PLM [Gzara, 00] [Le Duigou, 10] (cf. figure 4.6). Une des propositions construit le modèle métier PLM de l'entreprise à partir de patrons [Gzara, 00]. L'utilisation de patrons permet de créer le modèle métier PLM de l'entreprise de manière modulaire en fonction de besoins clairement identifiés. L'autre proposition s'appuie sur un modèle générique pour formaliser les échanges dans le cadre de l'entreprise étendue. Ce modèle est construit à partir des axes d'étude du cadre de modélisation d'entreprise GERAM [IFIP-IFAC Task Force, 99]. Ces deux modèles ne traitent que de l'aspect structurel (c'est-à-dire « donnée ») du système PLM avec des éléments génériques : « articles », « document », « relation de nomenclature », etc.

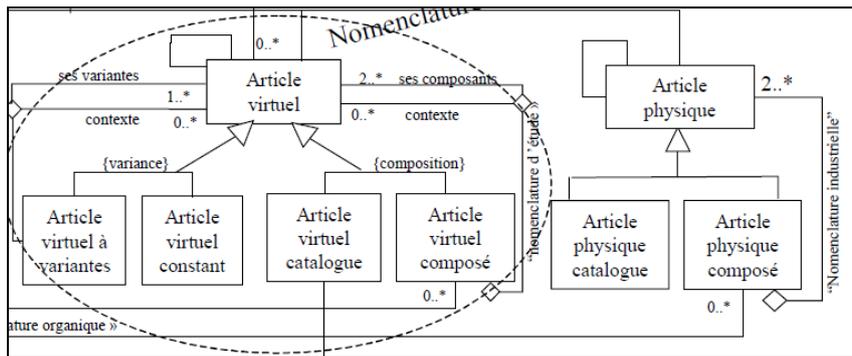


Figure 4.6 : Extrait du modèle générique métier du Système d'Information Produit de [Gzara, 00]

Les processus traduisant les aspects « fonctionnel » et « comportemental » sont spécifiques à l'entreprise ; il n'existe malheureusement pas de travaux proposant un modèle générique.

Les modèles génériques proposés sont construits à partir du diagramme de classes du langage UML mais aucun de ces modèles ne définit les opérations des classes d'objets. Cependant, certaines opérations sont définies dans un modèle illustrant un exemple de mise en œuvre d'un système PLM [Eynard et al., 06]. Il serait ainsi intéressant d'intégrer les opérations définies dans ces travaux dans un modèle générique PLM. Pour cela, il serait nécessaire de faire une analyse destinée à vérifier l'aspect générique des opérations définies.

Cet état de l'art a permis d'identifier différentes méthodes utilisables pour créer les modèles nécessaires à la définition et à la mise en œuvre des exigences fonctionnelles d'un système PLM. Ces méthodes proposent des solutions pour construire le modèle d'exigences PLM et le modèle métier PLM mais, force est de constater qu'aucune méthode ne fournit des éléments pour construire le modèle personnalisé du logiciel PLM.

3.1.2 L'utilisation d'un unique modèle pour créer les différentes représentations du produit dans le système PLM

Plusieurs représentations sont utilisées tout au long de la vie du produit. Les représentations du produit conçu, du produit pour fabrication, etc., définissent le produit théorique à différents états de maturité. Les représentations du produit comme fabriqué, comme maintenu, etc., enregistrent les propriétés contrôlées sur les produits physiques.

[Gzara, 00] et [Le Duigou, 10] proposent de créer les différentes représentations du produit à partir d'un unique modèle (cf. paragraphe 4.2.1 du chapitre 2) dans le système PLM. L'utilisation d'un unique modèle est illustrée dans la figure 4.7. A droite, le modèle contient et structure tous les éléments qui sont utilisés par les différents acteurs. A gauche, chaque acteur a accès à une partie de ces éléments, appelée projection du modèle. Certains des éléments peuvent être communs à plusieurs projections. Dans les travaux existants, les entités « fonction » [Gzara, 00] [Le Duigou, 10], « article » [Gzara, 00] [Le Duigou, 10] et « article physique » [Gzara, 00] permettent de construire les différentes représentations du produit.

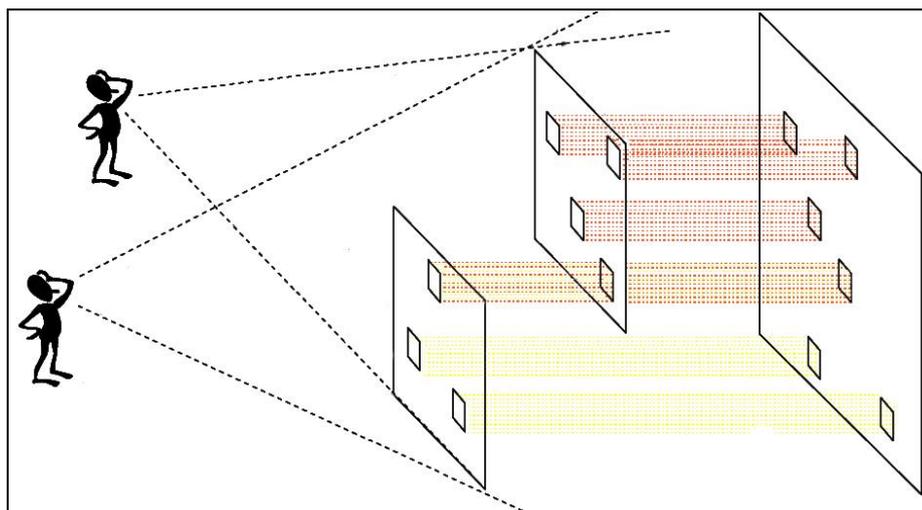


Figure 4.7 : Utilisation d'un unique modèle (adapté de [Lombard, 06])

3.2 Les mécanismes assurant les interfaces entre le système PLM et d'autres systèmes

Les travaux proposant des mécanismes d'interface pour assurer le transfert de représentations du produit entre le système PLM et d'autres systèmes peuvent être classés en trois catégories.

Dans la première catégorie, les travaux définissent un mécanisme basé sur l'utilisation de modèles standards [ISO, 10] ou de modèles unifiés [Paviot, 10]. Ce mécanisme consiste à faire des exportations et des importations de représentations structurées conformément au modèle standard ou unifié. Les modèles standards sont indépendants des modèles utilisés par les logiciels pour créer les représentations. Pour exporter des représentations du produit sous un format standard, les logiciels doivent ainsi être capables de transformer leurs représentations du produit afin de les rendre conformes au modèle du standard souhaité. Elles doivent également être capables d'interpréter correctement des représentations importées dans ce format. Par exemple, le logiciel de CAO CATIA a son propre modèle pour construire les représentations du produit, il est également capable d'exporter et d'importer des représentations construites conformément à différents modèles standards (IGES, STEP, etc.).

La norme STEP (norme pour l'échange de données de produit - **S**Tandards for the **E**xchange of **P**roduct data), ou norme ISO 10303, fournit des modèles pour décrire les données du produit sur l'ensemble du cycle de vie. Cette norme est constituée de plusieurs protocoles d'application. Chaque protocole d'application propose des modèles correspondant à un problème donné dans un contexte donné. Par exemple, le protocole d'application 10303-214 fournit des modèles pour supporter l'échange des données de base des produits pour la construction automobile. Les donneurs d'ordre aéronautiques et les donneurs d'ordre automobiles sont des acteurs principaux de la construction de ces standards. En effet, il est indispensable de disposer de standards communs pour supporter les échanges entre les nombreux partenaires de l'entreprise étendue (cf. § 3.5 du chapitre 1). Les standards constituent également un moyen de pérenniser l'information. En effet, les modèles standards ont une durée de vie plus importante que les modèles utilisés par les logiciels pour manipuler les représentations du produit. Ainsi, la transcription des représentations du produit dans un format standard est un moyen de pérenniser les informations relatives au produit. Plusieurs travaux relatifs à l'approche PLM font référence à la norme STEP : norme 10303-239 [Paviot, 10], correspondances entre les éléments d'un modèle générique et ceux de la norme 10303-214 [Le Duigou, 10], définition de services à partir de la norme 10303-214 [OMG, 11].

Il existe également des travaux qui définissent un mécanisme, similaire à celui utilisant les modèles standards, pour réaliser des échanges entre les logiciels CAO et les logiciels PLM. Ce mécanisme s'appuie sur un modèle unifié (cf. paragraphe 4.2.1 du chapitre 2). Ce modèle identifie les concepts partagés par les modèles des différents logiciels concernés (logiciels CAO, par exemple). Les

représentations des produits construites avec ce modèle unifié peuvent être manipulées par les différents logiciels concernés.

Dans la deuxième catégorie de travaux, le mécanisme est similaire au précédent mais il est mis en œuvre à l'aide d'un logiciel unificateur qui gère les interactions entre les différents logiciels métiers via des connecteurs (cf. figure 4.8). Toute représentation échangée par ce mécanisme est restructurée conformément au modèle unifié du logiciel unificateur. [Noël et al., 08] et [Demoly, 10] proposent ainsi des logiciels utilisables dans le cadre du développement collaboratif de nouveaux produits. Le modèle unifié peut décrire le produit et la gestion de projet [Noël et al., 08] ou le produit et son procédé de fabrication [Demoly, 10].

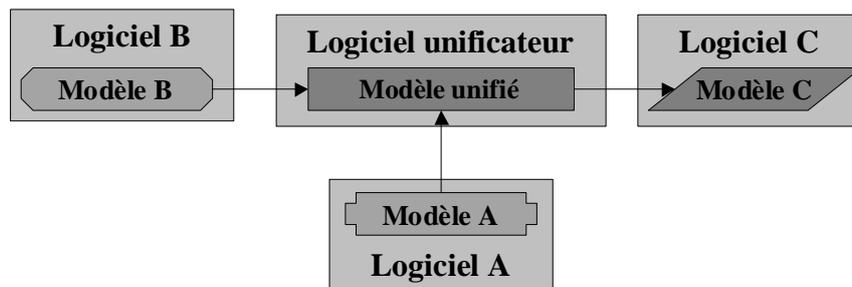


Figure 4.8 : Utilisation d'un logiciel unificateur

Dans la troisième catégorie de travaux, le mécanisme d'interface entre trois logiciels A, B et C est réalisé en déclarant des relations de correspondances entre les modèles des trois logiciels. Ce mécanisme est mis en œuvre au moyen d'un environnement de modélisation fédérateur basé sur l'ingénierie dirigée par les modèles (cf. figure 4.9) [Iraqi-Houssaini et al., 11]. Il s'agit dans ce cas d'identifier les méthodes pour formaliser les modèles et les règles de transformations entre ces modèles. Les modèles formalisés dans l'environnement de modélisation correspondent au modèle d'un logiciel ou au modèle d'un standard (norme STEP, par exemple). L'utilisation de ce mécanisme a pour objectif de définir des transformations proches du besoin sans avoir à utiliser un modèle intermédiaire. Cette formalisation des modèles et des règles de transformation permet également de capitaliser le savoir-faire relatif au système d'information.

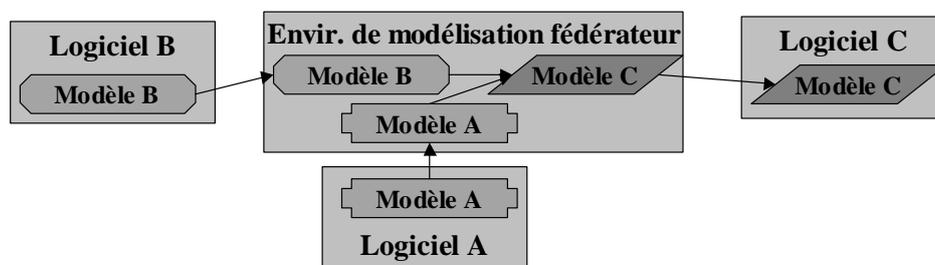


Figure 4.9 : Utilisation d'une architecture fédérée

3.3 Conclusions vis à vis du problème concernant les modèles utilisés pour la mise en œuvre du système PLM

Le problème formulé dans le paragraphe 2.1.4 était le suivant : comment aider l'entreprise à construire successivement le modèle d'exigences PLM, le modèle métier PLM et le modèle personnalisé du logiciel PLM ? Comment assurer la cohérence entre ces différents modèles qui représentent une même réalité : l'entreprise ? Comment créer le modèle personnalisé du logiciel PLM à partir du métamodèle de personnalisation du logiciel PLM choisi ?

Les méthodes proposées pour la mise en œuvre du système PLM prennent bien en compte les trois modèles : modèle d'exigences PLM, modèle métier PLM et modèle personnalisé du logiciel PLM. Certains travaux proposent de construire le modèle d'exigences PLM et le modèle métier PLM à partir d'un modèle générique, d'autres s'appuient sur l'utilisation de langages et de cadres de modélisation

issus de l'ingénierie des systèmes d'information ou de la modélisation d'entreprise. Aucune de ces méthodes n'est explicite quant à la construction du modèle personnalisé du logiciel PLM. Concernant la cohérence entre les modèles, seule l'utilisation du cadre de modélisation Zachman [Bacha, 02] définit des mécanismes de cohérence. Cependant, les mécanismes de construction du modèle personnalisé du logiciel PLM à partir du métamodèle de personnalisation du logiciel PLM et du modèle métier PLM ne sont pas pris en compte par ce cadre.

3.4 Conclusions vis à vis du problème de cohérence et de traçabilité entre les différentes représentations du produit

Le problème formulé dans le paragraphe 2.2 était le suivant : comment assurer la cohérence et la traçabilité entre les différentes représentations du produit utilisées tout au long de sa vie ? Comment s'assurer que les exigences définies par la représentation construite dans une étape (une étape correspondant à un univers de discours propre à un métier : conception, fabrication, etc.) sont bien comprises par l'étape suivante dont le métier est différent ? Comment s'assurer que de nouvelles exigences définies par une représentation du produit propre à un métier ne sont pas en conflit avec les exigences définies par une représentation du produit fournie par un autre métier ?

Les différentes représentations du produit peuvent être construites à partir d'un unique modèle [Gzara, 00] [Le Duigou, 10]. Dans le cadre de l'entreprise étendue, les représentations du produit sont dispersées entre les organisations. Par exemple, le donneur d'ordre crée la « nomenclature comme conçue » et le sous-traitant crée la « nomenclature pour fabrication » et conserve les « nomenclatures fabriquées » décrivant les produits physiques. Dans ce contexte, il est difficile de contraindre tous les acteurs travaillant sur un produit à adopter un unique modèle pour construire toutes les représentations. De plus, chaque représentation correspond à un métier de l'entreprise (conception, fabrication...). L'utilisation d'un unique modèle présente alors deux inconvénients. Le premier est qu'il contraint des métiers très différents à intégrer leurs spécificités dans un « moule commun ». Le deuxième est que l'utilisation d'éléments communs peut être source d'ambiguïtés. Par exemple, utiliser le même terme « article » pour désigner un organe du produit conçu, un brut de fabrication et un organe à maintenir (pièce de rechange) peut amener des confusions.

Les travaux concernant les interfaces permettent d'identifier des mécanismes capables d'assurer la cohérence et la traçabilité entre des représentations du produit construites à partir de modèles différents [ISO, 10] [Paviot, 10] [Noël et al., 08] [Demoly, 10] [Iraqi-Houssaini et al., 11]. Ces mécanismes s'appuient sur la formalisation de règles de transformations entre les modèles. Ces transformations se font directement entre deux représentations [Iraqi-Houssaini et al., 11] ou indirectement en utilisant une transformation basée sur l'utilisation d'un modèle standard ou unifié [ISO, 10] [Paviot, 10] [Noël et al., 08] [Demoly, 10]. Toutefois, il est à noter que les mécanismes proposés par ces travaux sont illustrés par des exemples dans lesquels les structures des produits ne sont pas modifiées. Il n'y a pas d'ajout ou de retrait de sous-ensembles ou de composants comme dans le problème que nous avons identifié précédemment (cf. § 2.2).

3.5 Conclusion

Ces états de l'art nous ont permis de tirer plusieurs conclusions. Concernant le premier problème, il n'existe pas aujourd'hui de travaux explicitant les mécanismes de transformation entre les modèles organisationnels (modèle exigences PLM et modèle métier PLM) et le modèle informationnel (modèle personnalisé du logiciel PLM) pour un système PLM.

Concernant le deuxième problème, ce paragraphe a mis en lumière les limites et les contraintes induites par l'utilisation d'un unique modèle. Ainsi, notre cadre de modélisation doit supporter l'utilisation de plusieurs modèles métiers pour créer les différentes représentations du produit. Plusieurs travaux ont montré que l'ingénierie dirigée par les modèles est un facteur clé pour la mise en œuvre de l'interopérabilité entre les représentations du produit. Elle se révèle particulièrement adéquate pour répondre à nos problèmes pour plusieurs raisons. Elle n'impose pas de langage ce qui répond au besoin d'indépendance de la modélisation dans un contexte d'entreprise étendue. L'autonomie donnée à chaque modèle est également en adéquation avec la recherche d'agilité

opérationnelle. Elle prend aussi en compte les trois niveaux de modélisation concernés par nos problèmes. Enfin, certains travaux présentent des mécanismes de traces intéressants et ré-exploitable pour assurer la traçabilité entre les représentations du produit (cf. paragraphe 4.2.2 du chapitre 2) [Jézéquel et al., 06].

4 Notre proposition : un cadre de modélisation basé sur le paradigme d'ambivalence

Le chapitre 2 avait présenté une première ébauche du cadre. Suite à la formulation des deux problèmes (§ 2) et à l'état de l'art des travaux des méthodes supportant la construction du système PLM (§ 3), ce paragraphe a pour but de définir de manière détaillée le cadre de modélisation de notre proposition.

Ce paragraphe définit en détail les différents axes de ce cadre de modélisation (cf. § 4.1). Il positionne ensuite les problèmes identifiés dans le cadre de modélisation proposé (§ 4.2). Une question est alors posée : « Sur quel paradigme s'appuyer pour déterminer des mécanismes de cohérence entre différentes perceptions d'une même réalité ? » Ce paragraphe revient sur le paradigme d'ambivalence, qui répond à ce besoin (§ 4.3). Enfin, il met en lumière trois défis à relever par l'entreprise pour pouvoir utiliser le cadre de modélisation proposé (§ 4.5).

4.1 Les axes du cadre de modélisation

Le cadre de modélisation proposé a pour objectif de guider le raisonnement du modélisateur dans la mise en œuvre d'un système PLM. Il doit donc supporter l'analyse du système PLM et l'analyse des interfaces entre le système PLM et d'autres systèmes. Le système PLM supporte la création et la gestion de certaines représentations du produit. Ces représentations sont des instances du modèle personnalisé dans le logiciel PLM. Ainsi, le cadre doit supporter l'analyse conjointe de deux systèmes : le système PLM et le produit.

Suite à l'étude détaillée des problèmes, nous proposons d'affiner les axes d'étude définis dans le chapitre 2 comme suit (cf. figure 4.10) : un axe « maturité du produit », un axe « abstractions » et un axe « instanciation ».

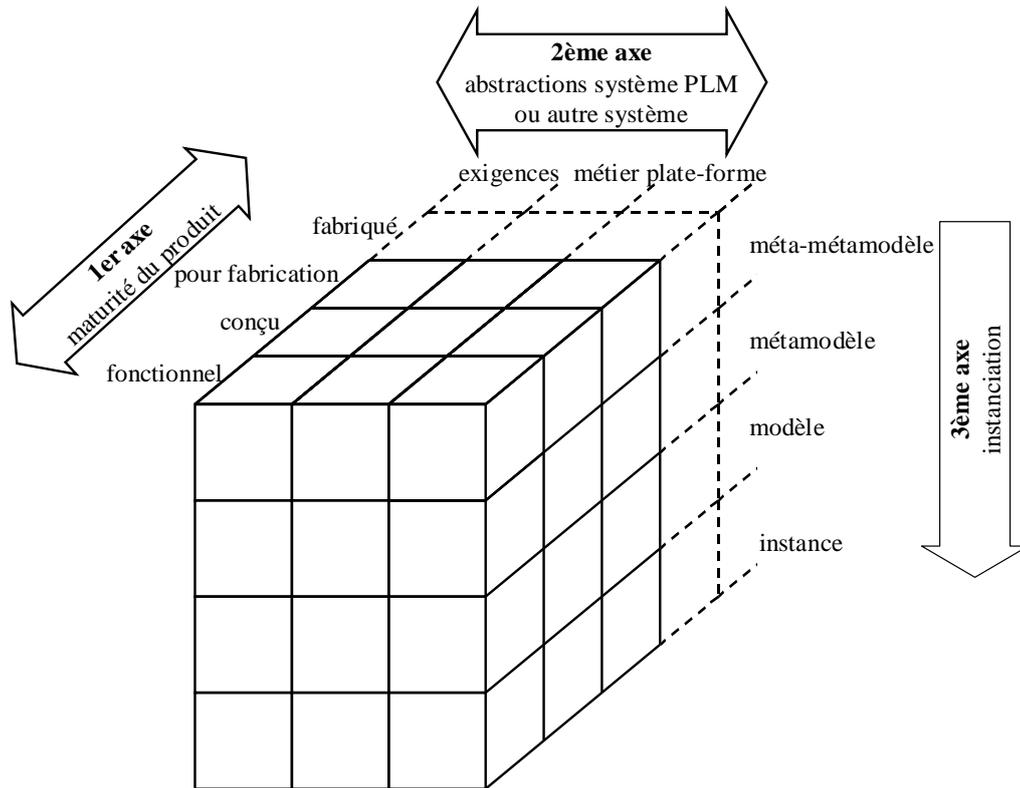


Figure 4.10 : Cadre de modélisation proposé pour supporter la mise en œuvre d'un système PLM

Les paragraphes suivants définissent la nature et les différents thèmes de chacun des axes en identifiant quel système (du système PLM ou du produit) est considéré. Nous définissons également les correspondances entre les thèmes du cadre et les caractéristiques du paradigme systémique à partir duquel le cadre de modélisation est construit.

4.1.1 Le premier axe

Dans le premier axe, le système considéré est le produit. Cet axe définit les états de maturité pour lesquels une représentation du produit est réalisée. Il inclut aussi bien les représentations théoriques du produit (représentations « TO BE ») que les enregistrements réalisés sur les produits physiques (représentations « AS IS »). Les états contenus par cet axe sont variables selon les entreprises. En effet, si l'entreprise propose à ses clients des produits qui sont configurés à la commande, elle doit gérer au sein de son système d'information des représentations des produits avant et après la configuration. Si elle fait du montage sur site, elle gère des représentations de fabrication en atelier et des représentations d'assemblage sur site. Si elle fait de la maintenance, elle gère des représentations des pièces de rechange. Dans la figure 4.10, les états proposés sont **fonctionnel**, **conçu**, **pour fabrication** et **fabriqué**. Fonctionnel, conçu et pour fabrication correspondent à des représentations du produit théorique. Fabriqué correspond à un enregistrement des propriétés du produit physique.

Dans ces travaux, nous ne traitons pas du pôle d'observation comportemental du paradigme systémique concernant le produit. Nous ne traitons pas non plus des interfaces du produit avec son environnement. Les différents thèmes de ce premier axe décrivent la structure des produits de l'entreprise (conformité au pôle d'observation « structure » du paradigme systémique). L'application du pôle d'observation génétique (évolution) du paradigme systémique se fait de deux manières. La première manière consiste à avoir des représentations qui définissent le produit à différents états de maturité. La deuxième manière consiste à gérer des versions des représentations. Le thème « fonctionnel » de l'axe correspond à l'identification de la finalité du système considéré, c'est-à-dire le produit.

4.1.2 Le deuxième axe

Dans le deuxième axe, les systèmes considérés sont le système PLM et les systèmes avec lesquels le système PLM est interfacé. Pour créer et gérer les représentations du produit correspondant aux différents états de maturité, plusieurs logiciels peuvent être utilisés. Dans ces logiciels, les représentations sont créées dans le format natif du logiciel et elles peuvent être transformées dans des formats standards. Le support informatique du système PLM est souvent réalisé à l'aide d'un logiciel PLM. Selon les entreprises, ce logiciel gère une ou plusieurs des représentations du cycle de vie du produit. Dans l'exemple de la figure 4.11, la représentation fonctionnelle est créée à partir d'un logiciel d'analyse fonctionnelle du produit. La représentation du produit conçu est créée et gérée à l'aide des logiciels CAO et PLM. La représentation pour fabrication est créée et gérée avec un logiciel ERP. Les représentations créées à partir du logiciel CAO sont en format natif et en format standard.

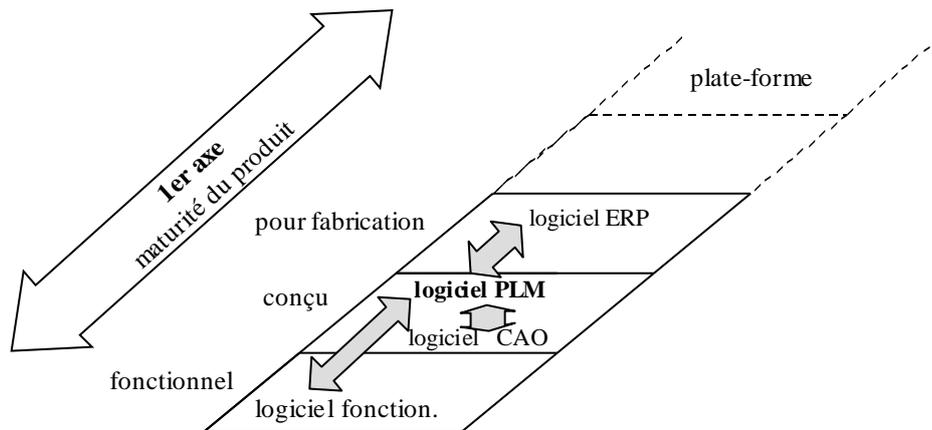


Figure 4.11 : Premier axe du cadre proposé : les interfaces entre les différents logiciels

Le logiciel PLM gère la cohérence entre les représentations du produit en s'appuyant sur des règles d'interface avec les autres logiciels. En supportant la formalisation de ces règles, l'utilisation du cadre traite la notion d'interface du paradigme systémique pour le système PLM.

Le deuxième axe définit les trois niveaux d'abstraction du système PLM et des systèmes interfacés avec le système PLM. Ces niveaux, identifiés dans le paragraphe 4.3 du chapitre 2, sont le niveau « **exigences** », le niveau « **métier** » et le niveau « **plate-forme** ».

La figure 4.12 illustre la combinaison de cet axe avec le premier axe au travers d'un exemple. Une entreprise définit au travers de trois cahiers des charges les besoins relatifs à la création et gestion des représentations fonctionnelles, conçues et pour fabrication du produit. Des modèles métiers PLM sont définis à partir de ces cahiers des charges. Enfin, ces modèles métier PLM doivent être implémentés dans trois plates-formes : un logiciel d'analyse fonctionnelle du produit, un logiciel PLM et un logiciel ERP.

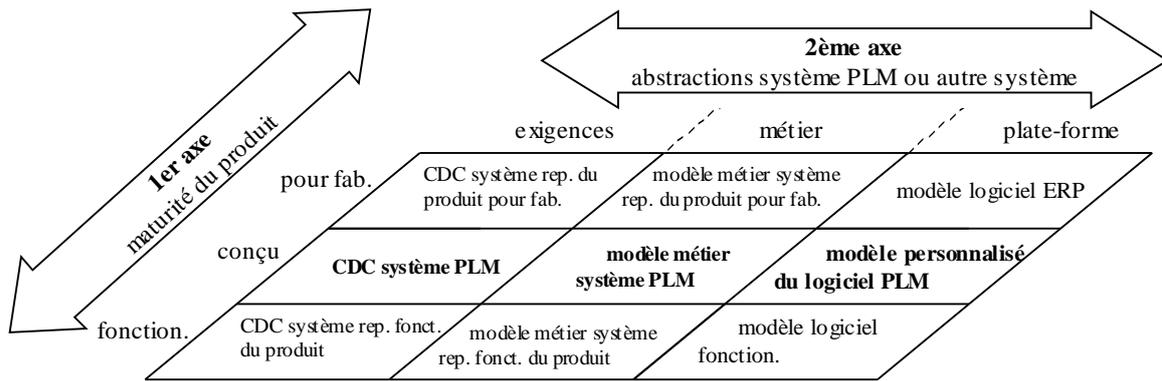


Figure 4.12 : Deuxième axe du cadre proposé : les différents modèles des systèmes

Dans ces travaux, nous ne traitons pas du pôle d’observation comportemental du paradigme systémique pour le système PLM. Les différents thèmes de l’axe décrivent la structure du système PLM (conformité au pôle d’observation « structure » du paradigme systémique). Le thème « exigences » correspond à l’identification de la finalité du système. Nous avons vu que la conformité au pôle d’observation génétique (évolution) du paradigme systémique se fait de deux manières. La première consiste à avoir des modèles du système PLM qui le définissent à différents états de maturité. Les trois modèles utilisés peuvent être considérés comme des définitions de plus en plus matures du système PLM. La seconde consiste à gérer des versions dans les modèles. Cette gestion de versions de modèles n’est pas abordée dans nos travaux.

4.1.3 Le troisième axe

Dans le troisième axe, les systèmes considérés sont le système PLM et le produit. Cet axe définit les niveaux de modélisation. La figure 4.13 illustre ainsi au travers d’un exemple que selon le niveau de modélisation considéré, les intersections entre les thèmes des axes définissent le modèle des produits dans un logiciel ou une représentation numérique d’un produit.

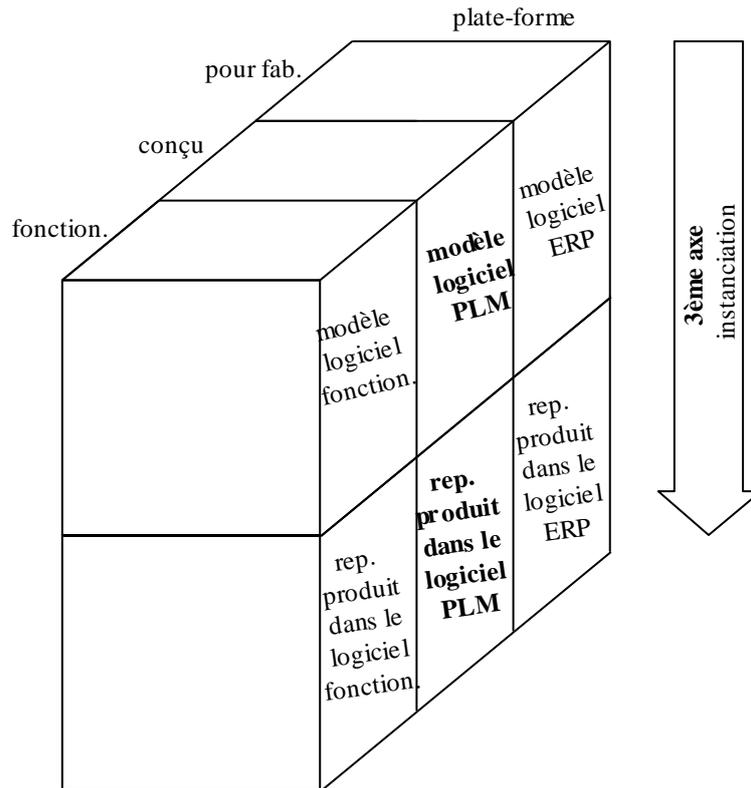


Figure 4.13 : Troisième axe du cadre proposé : les modèles et leurs instances (représentations produit)

Le paragraphe 2.1.1 de ce chapitre définit trois niveaux de modélisation à prendre en compte pour mettre en œuvre un logiciel PLM: métamodèle, modèle et représentation. Dans le paragraphe 2.3 du chapitre 2, l'architecture MDA précise que pour définir un métamodèle, il est nécessaire d'utiliser un méta-métamodèle (cf. chapitre 2, § 2.3). Le troisième axe définit ainsi les quatre niveaux de modélisation impliqués par la mise en œuvre d'un système PLM (cf. figure 4.14).

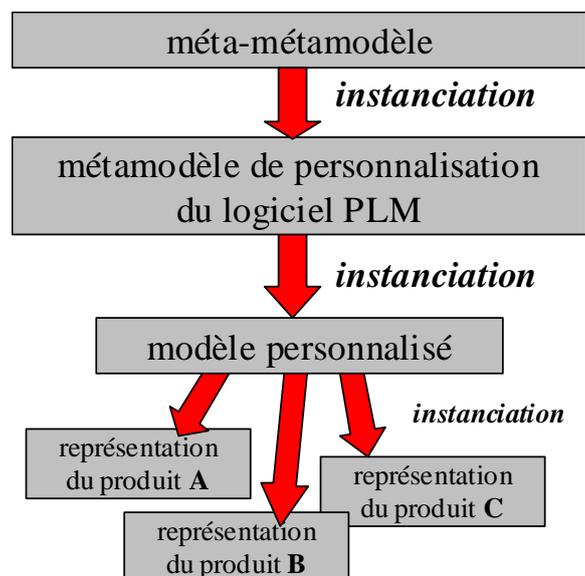


Figure 4.14 : Les quatre niveaux de modélisation concernant la mise en œuvre d'un logiciel PLM

4.2 Positionnement des deux problèmes dans notre cadre

Suite à sa construction, nous pouvons à présent positionner les deux problèmes identifiés dans le cadre de modélisation proposé (cf. figure 4.15). Chaque problème correspond à une projection sur le cadre de modélisation. Pour le problème 1, les différents modèles utilisés pour mettre en œuvre le système PLM ou les autres systèmes doivent être conformes à une même réalité : l'entreprise. Pour le problème 2, les différentes représentations utilisées pour décrire le produit doivent être conformes à une même réalité : le produit. Pour ce problème, le cadre de modélisation peut être utilisé de deux façons. La première utilisation supporte la définition des mécanismes de cohérence entre les représentations du produit gérées dans le système PLM. La deuxième utilisation supporte la définition des mécanismes de cohérence entre les représentations du produit gérées dans le système PLM et celles gérées dans d'autres systèmes.

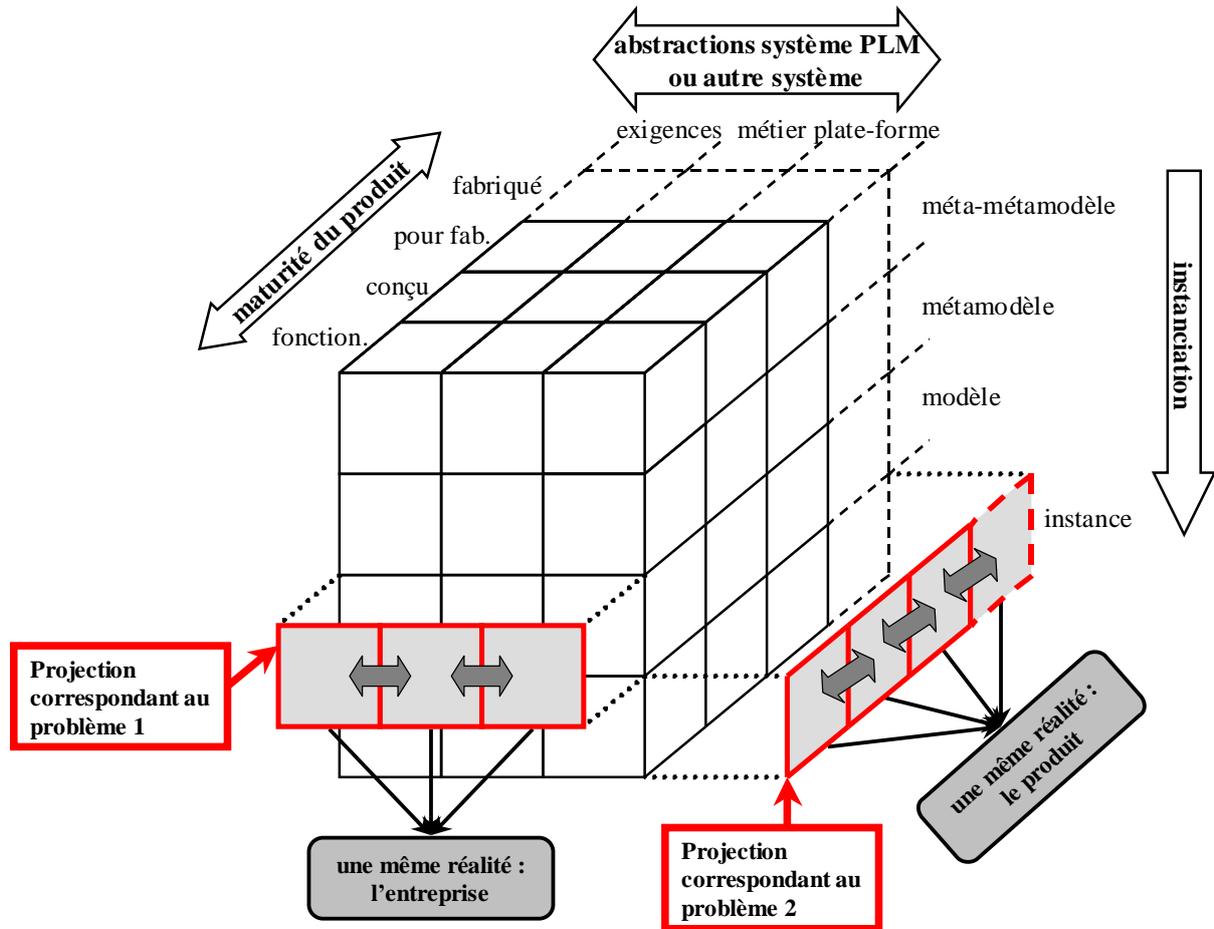


Figure 4.15 : Notre cadre de modélisation : identification des deux problèmes

Dans le chapitre 1, nous avons mis en lumière que les perceptions représentant une même réalité sont équivalentes et définies selon des pôles d'observation et des concepts certes différents, mais ni disjoints ni contradictoires. Les travaux traitant de l'interopérabilité entre les modèles visent à créer des alignements entre les modèles. Cependant, cet alignement ne suffit pas à assurer la cohérence et il incite à considérer que les différents modèles sont disjoints. Il a été défini dans le chapitre 2 que tout travail de modélisation s'inscrit dans une manière de percevoir le monde et donc, relativement à un paradigme. Ainsi, la question qui se pose à nous maintenant est : sur quel paradigme s'appuyer pour déterminer des mécanismes de cohérence entre différentes perceptions d'une même réalité ?

4.3 Le paradigme d'ambivalence

Le paradigme d'ambivalence perçoit une réalité ambivalente au moyen de plusieurs pôles d'observation. L'ambivalence d'une réalité implique que les perceptions correspondant à chaque pôle

d'observation ne doivent pas être juxtaposées mais considérées de manière complémentaire et corrélée [Chelli, 03]. Pour les problèmes identifiés, l'ambivalence concerne deux réalités, l'entreprise et le produit. En effet, l'entreprise est à la fois organisationnelle et informationnelle (problème 1) et le produit est à la fois une réponse à un besoin, un objet à fabriquer, un objet utilisé, un objet à maintenir, etc. (problème 2).

Le premier principe du paradigme d'ambivalence définit que ce qui s'applique à une perception d'une réalité s'applique aux autres perceptions de la même réalité. En s'appuyant sur ce constat, l'application de ce paradigme a pour but de promouvoir un raisonnement synergique au détriment du raisonnement réactif habituel (cf. figure 2.7, chapitre 2 § 3.2) [Chelli, 03]. Ce raisonnement nous incite, chaque fois que nous souhaitons comprendre, analyser ou agir, à nous servir simultanément des différentes perceptions de la réalité considérée et non de l'une ou de l'autre. Il devient alors possible de s'appuyer sur ce que nous savons d'une perception pour comprendre l'autre et rechercher systématiquement ce qui dans l'une correspond à ce qui est dans l'autre. Chaque pôle d'observation s'appuie sur des concepts qui lui sont propres pour créer des perceptions de la réalité étudiée. La correspondance entre les concepts des différents pôles d'observation doit ainsi être identifiée pour promouvoir le raisonnement synergique.

Ce raisonnement a pour objectif d'exploiter la valeur ajoutée amenée par chaque perception et non pas d'assujettir une perception à une autre perception. Le terme synergie définit ainsi un travail coordonné entre plusieurs familles d'acteurs ayant des perceptions différentes. La synergie crée une valeur ajoutée que chaque famille d'acteurs n'est pas capable de créer de manière indépendante. Cette valeur ajoutée n'est accessible que par le côté exploratoire des réflexions rendu possible grâce à un décloisonnement des services.

Pour promouvoir le raisonnement synergique, il est nécessaire de casser le fonctionnement de client fournisseur instauré entre les services. Dans le cas du problème 1, ce fonctionnement existe entre les responsables métiers (client) et les techniciens de l'information (fournisseur). Dans le cas du problème 2, ce fonctionnement existe entre les différents acteurs des étapes de la vie du produit. Dans le cycle de vie du produit, le service de conception du produit doit fournir les plans des produits afin que le service de fabrication puisse fabriquer ces derniers. Le service de fabrication est ainsi le client du service de conception. Promouvoir un raisonnement synergique implique de créer un fonctionnement collaboratif dans lequel tous les acteurs travaillent « autour » d'une même réalité.

Promouvoir un raisonnement synergique implique également que les connaissances de chaque perception doivent enrichir l'autre perception. Concernant le problème 1, les connaissances en cause sont celles relatives au fonctionnement de l'entreprise. Les techniciens de l'information sont capables de construire des systèmes d'information répondant aux besoins de l'entreprise en côtoyant les responsables métiers. De leur côté, les responsables métiers progressent par la mise en œuvre d'outils informatiques qui les amènent à prendre du recul et à analyser leurs méthodes de travail. Par exemple, la construction d'un système PLM amène une véritable réflexion sur le travail collaboratif autour du produit. Concernant le problème 2, les connaissances en cause sont celles relatives au produit. D'une part, les collaborateurs fabricant le produit doivent connaître ses fonctionnalités afin de choisir au mieux le procédé de fabrication. D'autre part, les concepteurs doivent avoir connaissance des opportunités offertes par les nouveaux moyens de production. Ils doivent également avoir des retours sur les conséquences de leur choix de conception afin d'améliorer la définition du produit concerné et celles des produits futurs.

La figure 4.16 illustre ainsi l'application du paradigme d'ambivalence sur les deux axes d'étude identifiés. Pour promouvoir un raisonnement synergique, le paradigme doit être appliqué entre tout couple de perceptions représentant une même réalité. Pour le système PLM, cela se traduit par des correspondances entre le modèle d'exigences PLM, le modèle métier PLM et le modèle plate-forme PLM. Pour le produit, cela se traduit par des correspondances entre les différentes représentations utilisées tout au long de la vie du produit.

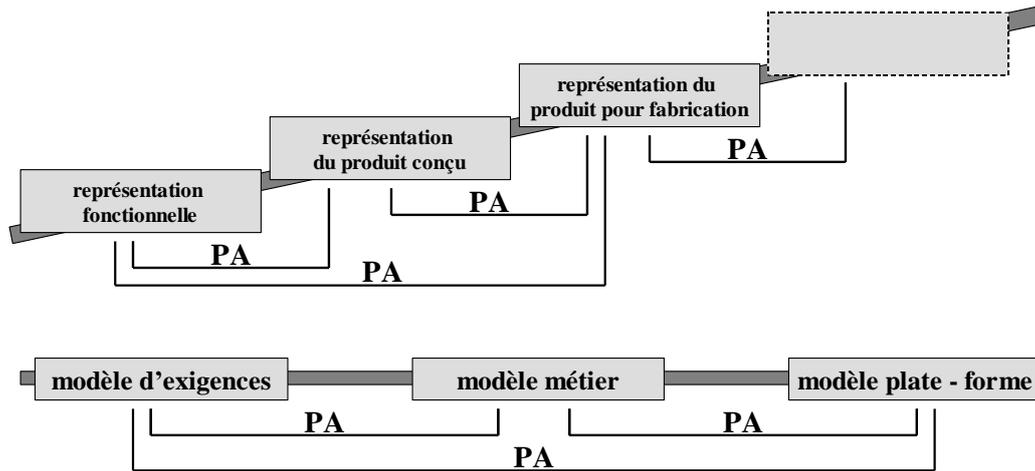


Figure 4.16 : Mise en œuvre du Paradigme d'Ambivalence (PA) sur les deux axes identifiés

Rappelons que pour passer du raisonnement réactif au raisonnement synergique, il est nécessaire d'établir une relation de correspondance bidirectionnelle et non de conformité unilatérale entre les perceptions (cf. figure 2.7, chapitre 2 § 3.2). Cette correspondance bidirectionnelle nécessite d'identifier les concepts à partir desquelles sont construites les perceptions et de formaliser les correspondances entre ces concepts. Ainsi, la résolution des problèmes de cohérence que nous avons formulés sur un niveau de modélisation implique l'identification de correspondances entre les concepts du niveau de modélisation supérieur.

Pour obtenir la plus grande cohérence possible, les correspondances sont à établir entre les entités mais également entre les relations qui structurent ces entités. Ainsi, dans le cadre de nos travaux, la figure 4.17 illustre le besoin d'établir des correspondances entre les entités et les relations des métamodèles pour répondre au besoin de cohérence entre les différents modèles du système PLM (problème 1). Elle illustre également le besoin d'établir des correspondances entre les entités et les relations des modèles pour répondre au besoin de cohérence entre les représentations du produit (problème 2).

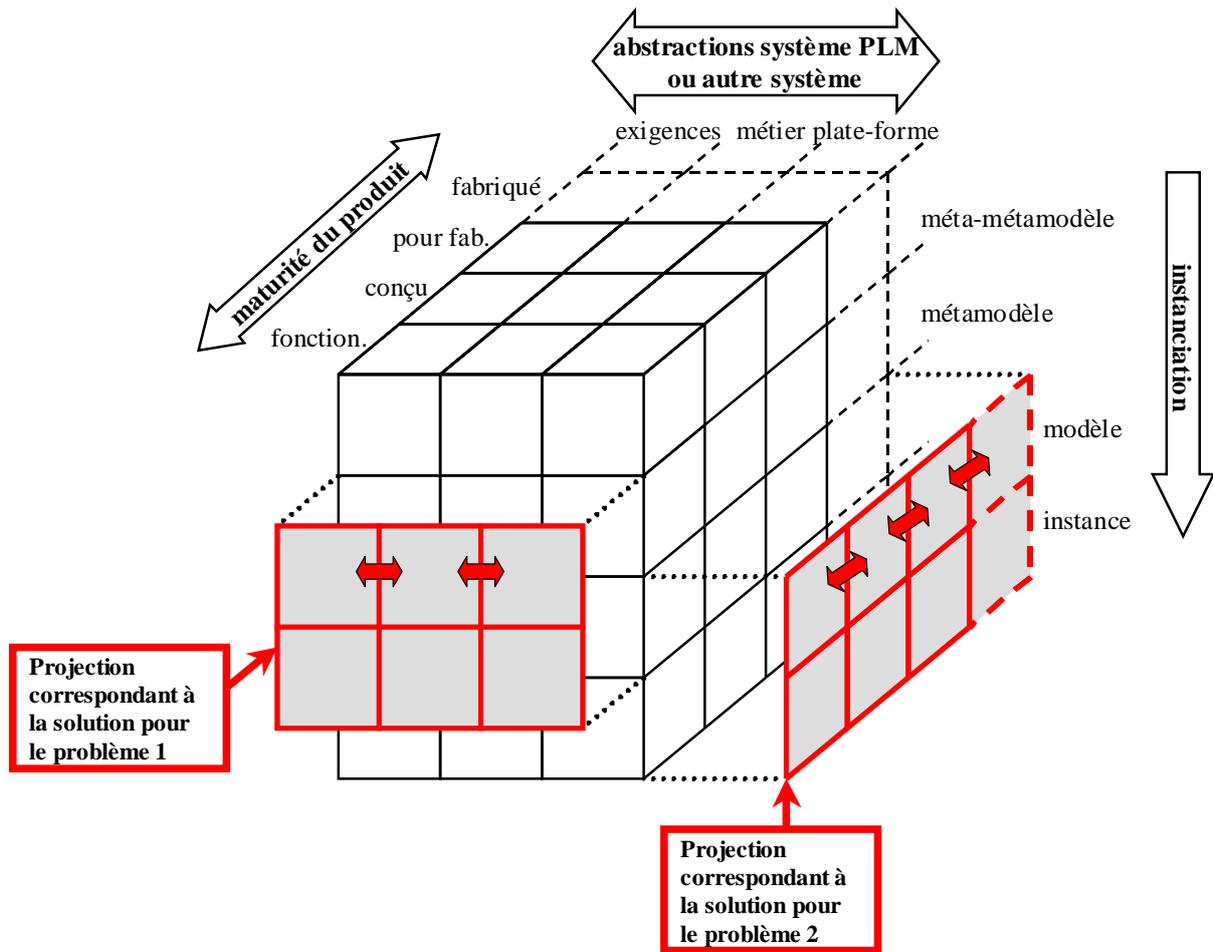


Figure 4.17 : Notre cadre de modélisation : identification des correspondances préconisées pour la mise en œuvre du paradigme d’ambivalence

4.4 Utilisation du cadre de modélisation

Pour utiliser le cadre de modélisation proposé, nous identifions trois défis pour l’entreprise.

Le premier défi consiste à avoir des perceptions formalisées et exploitables selon les différents pôles d’observation. Pour le problème 1, il a été mis en lumière dans le paragraphe 2.1.3 que le modèle métier PLM est souvent construit après le choix du logiciel PLM et que sa construction est influencée par les capacités du logiciel PLM choisi. Ce modèle métier PLM est la plupart du temps construit à partir d’outils bureautiques donc difficilement exploitables pour du travail de modélisation. Pour le problème 2, rendre la formalisation exploitable implique par exemple de définir les nomenclatures des produits à l’aide de structures d’objets dans des logiciels PLM au lieu de les gérer à l’intérieur de documents (les plans la plupart du temps).

Le deuxième défi consiste à avoir des représentations construites à partir de concepts qui permettent l’identification de correspondances. Pour obtenir une synergie la plus grande possible, il est nécessaire que les concepts des métamodèles (problème 1) et des modèles (problème 2) soient pertinents et précis.

Le troisième défi consiste à ce que les perceptions soient cohérentes. Pour cela, nous proposons la mise en œuvre de deux mécanismes visant à implémenter des règles de cohérence. Le premier est la formalisation de règles de transformations entre les métamodèles (problème 1) et de règles de transformations entre les modèles (problème 2). Le deuxième est la mise en œuvre de vérification via des comparaisons.

La figure 4.18 illustre les mécanismes de transformations pour notre cadre de modélisation. En formalisant des règles de cohérence sur un niveau, les transformations et les vérifications peuvent être faites de manière assistée au niveau inférieur [Pinel et al., 11a].

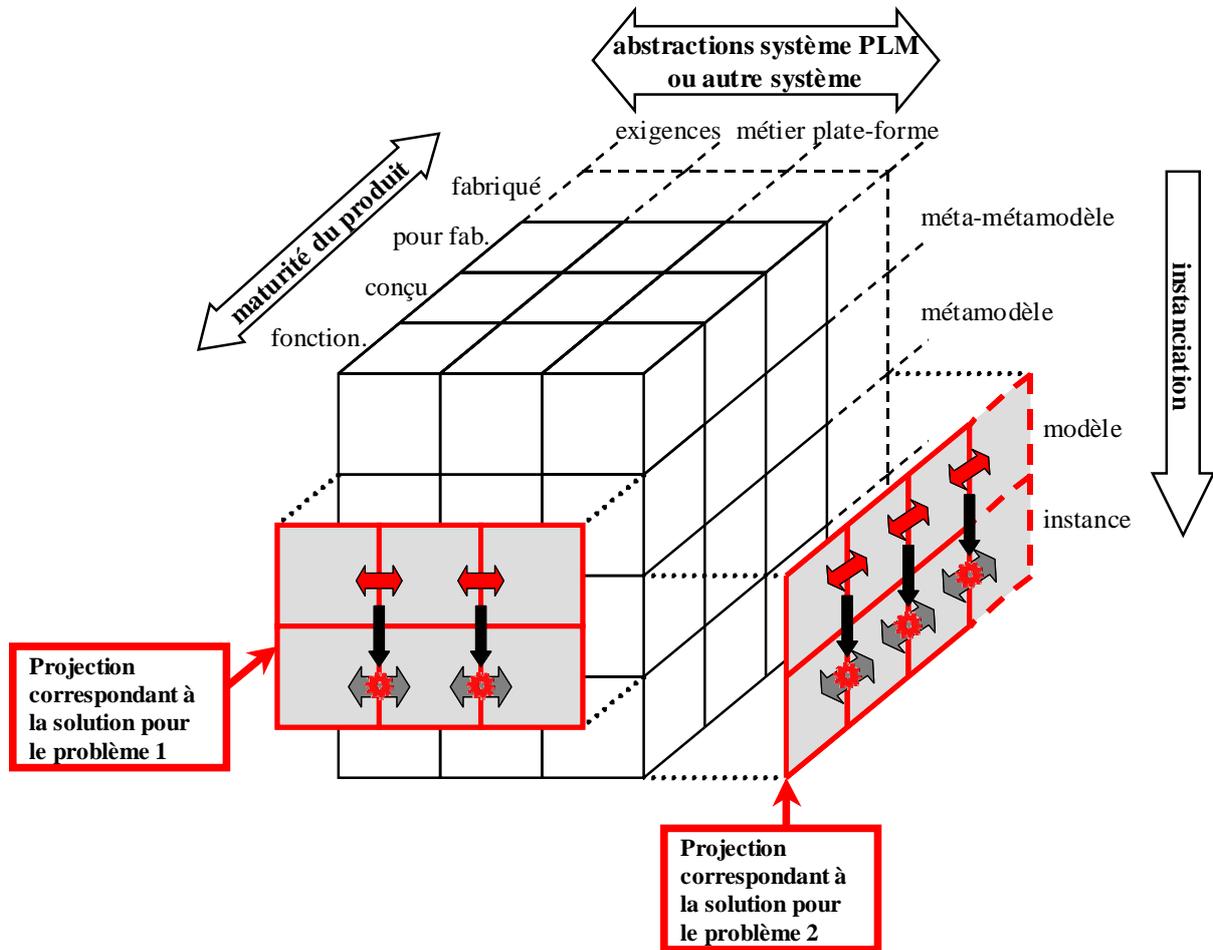


Figure 4.18 : Notre cadre de modélisation : identification des mécanismes de transformations et de vérification

4.5 Conclusion

Tout au long de ce paragraphe, le cadre de modélisation que nous proposons en réponse aux problèmes a été construit de manière détaillée. Les axes du cadre ont tout d'abord été redéfinis et des correspondances avec le paradigme systémique ont été établies. Il a été ainsi déterminé que le cadre dans son état actuel ne prend pas en compte tous les pôles d'observation nécessaires. Il faudra ainsi faire des travaux complémentaires pour le rendre complet au regard du paradigme systémique. Une fois les trois axes du cadre définis, le paradigme d'ambivalence sur lequel s'appuie le cadre et les conséquences de ce paradigme sur le cadre ont été introduits.

Le problème formulé à la fin du chapitre 1 était comment réaliser une mise en œuvre du système PLM en adéquation avec les deux buts de l'entreprise à savoir : avoir la capacité de fournir le bon produit sur le bon marché au bon moment et avec le bon coût et développer l'agilité opérationnelle ?

Promouvoir un raisonnement synergique plutôt qu'un raisonnement réactif constitue un élément de réponse à ce problème. Concernant le fait de fournir le bon produit, adopter un raisonnement synergique permet de limiter les distorsions entre les différentes représentations du produit. Concernant l'agilité opérationnelle, [Chelli, 03] a montré qu'un frein important à la mise en œuvre de l'agilité est la séparation entre les familles d'acteurs. Il a également défini que la mise en œuvre de synergies et l'identification de correspondances entre les concepts des différents pôles d'observation permettent de mettre en œuvre efficacement et rapidement des évolutions de la réalité opérationnelle.

En promouvant les interactions entre les perceptions informationnelle et organisationnelle, le raisonnement synergique permet ainsi de développer l'agilité opérationnelle recherchée.

Au-delà de l'utilisation du cadre de modélisation, l'objectif à atteindre est bien un changement de paradigme et ce changement est porté par les hommes. Les buts de l'approche PLM définis dans le paragraphe 4.1.6. et les principes définis dans le paragraphe 4.2.1. du chapitre 1 s'en trouvent éclairés. En effet, les correspondances bidirectionnelles et la rupture du mode client-fournisseur correspondent bien à la mise en œuvre de l'intégration recherchée (problèmes 1 et 2). Elles répondent également au besoin de créer un environnement centré sur le produit (problème 2). Les critères de maturité concernant la mise en place d'une organisation plus transversale et transdisciplinaire prennent également tout leur sens (cf. chapitre 3, § 4.1.3). Adopter un raisonnement synergique implique ainsi de ne plus assujettir une perception à une autre et de donner une importance équivalente à chacune. Le but recherché est la mise en place d'une boucle vertueuse d'amélioration continue commune.

Par conséquent, la formulation de règles de cohérence est le fruit d'un travail collaboratif fait par les hommes concernés par les perceptions en cause. Pour le problème 1, la collaboration se fait entre les responsables métiers et les techniciens de l'information. Pour le problème 2, la collaboration se fait entre les responsables des différents métiers impliqués tout au long de la vie du produit.

Pour mettre en œuvre les correspondances entre les entités des métamodèles (problème 1) et modèles (problème 2), nous proposons d'utiliser les mécanismes de l'ingénierie dirigée par les modèles. Cependant, ces derniers ne prennent pas en compte les correspondances entre le sens des entités, c'est à dire entre leurs sémantiques. Il sera ainsi nécessaire dans de futurs travaux d'établir des correspondances entre des ontologies³⁷ afin de compléter notre proposition.

Suite à la définition du cadre de modélisation, les paragraphes suivants ont pour objectif d'illustrer son utilisation pour répondre aux deux problèmes identifiés.

5 Cadre de modélisation et mise en œuvre du système PLM (Problème 1)

Il a été défini dans le paragraphe 3.4 qu'aucune des méthodes traitant de la mise en œuvre des systèmes PLM n'est explicite quant à la construction du modèle personnalisé du logiciel PLM. Le paragraphe 4 a également mis en lumière l'importance de créer des correspondances entre les modèles des perceptions informationnelle et organisationnelle.

Ainsi, ce paragraphe a pour objectif de montrer comment utiliser le cadre de modélisation proposé pour répondre au problème formulé dans le paragraphe 2.1.4.

5.1 Rappel du problème

Pour rappel, trois questions ont été identifiées dans le paragraphe 2.1.4. pour définir le problème 1.

La première, que nous appelons **problème A**, est « Comment aider l'entreprise à construire successivement les modèles d'exigences PLM, métier PLM et personnalisé du logiciel PLM ? »

La deuxième, que nous appelons **problème B**, est « Comment assurer la cohérence entre ces différents modèles qui représentent une même réalité : l'entreprise ? »

La troisième, que nous appelons **problème C**, est « Comment créer le modèle personnalisé du logiciel PLM à partir du métamodèle de personnalisation du logiciel PLM choisi ? »

³⁷ ontologie : une ontologie implique ou comprend une certaine vue du monde par rapport à un domaine donné. Cette vue est souvent conçue comme un ensemble de concepts – entités, attributs, processus -, leurs définitions et leurs inter-relations. On appelle cela une conceptualisation. Une ontologie peut prendre différentes formes mais elle inclura nécessairement un vocabulaire de termes et une spécification de leur signification. p 1264 de [Akoka et al., 06]

5.2 Comment utiliser le cadre pour répondre aux problèmes identifiés

5.2.1 Les métamodèles PLM

Pour créer les modèles d'exigences PLM et métiers PLM, plusieurs méthodes ont été identifiées dans l'état de l'art du paragraphe 3.2.1. Ces méthodes proposent de construire les modèles à partir d'un langage de modélisation de système d'information (le langage UML, par exemple) ou à partir de modèles génériques PLM. Dans le paragraphe précédent, il a été déterminé que plus les correspondances entre les entités et relations des métamodèles utilisés sont riches plus la synergie créée est grande.

Ainsi, notre proposition au regard de notre cadre consiste à créer des métamodèles PLM à partir des modèles génériques existants. Le métamodèle du modèle des exigences PLM peut être construit à partir de plusieurs travaux [Zancul, 09] [Le Duigou, 10] [OMG, 11]. Le métamodèle métier PLM relatif aux informations peut être construit à partir de différents travaux [Gzara, 00] [Le Duigou, 10] ; l'expérimentation (cf. chapitre 5) traite de la création de ce métamodèle. Nous prenons comme base de travail un modèle générique [Gzara, 00] que nous adaptons en fonction des résultats d'une étude réalisée au sein du Club PLM. La définition du métamodèle d'exigences PLM et l'intégration des traitements (sous forme d'opérations et de processus) dans le métamodèle métier PLM font partie de nos perspectives.

Une fois ces métamodèles définis dans un environnement de modélisation, l'entreprise construit ses modèles par un mécanisme d'instanciation (problème A).

5.2.2 Transformation entre modèle métier PLM et modèle personnalisé du logiciel PLM

Nous proposons de créer le modèle personnalisé du logiciel PLM (problème C) au moyen d'une transformation assistée du modèle métier PLM. Cette transformation est réalisée dans un environnement de modélisation. Conformément à notre cadre de modélisation, les règles de transformation sont basées sur l'identification de correspondances entre les entités et les relations du métamodèle métier PLM et du métamodèle de personnalisation du logiciel PLM. Cette transformation assistée permet de garantir la cohérence entre le modèle métier PLM et le modèle personnalisé du logiciel PLM (problème B).

La mise en œuvre de ce mécanisme implique de construire le métamodèle de personnalisation du logiciel PLM dans l'environnement de modélisation utilisé pour réaliser les transformations. Le paragraphe suivant présente le processus proposé pour cette construction.

5.2.3 Le métamodèle de personnalisation du logiciel PLM

La modélisation du système PLM concerne la structure des données gérées et les traitements mis en œuvre sur ces données. Comme pour le métamodèle métier PLM, le processus que nous définissons pour construire le métamodèle de personnalisation du logiciel PLM est focalisé sur la partie « données ». L'intégration des éléments relatifs aux traitements fait partie de nos perspectives. Les sources nécessaires à notre processus sont le modèle relationnel de la base de données du logiciel PLM, des tests de personnalisation réalisés sur le logiciel PLM et les connaissances de divers experts.

Le processus proposé est constitué de quatre étapes :

- Etape 1 : Etude du modèle relationnel et identification des tables à prendre en compte. Dans un logiciel PLM, certaines tables servent au fonctionnement du logiciel, d'autres servent à la personnalisation du modèle et d'autres servent au stockage des données. Notre objectif étant de définir le modèle personnalisé, il est nécessaire d'identifier les tables concernées par la personnalisation du logiciel PLM.
- Etape 2 : Si nécessaire, reconstruction du modèle relationnel pour obtenir un modèle relationnel normalisé afin d'avoir une structure de données cohérente et non redondante.

- Etape 3 : Construction du modèle entité-association de la base de données par rétro-conception du modèle normalisé relationnel. Outre les entités et les associations, ce modèle entité-association décrit les différentes contraintes liées au fonctionnement du logiciel PLM (cardinalités, contraintes d'inclusion, d'exclusion, de totalité, etc.). Les environnements de modélisation permettent de générer automatiquement le modèle relationnel à partir du modèle entité-association. Il est ainsi possible de valider la rétro-conception en comparant le modèle relationnel obtenu avec le modèle initial.
- Etape 4 : Définition du métamodèle de personnalisation du logiciel PLM dans l'environnement de modélisation utilisé pour mettre en œuvre les transformations entre modèle métier PLM et modèle personnalisé du logiciel PLM.

5.3 Conclusion

L'objectif de ce paragraphe était d'illustrer l'utilisation du cadre de modélisation pour répondre au problème relatif à la mise en œuvre du système PLM.

Ce paragraphe a présenté les processus de définition des métamodèles des modèles d'exigences PLM, métiers PLM et personnalisé du logiciel PLM. Le chapitre 5 a pour objectif de décrire plus précisément les processus relatifs au métamodèle du modèle métier PLM et au métamodèle du modèle personnalisé du logiciel PLM.

La formalisation des métamodèles et des règles de correspondances dans un environnement de modélisation rend les modèles productifs, elle facilite leur mise à jour et elle permet une capitalisation des savoirs-faire en ingénierie des systèmes d'information. Toutefois, elle n'est pas une fin en soi et elle requiert de nombreuses ressources. Ainsi, un juste équilibre entre formalisation et non-formalisation doit être trouvé. En effet, la synergie naît de la réduction de la séparation relative aux langages. Cependant, elle doit également trouver sa source dans des approches visant à réduire la séparation relative aux échelles (cf. § 2.3.3 du chapitre 1). La séparation des échelles est due au fait que le cycle organisationnel est court (exigence du marché) alors que celui informationnel est long (au regard du cycle organisationnel). La mise en œuvre du système PLM selon une méthode agile [Cunningham et al., 01] peut ainsi faire partie des solutions applicables pour réduire cette séparation et pour favoriser l'agilité opérationnelle. En effet, en appliquant cette méthode, le système est délivré sous forme de fragments à intervalles réguliers de temps ce qui contribue à diminuer la longueur du cycle informationnel au regard de celle du cycle organisationnel.

Ce besoin de synergie et de progressions corrélées entre les deux perceptions illustre que l'entreprise doit travailler de manière simultanée ses différents axes de maturité (cf. § 4.1.3 du chapitre 3). En effet, l'efficacité globale est le fruit d'une montée en maturité conjointe entre les perceptions informationnelle et organisationnelle.

6 Cadre de modélisation et cohérence entre les différentes représentations du produit (Problème 2)

Le paragraphe 4 a mis en lumière l'importance de la mise en œuvre d'un raisonnement synergique. En réduisant les séparations entre différentes perceptions d'une réalité, ce raisonnement permet de développer l'agilité opérationnelle indispensable à l'entreprise. Pour réduire ces séparations, il est nécessaire d'identifier des correspondances entre les entités et les relations permettant de construire les perceptions. Cette identification est le résultat d'un travail collaboratif entre les acteurs concernés par les perceptions en cause. C'est également un travail très spécifique sur les représentations des produits car elles sont propres à l'entreprise.

Ainsi, l'objectif de ce paragraphe n'est pas de définir des pratiques à appliquer mais plutôt d'illustrer comment les concepts relatifs à notre cadre de modélisation peuvent amener des éléments de réponse au problème identifié [Pinel et al., 12a] [Pinel et al., 12b].

Après avoir rappelé les différentes questions relatives à notre problème (§ 6.1), nous proposons d'illustrer l'utilisation du cadre de modélisation au travers de l'exemple des nomenclatures du produit de conception et pour fabrication (§ 6.2). Des éléments de réponse sont ainsi identifiés pour les problèmes de traçabilité et d'analyse d'impact (§ 6.3) et pour les problèmes de cohérence (§ 6.4).

6.1 Rappel du problème

Pour rappel, quatre questions ont été identifiées dans le paragraphe 2.2 pour définir le problème 2.

La première question, que nous appelons **problème D**, est « Comment assurer la traçabilité entre les différentes représentations du produit utilisées tout au long de sa vie ? »

La deuxième question, que nous appelons **problème E**, est « Comment supporter l'analyse d'impact d'une modification ? »

La troisième question, que nous appelons **problème F**, est « Comment s'assurer que les exigences définies par la représentation construite dans une étape (une étape correspondant à un univers de discours propre à un métier : conception, fabrication, etc.) sont bien comprises par l'étape suivante dont le métier est différent ? »

La quatrième question, que nous appelons **problème G**, est « Comment s'assurer que de nouvelles exigences définies par une représentation du produit propre à un métier ne sont pas en conflit avec les exigences définies par une représentation du produit fournie par un autre métier ? »

6.2 Les représentations du produit de conception et pour fabrication

Dans le paragraphe 2.2, deux représentations de la structure d'un produit correspondant à deux métiers différents ont été présentées.

La *nomenclature de conception* du produit représente la structure des organes du produit qui répond au besoin fonctionnel exprimé par le cahier des charges, elle est réalisée dans une logique de conception fonctionnelle. Des sous-ensembles fonctionnels sont définis pour être utilisés sur d'autres produits ou pour être validés de manière indépendante. Une fois le produit spécifié, le procédé de réalisation est défini pour préciser si chaque organe est acheté ou fabriqué. Ces précisions permettent de déterminer la *nomenclature pour fabrication* du produit. Elle représente la structure des organes du produit dans le but de le fabriquer. Elle répond aux besoins des gestions d'approvisionnement, de la fabrication, des flux et des stocks dans l'atelier.

Ainsi, la *nomenclature de conception* du produit est constituée d'organes spécifiés et répondant à un besoin fonctionnel : composant catalogue, composant de conception, sous-ensemble de conception. Cette représentation du produit est habituellement gérée dans un logiciel PLM. La *nomenclature pour fabrication* du produit est constituée d'organes répondant aux besoins de réalisation : composant et sous-ensemble achetés, composant et sous-ensemble fabriqués (états finis ou intermédiaires) et matières premières. Cette représentation du produit est habituellement gérée dans un logiciel ERP (Enterprise Resource Planning).

Tout au long de ce paragraphe, nous nous référons à ces deux représentations issues du métier de la conception et du métier de la fabrication pour illustrer des réponses à différents problèmes de cohérence. La figure 4.19 positionne notre cas d'étude dans le cadre de modélisation.

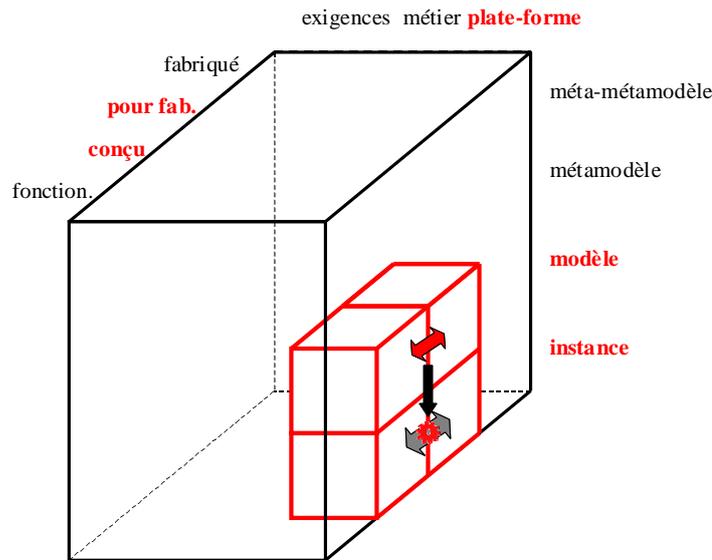


Figure 4.19 : Positionnement du cas d'étude dans le cadre de modélisation

Le paragraphe 6.3. est consacré à la résolution des problèmes D et E et le paragraphe 6.5. à celle des problèmes F et G.

6.3 Réponse aux problèmes D et E : mise en œuvre de références croisées

Pour mettre en œuvre des correspondances bidirectionnelles entre deux perceptions, chaque entité d'une perception doit porter l'identifiant du ou des entités qui lui correspondent dans l'autre perception. Dans les nomenclatures, ceci peut être mis en œuvre par un mécanisme de « référence + version » croisé. L'organe de la nomenclature de conception a parmi ses attributs la « référence + version » de l'organe de la nomenclature pour fabrication auquel il correspond et inversement.

La figure 4.20 illustre ce principe avec la nomenclature de conception et la nomenclature pour fabrication d'un roulement à billes. Dans le modèle personnalisé du logiciel PLM, un champ a été prévu pour définir les références et versions des organes pour fabrication correspondants. Dans le modèle du logiciel ERP, un champ a été prévu pour définir les références et versions des organes de conception correspondants. Dans notre exemple, les deux nomenclatures ont un système de référencement qui leur est propre. Des versions sont utilisées pour les organes de conception mais aussi pour les organes de fabrication. En effet, la version d'un organe de fabrication est incrémentée lorsque le procédé de fabrication de l'organe évolue. Ces versions « de fabrication » permettent de différencier les lots de produits physiques ayant été fabriqués avec l'ancien procédé de fabrication de ceux fabriqués avec le nouveau procédé.

Les correspondances entre la nomenclature de conception et la nomenclature pour fabrication ne sont pas bijectives. Ainsi, lorsqu'un sous-ensemble de conception « disparaît » dans la nomenclature pour fabrication, son référencement est réalisé au niveau supérieur. Dans notre exemple, le sous-ensemble cage bille est référencé dans le roulement. En effet, il est nécessaire que les exigences définies dans ce sous-ensemble soient prises en compte par la fabrication. Lorsqu'un sous-ensemble de fabrication apparaît, le référencement est réalisé au niveau supérieur des organes qui le composent. Dans notre exemple, le sous-ensemble joint bague a pour référence de conception le roulement. En effet, c'est dans ce dernier que sont décrites les exigences d'assemblage que ce sous-ensemble doit respecter. Lorsqu'un brut ou un état intermédiaire est créé, il hérite du référencement de l'organe auquel il est rattaché. Dans notre exemple, le brut de la bague a pour référence de conception la bague elle-même. En effet, c'est dans l'organe lui-même que sont définies les exigences de matière, diamètre extérieur, etc., qui permettent de définir le brut.

Ainsi, on retrouve le principe de l'ambivalence selon lequel ce qui s'applique à une perception s'applique à l'autre perception et inversement. Même si les correspondances ne sont pas bijectives, à tout organe de la nomenclature de conception correspond au moins un organe de la nomenclature pour

fabrication et à tout organe de la nomenclature pour fabrication correspond au moins un organe de la nomenclature de conception.

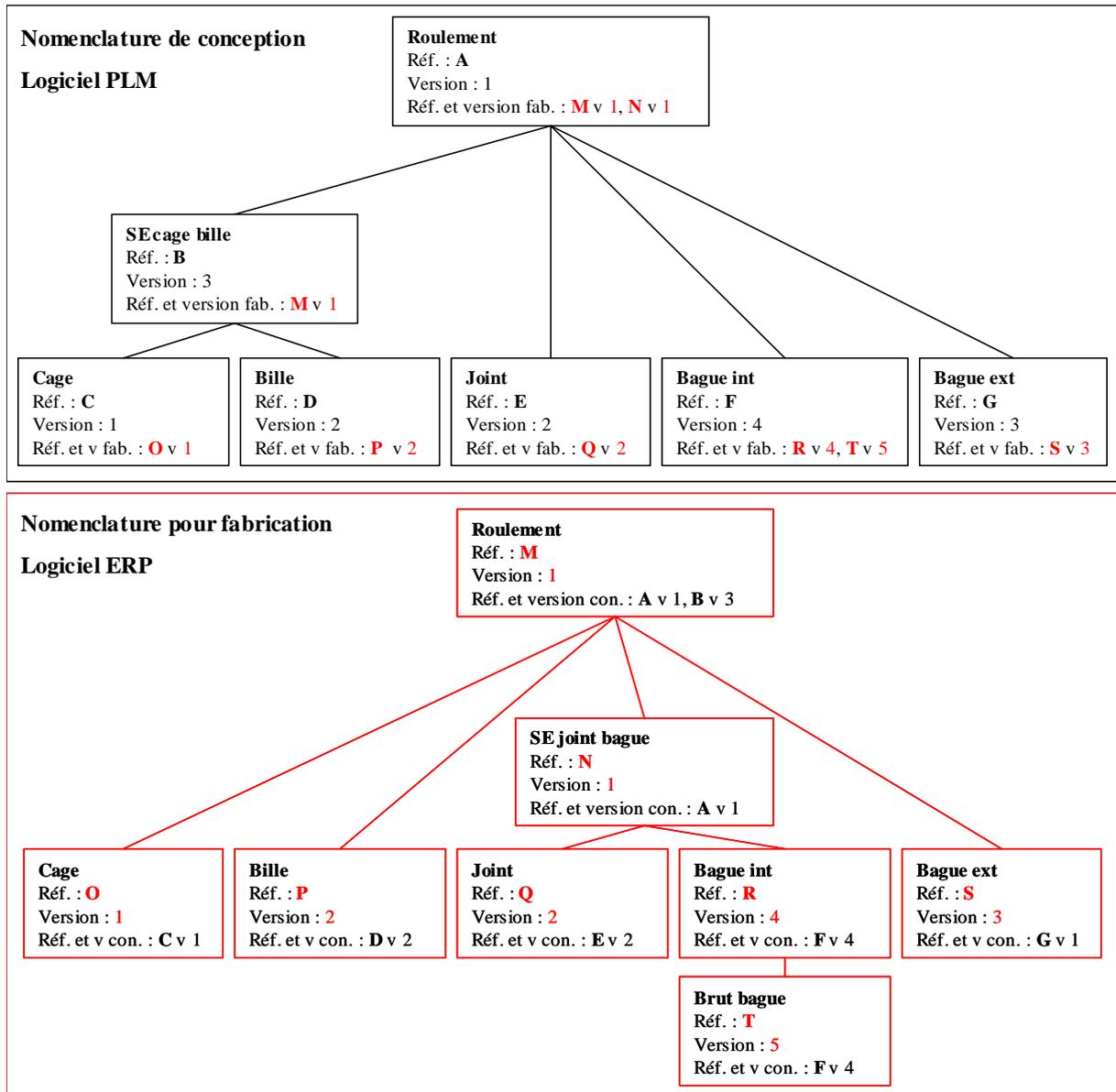


Figure 4.20 : Exemple de mise en œuvre de références croisées

Ce système croisé fait ainsi partie des solutions possibles pour répondre aux besoins de traçabilité (problème D) et d'analyse de l'impact d'une modification d'une représentation sur les autres représentations (problème E).

6.4 Définition de règles de cohérence

6.4.1 Nos réponses aux problèmes F et G

Dans le paragraphe 4.3, nous avons vu que la mise en œuvre du raisonnement synergique implique d'établir des correspondances entre les entités et les relations permettant de créer les représentations des deux perceptions. Nous proposons de nous appuyer sur ces correspondances pour amener des éléments de réponse aux problèmes F et G.

Pour rappel, le problème F est : « Comment s'assurer que les exigences définies par la représentation construite dans une étape (une étape correspondant à un univers de discours propre à un métier : conception, fabrication, etc.) sont bien comprises par l'étape suivante dont le métier est différent ? »

Ce problème requiert d'éviter les ambiguïtés en interdisant l'emploi de synonymes et de polysèmes entre les modèles plate-forme de deux métiers différents. Cette règle qui vise à n'utiliser que des termes spécifiques pour chaque concept permet en effet d'éviter les confusions. Par exemple, un sous-ensemble fonctionnel et un sous-ensemble de fabrication n'ont pas la même signification. Spécifier le terme « sous-ensemble » en lui ajoutant le qualificatif « fonctionnel » ou « de fabrication » permet ainsi une meilleure compréhension. Adapter le modèle de données utilisé par le logiciel ERP est souvent difficile. Dans ce cas, une solution est la dénotation des éléments du modèle personnalisé du logiciel PLM par des termes qui ne sont pas utilisés par le logiciel ERP.

Pour rappel, le problème G est : « Comment s'assurer que de nouvelles exigences définies par une représentation du produit propre à un métier ne sont pas en conflit avec les exigences définies par une représentation du produit fournie par un autre métier ? »

La *nomenclature de conception* et celle *pour fabrication* représentent des perceptions différentes de la structure d'un même produit. Ainsi, les correspondances doivent être établies entre les organes des deux nomenclatures, entre les relations hiérarchiques entre ces organes et entre les quantités associées à ces relations. La mise en œuvre de règles de cohérence limite les contradictions entre des exigences définies par des métiers différents. Cette mise en œuvre peut être faite de deux manières. La première consiste à définir des règles de transformation afin de **supporter** la construction d'une représentation à partir d'une autre représentation. Dans notre exemple, cela revient à supporter la création de la définition du procédé de fabrication et de la *nomenclature pour fabrication* associée à partir de la nomenclature de conception. Par exemple, des correspondances peuvent être établies entre deux définitions génériques. Ceci est particulièrement approprié quand des familles de produits sont identifiables. Pour illustrer, prenons le cas d'un roulement à billes. Il peut être défini, que pour tout roulement à billes conçu, correspond une gamme de fabrication type et une nomenclature associée. Ces mécanismes peuvent être implémentés via des outils tels que les modules MPM (Manufacturing Process Management) des logiciels PLM ou via les environnements de modélisation de l'ingénierie dirigée par les modèles. La deuxième manière de mettre en œuvre des règles de cohérence consiste à **vérifier** que la nouvelle représentation construite ne contient aucune contradiction vis à vis des représentations existantes. Cette vérification est réalisée en restructurant les deux nomenclatures à confronter. Cette réécriture de graphes a pour objectif d'obtenir deux structures pouvant être comparées. Dans notre exemple, elle consiste pour la *nomenclature de conception* à enlever les sous-ensembles (tout en gardant leurs composants avec la bonne quantité). Elle consiste pour la *nomenclature pour fabrication* à enlever les sous-ensembles (tout en gardant leurs composants avec la bonne quantité) et à enlever les bruts de fabrication et les organes correspondant à des états intermédiaires. Les deux structures obtenues peuvent être comparées suite à cette réécriture.

6.4.2 Formalisation des règles de cohérence

La **première étape** consiste à identifier les entités des univers de discours et à identifier les règles pour définir les règles de cohérence ; ces règles sont spécifiques à chaque entreprise. Elles peuvent être génériques, par exemple : un composant de la *nomenclature pour fabrication* ne peut pas être en état « valide » si le composant correspondant dans la *nomenclature de conception* n'est pas en état « valide ». Elles peuvent aussi être très spécifiques aux produits de l'entreprise. Par exemple, toute bague de la *nomenclature de conception* correspond à un article fabriqué dans la *nomenclature pour fabrication*. La **deuxième étape** consiste à définir des entités adaptées dans les modèles plate-forme afin de pouvoir implémenter les règles identifiées dans le système d'information. La **troisième étape** consiste à formaliser les règles de cohérence entre les modèles plate-forme.

Pour illustrer ces étapes, prenons deux exemples, un relatif à la mise en œuvre de correspondances entre les entités et un relatif à la mise en œuvre de correspondances entre les structures.

Dans le premier exemple, la première étape consiste à identifier les entités « composants standards » pour le métier de la conception, les entités « composants achetés » pour le métier de la fabrication et la

règle métier « tout composant standard dans la *nomenclature de conception* correspond à un composant acheté dans la *nomenclature pour fabrication* ». La deuxième étape consiste à choisir des entités permettant d'implémenter cette règle. Nous choisissons de représenter l'entité « composant standard » par une classe d'objets « composant standard » dans le modèle personnalisé du logiciel PLM et l'entité « composant acheté » par une classe d'objets « article » avec une propriété ayant une valeur « acheté » dans le modèle du logiciel ERP. La troisième étape consiste à formaliser la règle entre ces modèles plate-forme : à tout objet de type composant standard dans la *nomenclature de conception* du logiciel PLM correspond un objet de type article avec une propriété acheté dans la *nomenclature pour fabrication* du logiciel ERP.

Dans le deuxième exemple, la première étape consiste à identifier les entités « composants élémentaires » pour le métier de la conception, les entités « composants élémentaires » pour le métier de la fabrication et la règle métier « S'il y a n fois le composant élémentaire A dans la *nomenclature de conception*, il doit y avoir n fois le composant élémentaire B correspondant dans la *nomenclature pour fabrication* ». La deuxième étape consiste à choisir des entités permettant d'implémenter cette règle. Nous choisissons de représenter l'entité « composant élémentaire » par les classes d'objet « composant standard » et « composant » dans le modèle personnalisé du logiciel PLM et l'entité « composant élémentaire » par une classe d'objet « article » avec une propriété ayant une valeur « acheté » ou « élémentaire » dans le modèle du logiciel ERP. La troisième étape consiste à formaliser la règle entre ces modèles plate-forme : « S'il y a n fois un objet A de type composant standard ou composant dans la *nomenclature de conception* du logiciel PLM, il doit y avoir n fois un objet B correspondant de type article avec une propriété acheté ou élémentaire dans la *nomenclature pour fabrication* du logiciel ERP ».

6.4.3 Exemples de mises en œuvre de règles de cohérence

Ce paragraphe illustre la mise en œuvre des règles de cohérence à partir de différents exemples.

Le premier exemple est une règle concernant le **support** de la construction de la *nomenclature pour fabrication* à partir de la *nomenclature de conception* via des correspondances entre les entités du modèle personnalisé du logiciel PLM et du modèle du logiciel ERP : à tout objet de type composant standard dans la *nomenclature de conception* du logiciel PLM correspond un objet de type article avec une propriété acheté dans la *nomenclature pour fabrication* du logiciel ERP. Un programme peut créer automatiquement les articles dans le logiciel ERP à partir d'un fichier export du logiciel PLM. Dans ce cas, la règle peut être implémentée dans ce programme d'import.

Le second exemple est une règle concernant le **support** de la construction de la *nomenclature pour fabrication* à partir de la *nomenclature de conception* via des correspondances entre les relations hiérarchiques et les quantités dans les structures. Dans cet exemple, nous prenons l'hypothèse que la structure de la *nomenclature pour fabrication* est créée en deux étapes. Tout d'abord, une copie de la structure de la *nomenclature de conception* est réalisée. Puis, cette structure est modifiée pour répondre aux besoins relatifs aux gestions d'approvisionnement, de la fabrication, des flux et des stocks dans l'atelier. Une règle de cohérence peut alors être appliquée lors de la modification de la *nomenclature pour fabrication* : si une relation avec une quantité q d'un composant X est enlevée, une ou plusieurs relations doivent être créées ou modifiées pour assurer que la quantité q du composant X est préservée. Cette règle peut être implémentée dans le module MPM (Manufacturing Process Management) du logiciel PLM par exemple.

Le troisième exemple est une règle concernant la **vérification** via des correspondances entre les relations hiérarchiques et les quantités dans les structures : s'il y a n fois l'élément élémentaire A dans la *nomenclature de conception*, il doit y avoir n fois l'élément élémentaire B correspondant dans la *nomenclature pour fabrication*.

6.5 Conclusion

L'objectif de ce paragraphe était d'illustrer comment un travail de modélisation basé sur le cadre proposé peut fournir des éléments de réponse au problème identifié.

Cette illustration a été réalisée pour la structure organique (nomenclature) du produit. Une perspective de ce travail est de réaliser une mise en œuvre pour la structure géométrique (notion de chaîne numérique, cf. chapitre 1, § 4.1.1) voire pour la structure des exigences spécifiant le produit.

Pour réussir à délivrer un « bon » produit, nous avons montré qu'il est nécessaire de créer une synergie entre les acteurs des différents métiers. Cette synergie est à la fois source d'agilité opérationnelle et de valeur ajoutée pour le produit. Cependant, il a été souligné que la maturité de l'entreprise s'acquiert pas à pas (cf. chapitre 3, § 4.1.3). Pour identifier les différents stades de maturité relatifs à notre problème, revenons sur les bases de l'efficacité de l'entreprise. L'efficacité d'une entreprise repose à la fois sur l'expertise de chaque métier et sur leur capacité à collaborer. Ainsi, le premier stade de maturité que nous identifions est d'être capable de définir un environnement de collaboration. Les modèles utilisés ont alors beaucoup de points communs. Une fois cette synergie créée, le stade suivant consiste à instaurer un maximum d'autonomie à chaque métier afin que ces derniers valorisent au mieux leurs compétences respectives. Ce deuxième stade consiste également à valoriser les synergies au travers de l'établissement de règles de cohérence. Les modèles utilisés alors prennent en compte les spécificités de chaque métier et les caractéristiques nécessaires pour gérer au mieux les interfaces.

7 Conclusion

L'hypothèse formulée dans le chapitre 2 était que l'utilisation d'un cadre de modélisation adapté constitue un élément de réponse pour réaliser une construction du système PLM en adéquation avec les deux buts de l'entreprise identifiés. Dans le chapitre 2, une ébauche du cadre a été proposée. Les objectifs de ce chapitre ont été de définir ce cadre de manière détaillée et d'illustrer son utilisation.

Tout d'abord, nous avons précisé nos problèmes au regard de la solution envisagée (§ 2).

Suite aux états de l'art des méthodes supportant la construction du système PLM (§ 3), nous avons pu positionner le cadre proposé au regard des travaux existants. Nous avons pu également identifier les mécanismes permettant d'assurer les interfaces entre le système PLM et d'autres systèmes.

Le paragraphe 4 a alors consisté à définir le cadre de modélisation de manière détaillée. Tout au long de la construction du cadre, des correspondances ont été établies avec le paradigme systémique. Il a été ainsi défini que le cadre dans son état actuel ne prend pas en compte tous les pôles d'observation nécessaires. Il faudra ainsi faire des travaux complémentaires pour le rendre complet au regard du paradigme. Une fois les axes redéfinis, un nouveau paradigme a alors été introduit : le paradigme d'ambivalence.

Pour mettre en application ce paradigme, nous avons fait le postulat que l'entreprise et le produit sont des réalités ambivalentes qui peuvent être considérées selon différentes perceptions. Ce paradigme considère que les perceptions d'une même réalité ne doivent pas être perçues de manière juxtaposée mais de manière simultanée, conjointe. Ainsi, l'ambivalence d'une réalité implique que ce qui s'applique à une perception s'applique à l'autre perception et inversement et qu'il est possible d'établir des correspondances entre les deux perceptions. Ces correspondances bidirectionnelles remplacent les relations de conformité unilatérale usuelles entre les perceptions. Elles permettent de passer du raisonnement réactif habituel à un raisonnement synergique, source d'agilité.

Les paragraphes 5 et 6 ont alors illustré l'utilisation du cadre pour nos deux problèmes. La conclusion tirée de ces paragraphes est que raisonner au travers de perceptions formalisées est indispensable mais que la synergie est avant tout le fruit d'un souhait de collaboration entre les différentes familles d'acteurs travaillant autour d'un but commun.

Pour aller plus loin dans la recherche de l'agilité opérationnelle et de synergies entre perception organisationnelle et perception informationnelle, une des perspectives que nous identifions est d'analyser les interactions possibles entre l'approche Lean (centrée perception organisationnelle) qui supporte l'agilité et l'approche PLM (centrée perception informationnelle) [Pinel et al., 11].

Dans le chapitre suivant, nous réalisons une mise en œuvre opérationnelle de notre cadre. L'objectif du chapitre n'est pas de valider le cadre mais de faire une étude de faisabilité de sa mise en œuvre avec un environnement de modélisation existant. En effet, valider le cadre signifierait être capable de prouver

qu'il convient aux problématiques de toutes les entreprises faisant des produits mécatroniques. Valider le cadre signifierait également être capable de prouver qu'il a une valeur ajoutée par rapport aux méthodes proposées dans les travaux existants. Ne pouvant amener ces preuves, nous avons fait le choix d'illustrer comment l'instrumentaliser. Nous réalisons ainsi l'expérimentation relative à la création du modèle métier et du modèle personnalisé du logiciel PLM et à la transformation de l'un à l'autre. Ce choix est justifié par le fait que ce sujet est peu explicité dans les travaux relatifs au PLM existants.

Chapitre 5

Expérimentation

Sommaire

- 1 Introduction
 - 2 *Sujet d'étude : métamodèles, modèles et transformations*
 - 3 *Choix d'un environnement de modélisation pour la mise en œuvre du cadre*
 - 4 *Sujet d'étude : nature des métamodèles, modèles et transformations*
 - 5 *Définition du métamodèle métier et exemple d'instanciation dans EMF*
 - 6 *Définition du métamodèle du modèle personnalisé d'un logiciel PLM dans EMF*
 - 7 Conclusion
-

1 Introduction

L'utilisation d'un cadre de modélisation guide le modélisateur dans la construction de ses modèles. Dans le paragraphe 6 du chapitre 4, nous avons illustré comment employer le cadre de modélisation proposé pour identifier des éléments de réponse au problème 2. Dans ce chapitre, notre objectif est d'expérimenter l'utilisation d'un environnement de modélisation avec lequel le modélisateur peut créer ses métamodèles et ses modèles. Notre choix s'est porté sur l'environnement de modélisation EMF (Eclipse Modelling Framework) [Eclipse, 13a]. Comme sujet d'étude, nous nous concentrons sur le passage entre le métamodèle du modèle métier PLM et le métamodèle du modèle personnalisé du logiciel PLM en focalisant sur l'aspect « données ». Ce chapitre s'intéresse plus particulièrement à la formalisation des métamodèles. La représentation du produit analysée est la représentation du produit de conception. Le logiciel PLM étudié est le logiciel Audros de l'éditeur Audros [Audros, 13].

Dans un premier temps, ce chapitre positionne notre sujet d'étude dans le cadre de modélisation (§ 2). Ce positionnement nous permet d'identifier les métamodèles, les modèles et les transformations qui doivent être mis en oeuvre dans l'environnement de modélisation. Il nous permet également d'identifier les transformations à instaurer entre l'environnement de modélisation et la base de données du logiciel PLM. Suite à cette définition, ce chapitre fournit une présentation rapide de l'environnement de modélisation EMF (§ 3). Ce chapitre identifie ensuite la nature des métamodèles, des modèles et des transformations induite par les choix d'EMF et du logiciel Audros (§ 4). Suite à cette présentation, ce chapitre présente l'élaboration du métamodèle du modèle métier PLM à partir de modèles métiers existants. Ce travail de construction consiste en un travail de métamodélisation. En effet, il implique d'identifier les entités et structure nécessaires pour construire les modèles métier PLM (§ 5). Le dernier paragraphe de ce chapitre définit le métamodèle de personnalisation du logiciel Audros dans un environnement de modélisation (§ 6). Ce travail consiste à traduire le métamodèle de personnalisation d'Audros, défini à partir d'un langage relationnel, en un métamodèle construit à partir du méta-métamodèle Ecore d'EMF.

2 Sujet d'étude : métamodèles, modèles et transformations

Dans ce chapitre, nous avons choisi de nous concentrer sur la création et la transformation entre le modèle métier PLM et le modèle personnalisé du logiciel PLM relatifs aux données. Les problèmes identifiés dans le chapitre précédent (chapitre 4, § 2.1.4) concernent la construction des modèles et la mise en place de mécanismes pour assurer la cohérence entre plusieurs modèles. Les éléments de solution identifiés pour répondre à ces problèmes consistent d'une part à formaliser les métamodèles du modèle métier PLM et du modèle personnalisé du logiciel PLM et, d'autre part, à identifier des

règles de transformation entre ces métamodèles (cf. chapitre 4, § 5). Ainsi, il a été proposé de construire un métamodèle du modèle métier PLM en prenant comme base de travail un modèle générique [Gzara, 00] que nous adaptions en fonction des résultats d'une étude réalisée au sein du Club PLM et de l'analyse d'autres modèles académiques (cf. chapitre 1, cf. § 5). Concernant le métamodèle de personnalisation du logiciel PLM relatif aux données, une démarche a été identifiée pour le construire à partir d'une rétro-conception de la base de données relationnelle du logiciel PLM. La définition de correspondances entre les entités et les structures des deux métamodèles permettent d'établir les règles de transformation.

Dans ce chapitre, nous choisissons de traiter la représentation de conception car c'est celle pour laquelle nous avons des retours d'expérience venant du Club PLM. Ainsi, la figure 4.1 positionne notre sujet d'étude dans le cadre de modélisation proposé dans le chapitre 4.

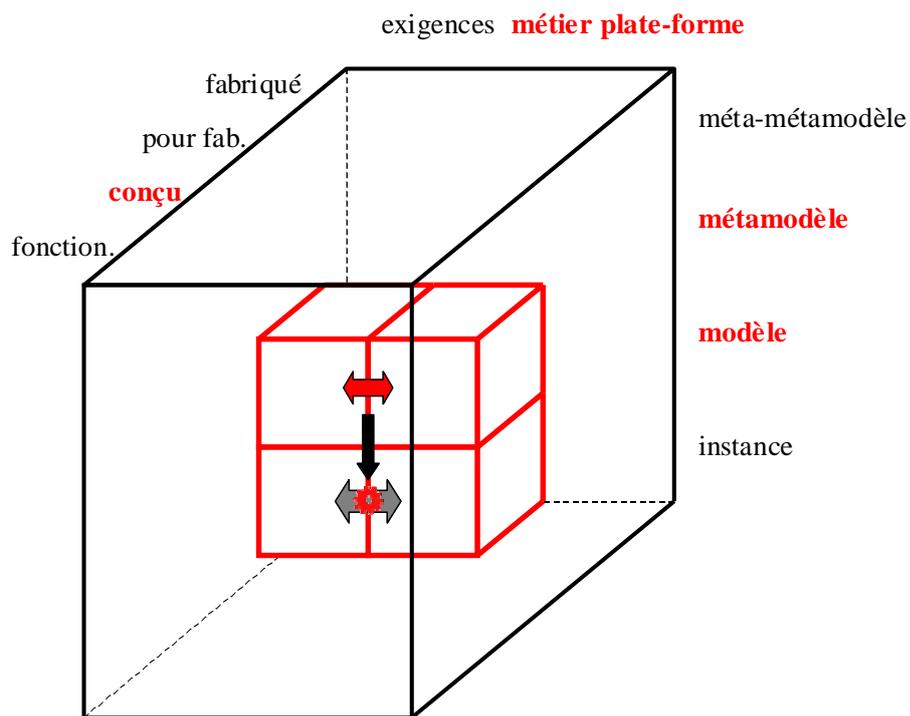


Figure 5.1 : Positionnement du sujet d'étude dans le cadre de modélisation

Pour mettre en oeuvre ce cadre, il est nécessaire d'utiliser un environnement de modélisation. Ainsi, deux espaces techniques sont identifiables (cf. figure 5.2). Un espace technique est défini comme l'ensemble des outils et techniques issus d'une pyramide de métamodèles dont le sommet est occupé par une famille de (méta)métamodèles similaires [Favre et al., 06]. Le premier espace technique de notre sujet d'étude est l'environnement de modélisation. Le deuxième est la base de données du logiciel PLM. Tout d'abord, les deux métamodèles sont créés dans l'environnement de modélisation et deux programmes sont définis pour décrire les transformations. Un premier programme est mis en oeuvre dans l'environnement de modélisation. Il identifie les règles de transformation entre le métamodèle du modèle métier PLM et le métamodèle du modèle personnalisé du logiciel PLM. Le deuxième programme est positionné entre les deux espaces techniques. Il définit les règles de transformation entre le métamodèle du modèle personnalisé du logiciel PLM décrit dans l'environnement de modélisation et le métamodèle du modèle personnalisé du logiciel PLM intégré dans le logiciel étudié. Selon le paradigme d'ambivalence, il est nécessaire de mettre en oeuvre les transformations de manière bidirectionnelle. Aujourd'hui, seule l'étude de faisabilité des transformations dans une direction a été faite.

Une fois les métamodèles et programme de transformation définis, l'entreprise peut créer son modèle métier PLM en instanciant le métamodèle. Le programme de transformation assiste alors la création du

modèle personnalisé du logiciel PLM. Le deuxième programme permet de créer le modèle personnalisé en décrivant la base de données du logiciel PLM.

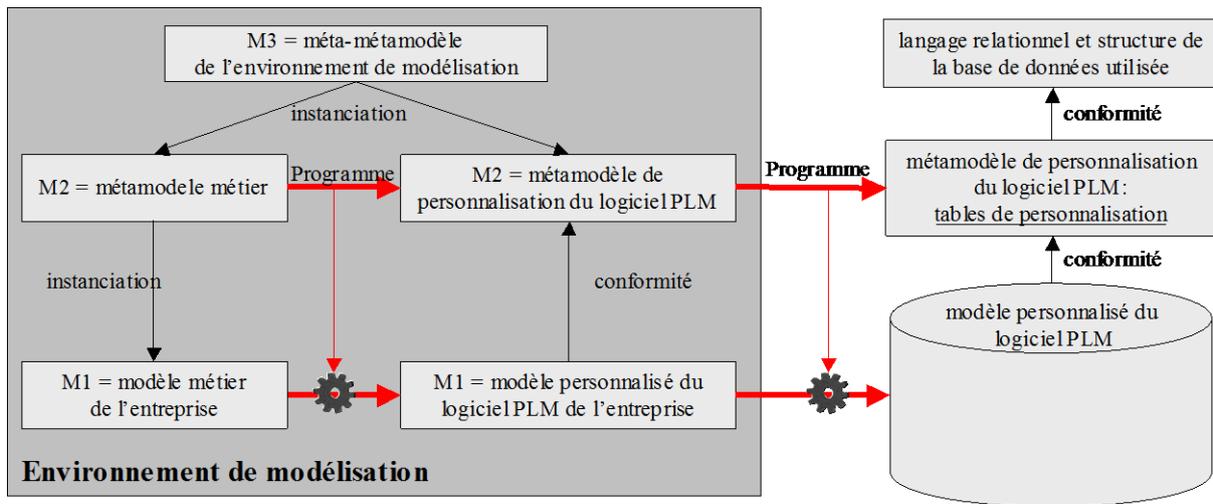


Figure 5.2 : Métamodèles, modèles et transformations de notre sujet d'étude

Dans le paragraphe suivant, nous faisons un état des lieux des différents environnements de modélisation à notre disposition et nous choisissons celui que nous utiliserons pour notre expérimentation.

3 Choix d'un environnement de modélisation pour la mise en œuvre du cadre

L'objectif de ce paragraphe est de choisir un environnement de modélisation pour outiller notre cadre de modélisation. Après avoir défini les fonctionnalités nécessaires et porté notre choix sur l'environnement EMF, nous présentons rapidement ce dernier.

3.1 Choix d'un environnement de modélisation

Pour mettre en œuvre le cadre de modélisation proposé, il est nécessaire que le (ou les) environnement(s) de modélisation utilisé(s) ait(ent) plusieurs fonctionnalités. Nous choisissons de les identifier non pas pour le traitement particulier de notre sujet d'étude mais pour une utilisation générale du cadre. Dans le chapitre 3, différents modèles relatifs aux fonctions du futur système PLM (hiérarchie d'objectifs, cas d'utilisation, exigences) ont été définis. Dans le chapitre 4, nous avons déterminé que le modèle métier PLM et le modèle personnalisé du logiciel PLM étaient relatifs aux aspects structurel (données) et comportemental (traitements) du système PLM.

Ainsi, les quatre principales fonctionnalités nécessaires sont :

- fonctionnalité 1 : mise en œuvre de mécanismes de cohérence entre les modèles représentant les aspects fonctionnel, structurel et comportemental du système PLM à concevoir et fabriquer ;
- fonctionnalité 2 : prise en compte des quatre niveaux de modélisation de l'architecture MDA (méta-métamodèle, métamodèle, modèle, instance) ;
- fonctionnalité 3 : mise en œuvre de mécanismes de transformation de modèles ;
- fonctionnalité 4 : interface conviviale dans laquelle les modèles sont créés sous forme de graphismes.

Parmi les environnements existants, ceux basés sur l'utilisation de langages standards (Win Design [Win Design, 13], Visual Paradigm [Visual Paradigm, 13], etc.) possèdent les fonctionnalités 1 et 4 mais pas les fonctionnalités 2 et 3. Les environnements basés sur la définition de métamodèles EMF [Eclipse, 13a], GME [Ledeczi et al., 01]) possèdent la fonctionnalité 2 partiellement et la fonctionnalité 3 de manière complète. Leurs graphismes (fonctionnalité 4) sont peu conviviaux.

Dans l'environnement EMF, la transformation des modèles est assurée via l'utilisation du langage de transformation ATLAS [Eclipse, 13b]. Pour avoir un graphisme convivial, les outils de GMF³⁸ peuvent être utilisés en complément d'EMF. Pour GME, il est nécessaire d'utiliser le logiciel GreAT³⁹ en complément de GME pour assurer la transformation de modèles.

Dans les environnements EMF et GME, la fonctionnalité 2 est assurée partiellement car ces logiciels travaillent sur trois niveaux de modélisation et non quatre. Ces trois niveaux correspondent usuellement au niveau méta-métamodèle, métamodèle et modèle. Pour définir des transformations relatives aux représentations du produit, les trois niveaux d'EMF ou GME peuvent être identifiés comme étant les niveaux métamodèle, modèle et instance [Iraqi-Houssaini et al., 11] (cf. chapitre 4, § 3.2).

Par conséquent, les environnements de modélisation basés sur la définition de métamodèles sont ceux qui correspondent le mieux à nos besoins. Pour assurer la cohérence entre les modèles décrivant les aspects fonctionnel, structurel et comportemental d'un système (fonctionnalité 1), il est nécessaire de créer les métamodèles correspondant à chacun de ces aspects dans l'environnement de modélisation choisi. Par exemple, pour décrire des processus avec le langage BPMN (Business Process Model and Notation), le métamodèle correspondant au langage BPMN doit être formalisé dans l'environnement de modélisation. Il est également nécessaire pour assurer la cohérence entre les modèles fonctionnel, structurel et comportemental de définir les relations entre les métamodèles utilisés.

Dans le cadre de notre étude, nous choisissons d'utiliser l'environnement de modélisation EMF d'Eclipse qui semble le plus adapté à notre approche⁴⁰.

3.2 L'environnement EMF

Créée en 2004, la fondation Eclipse est une organisation à but non lucratif qui supervise le développement de l'environnement intégré de développement open source Eclipse [Eclipse, 13c]. Cette fondation est constituée aujourd'hui d'environ 170 membres représentant la plupart des régions du monde ainsi que de nombreux secteurs industriels et technologiques. Eclipse trouve ses origines dans la création de l'environnement EMF (Eclipse Modelling Framework) [Eclipse, 13a] en 2001 par IBM.

3.2.1 Modélisation dans EMF

EMF travaille sur trois niveaux de modélisation : Ecore, un modèle construit à partir d'Ecore et des instances de ce modèle. Selon les besoins, Ecore peut être utilisé en tant que méta-métamodèle ou en tant que métamodèle. La figure 5.3 montre ces deux possibilités en reprenant les quatre niveaux de l'architecture MDA (cf. chapitre 2, § 2.3).

³⁸ <http://www.eclipse.org/modeling/gmp/>

³⁹ <http://www.isis.vanderbilt.edu/tools/GreAT>

⁴⁰ De plus, la communauté universitaire française utilisant ce logiciel est importante ce qui permet d'envisager de potentiels partenariats autour de ces travaux de métamodélisation.

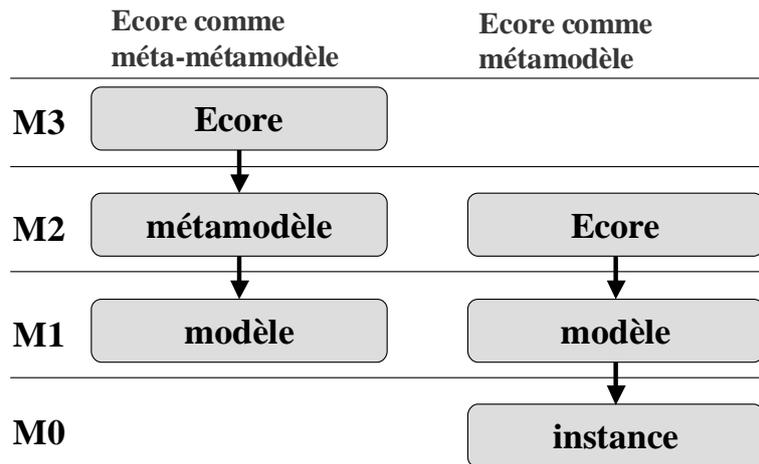


Figure 5.3 : Ecore comme méta-métamodèle ou comme métamodèle (adapté de [Vanwormhoudt, 12])

Dans ce chapitre, Ecore est utilisé comme méta-métamodèle. Ainsi, les métamodèles que nous créons sont des instances d'Ecore. Ecore a de fortes similarités avec MOF (chapitre 2, § cf. 2.3) mais il est plus simple. La figure 5.4 présente un extrait des entités d'Ecore. Ecore utilise des noms similaires à ceux utilisés dans MOF lorsqu'il existe des correspondances. Pour différencier les entités d'Ecore de celles du MOF, toute entité d'Ecore commence par « E ». Ainsi, aux « Class » de MOF correspondent les « **EClass** » d'Ecore, aux « Attribute » correspondent les « **EAttribute** »...

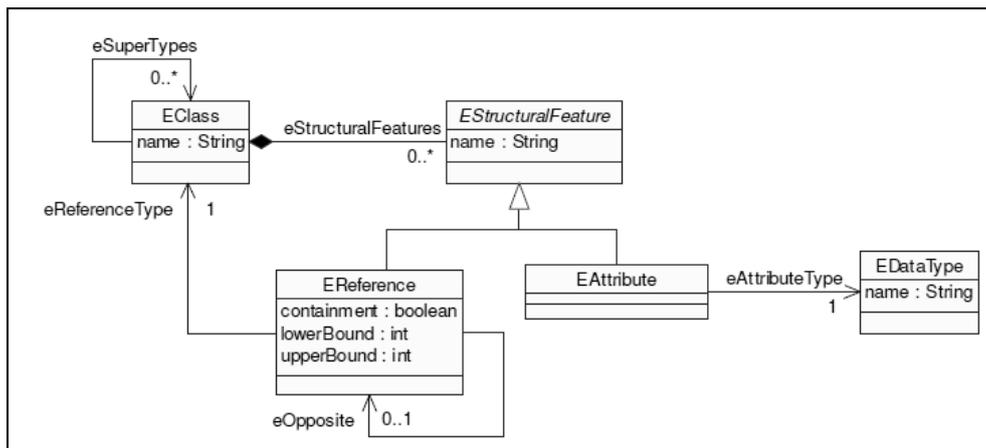


Figure 5.4 : Extrait des entités de Ecore [Steinberg, 08]

Une des principales différences entre Ecore et MOF est qu'il n'y a pas d'association dans Ecore. Ainsi pour créer une relation entre deux « EClass », un attribut, le « **EReference** », est utilisé sur une des deux « EClass ». Si la relation est bidirectionnelle, un EReference est utilisé dans les deux classes en précisant qu'elles sont en opposition grâce à la métadonnée eOpposite de la EReference.

Rappelons qu'un métamodèle est constitué d'éléments qui servent à construire les modèles. Par exemple, le métamodèle du langage UML contient des « classes », des « relations » et des « attributs ». La figure 5.5 illustre les différences entre les métamodèles construits à partir de MOF et ceux construits à partir d'Ecore. Dans l'exemple, nous utilisons deux entités, « entité » et « propriétés », reliées par une relation « est décrite par ».

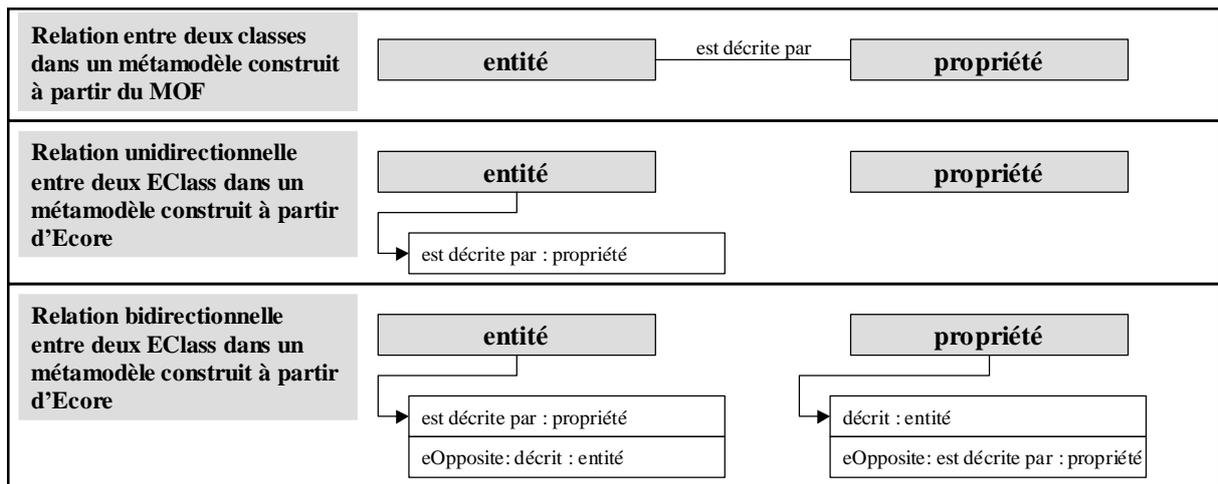


Figure 5.5 : Différences de formalisation d'une relation entre deux classes avec le MOF et avec Ecore

Pour créer les métamodèles à partir d'Ecore, EMF propose une forme arborescente (l'extension du métamodèle créé est alors .ecore) et une forme graphique (l'extension du métamodèle créé est alors .ecorediag). La figure 5.6 illustre ces deux formes au travers de notre exemple. Dans la figure, on retrouve un exemple de EClass, un exemple de EAttribute et un exemple de EReference. La forme graphique donne l'impression qu'il existe deux EClass et une relation entre elles. Comme le montre la forme arborescente, le métamodèle contient en réalité deux EClass et l'EClass « article » contient une EReference.

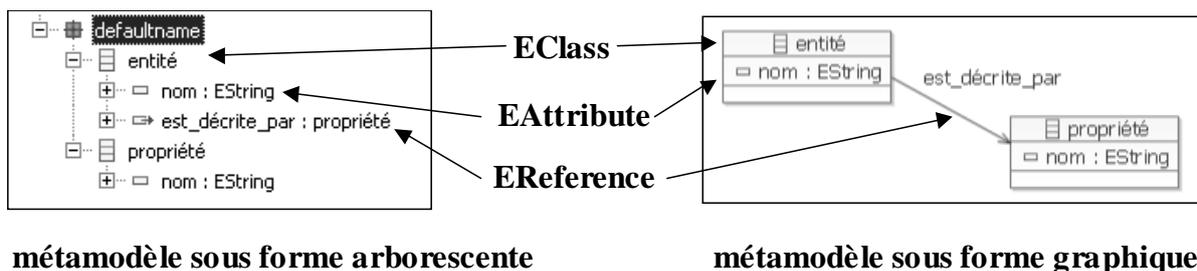


Figure 5.6 : Forme arborescente et forme graphique dans EMF

Une conséquence de cette manière de créer les relations est qu'il n'est pas possible de créer de relation n-aire avec EMF. Pour rappel, dans le langage UML, une relation n-aire représente une relation multiple. Dans l'exemple de la figure 5.7, une classe « relation » est reliée à deux classes « entités » par une relation n-aire. Pour modéliser les relations n-aire à partir d'Ecore, il est nécessaire d'utiliser une EClass supplémentaire. Dans l'exemple de la figure 5.7, l'EClass « association » remplace la relation n-aire.

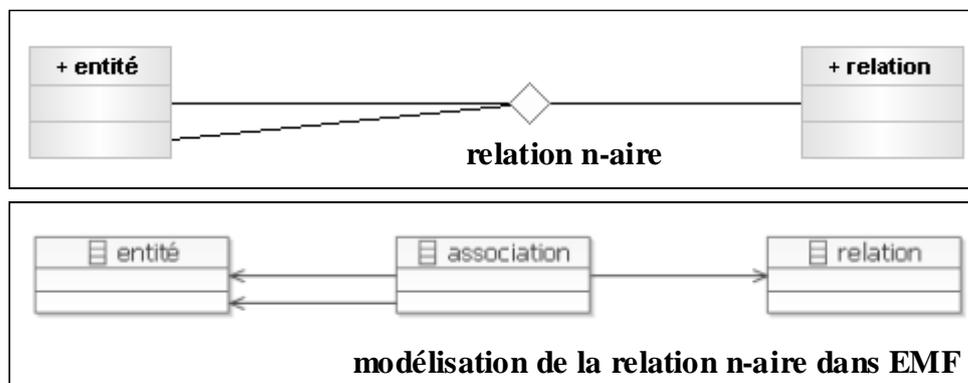


Figure 5.7 : Modélisation d'une relation n-aire avec EMF

Concernant les héritages, ils sont définis via une propriété « **ESuper Types** » dans les EClass. Par exemple, on définit qu'une EClass « cahier des charges » hérite des caractéristiques de la EClass « document » déclarée comme sa ESuper Type.

Comprendre la manière dont sont définies les structures dans EMF est essentiel afin de pouvoir par la suite établir des règles de correspondances entre les métamodèles.

EMF supporte l'importation et l'exportation de trois formats : UML, XMI et Java. Ecore ayant de fortes similarités avec MOF, il est capable d'importer et d'exporter des modèles construits à partir du langage **UML**. Si ces transferts sont réalisables avec le logiciel IBM Rational, leur mise en œuvre peut s'avérer plus délicate avec d'autres logiciels. Les métamodèles créés à partir d'Ecore étant décrits en **XMI** (XML Metadata Interchange), ils sont nativement compatibles avec ce langage. Ce langage constitue un standard de l'OMG utilisable pour échanger les métadonnées via XML (eXtensible Markup Language). Enfin, une des fonctionnalités d'EMF consiste à générer du code **Java** à partir d'un modèle. L'environnement EMF est ainsi également compatible avec ce langage. La figure 5.8 illustre ces diverses importations et générations.

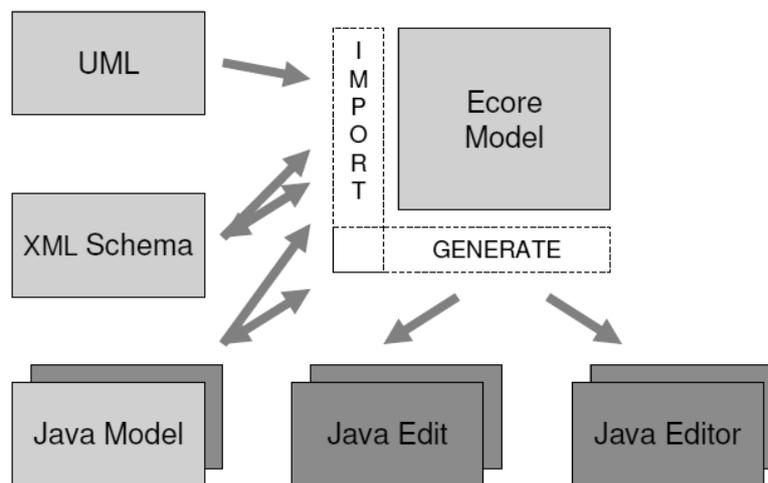


Figure 5.8 : Importations et générations possibles à partir d'Ecore [Steinberg, 07]

3.2.2 Transformation de modèles avec EMF

ATL [Eclipse, 13b], ou Atlas Transformation Language, est un langage de transformation de modèles. Pour assurer la transformation d'un modèle, ATL s'appuie sur la définition de correspondances entre les entités de métamodèles construits à partir d'Ecore. La figure 5.9 illustre un exemple de programme ATL. La définition de cette transformation nécessite d'identifier un métamodèle de départ (« IN » ou « source ») et un métamodèle d'arrivée (« OUT » ou « target »). Dans l'exemple, `metamodelemetier.ecore` est le métamodèle de départ et `metamodelaudros.ecore` est le métamodèle d'arrivée. Puis, des correspondances sont identifiées entre les entités des métamodèles. Dans l'exemple, l'EClass « element » correspond à l'EClass « classe_d_objets ».

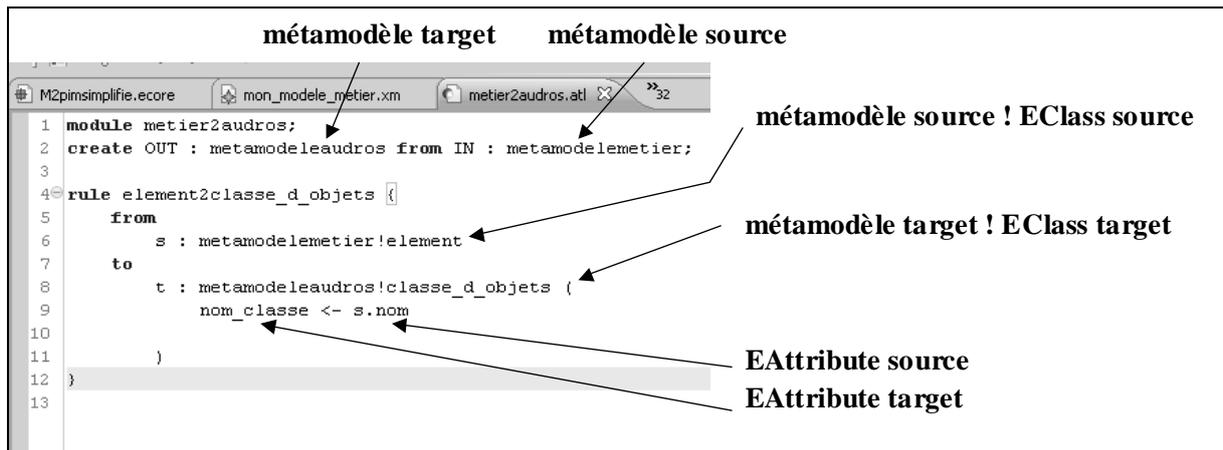


Figure 5.9 : Exemple de programme ATL

Grâce aux correspondances définies, un modèle instance du métamodèle d'arrivée peut être créé automatiquement à partir d'un modèle instance du métamodèle de départ. La figure 5.10 présente un exemple de mise en œuvre d'une transformation ATL. Un modèle a été créé à partir du métamodèle métier. Ce modèle est au format XMI : mon_modele_metier.xmi. En identifiant dans l'interface le programme ATL, les deux métamodèles, le modèle de départ et le nom et l'emplacement du modèle d'arrivée, ce dernier est créé automatiquement.

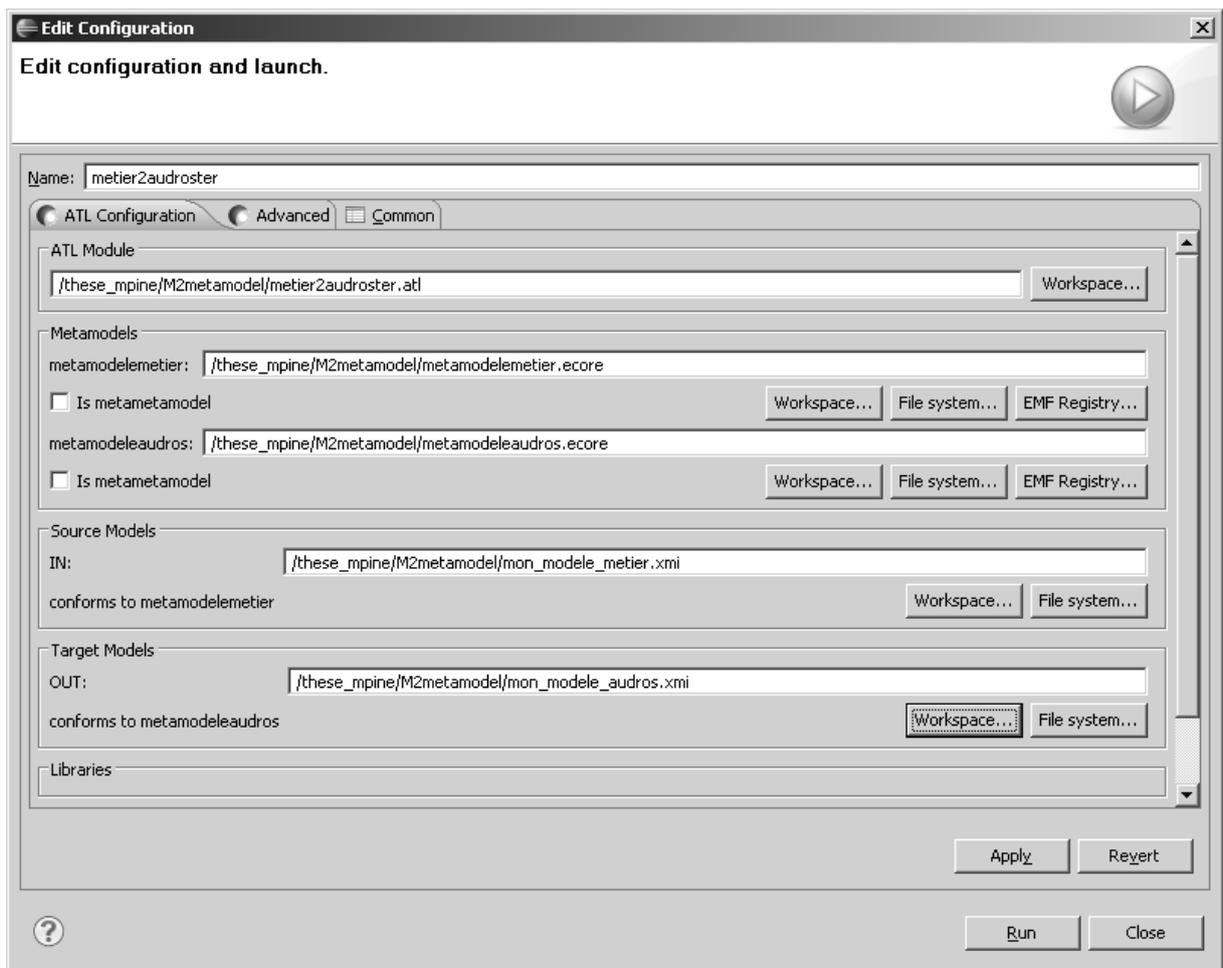


Figure 5.10 : Exemple de mise en œuvre d'une transformation ATL

Il est à noter que, durant la transformation, aucun message d'alerte n'avertit d'erreurs possibles. Ainsi, l'adéquation entre les règles de transformations et les entités, structures et contraintes des

métamodèles est garantie par le modélisateur. Avec les programmes ATL, il n'est également pas possible de réaliser des choix au cours de la transformation. Si des choix sont à réaliser, il est nécessaire de compléter le modèle de départ avant de lancer la transformation. Par exemple, une EClass « A » du métamodèle de départ peut correspondre selon des règles non généralisables à une EClass « B » ou à une EClass « C » du métamodèle d'arrivée. Avant de lancer la transformation, une indication peut être mise manuellement dans l'entité instance de « A » du modèle de départ pour définir si la transformation est souhaitée vers une entité « B » ou vers une entité « C » du modèle d'arrivée.

3.3 Conclusion

Ce paragraphe a montré que deux catégories d'environnement de modélisation sont envisageables pour la mise en œuvre de notre cadre. La première est basée sur l'utilisation de langages standards. La deuxième est basée sur la définition de métamodèles. Seule cette deuxième catégorie est capable de mettre en œuvre des mécanismes de transformations de modèles. Notre choix s'est ainsi porté sur l'environnement EMF qui fait partie de cette deuxième catégorie. Dans les paragraphes suivants, nous illustrons la création des métamodèles du modèle métier PLM et du modèle personnalisé du logiciel PLM avec cet environnement.

4 Sujet d'étude : nature des métamodèles, modèles et transformations

Suite au choix de l'environnement de modélisation EMF et du logiciel PLM Audros, nous sommes capables d'identifier la nature des métamodèles, modèles et transformations pour notre sujet d'étude (cf. figure 5.11).

Dans l'espace technique relatif à EMF, les métamodèles sont des fichiers en format ECORE et les modèles sont des fichiers en format XMI. La première transformation de modèles est faite à partir d'un programme ATL.

L'installation du logiciel Audros que nous utilisons est réalisée avec une base de données Oracle. L'écriture dans les tables de la base se fait à partir de requêtes en langage SQL (Structured Query Language). La personnalisation du logiciel Audros implique d'écrire des données dans les tables de personnalisation et de créer des tables de stockage de données. Une des spécificités d'Audros est que l'éditeur a développé un langage spécifique pour la mise en œuvre de développements spécifiques : l'AUDros Programing Language ou AUPL.

La deuxième transformation de modèles consiste à transformer un modèle personnalisé défini en XMI dans EMF en un modèle personnalisé défini en données dans les tables de personnalisation. Cette transformation inclut également la création de nouvelles tables dans la base de données. Pour cette transformation, nous utilisons un programme AUPL.⁴¹

⁴¹ Ce programme est basé sur un programme existant fourni par l'éditeur qui permet de créer de manière automatique le modèle personnalisé du logiciel Audros à partir d'une définition du modèle dans un fichier EXCEL.

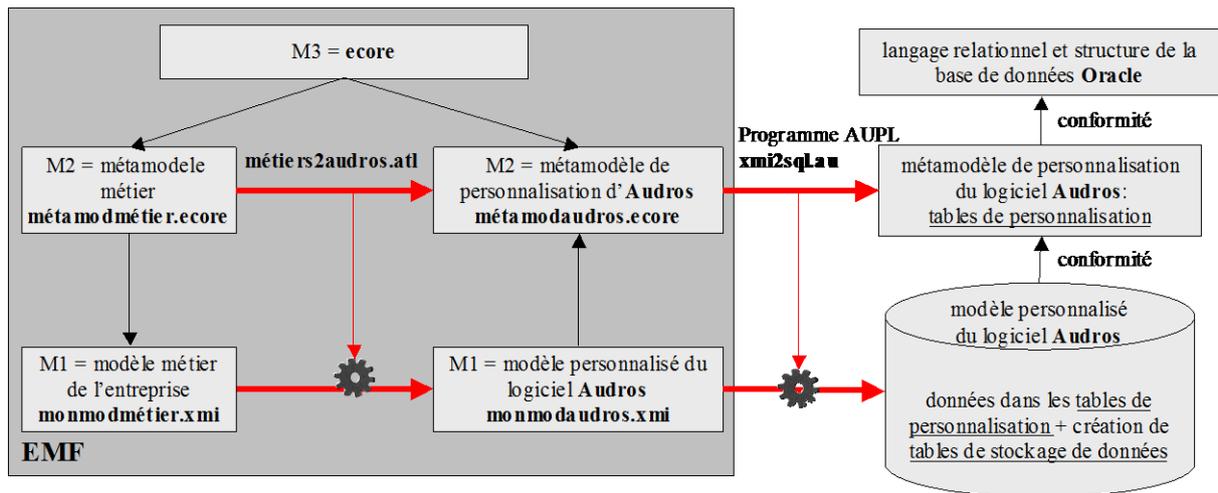


Figure 5.11 : Nature des métamodèles, modèles et transformations de notre sujet d'étude

5 Définition du métamodèle du modèle métier PLM et exemple d'instanciation dans EMF

L'objectif de ce paragraphe est de définir un métamodèle du modèle métier PLM dans l'environnement EMF et d'expérimenter l'instanciation de ce métamodèle. Nos travaux n'ont pas pour objectif de créer un métamodèle du modèle métier PLM complet. Leur but est de montrer comment construire un métamodèle du modèle métier PLM en illustrant la démarche par des points spécifiques aux systèmes PLM. Le métamodèle est un support à la construction du modèle métier PLM du système PLM d'une entreprise.

Pour définir le métamodèle du modèle métier PLM, nous nous appuyons sur une démarche basée sur trois étapes. La première étape analyse les entités et la structure des modèles métiers PLM utilisés par les membres du Club PLM (§ 5.1). La deuxième étape intègre les résultats de l'étape précédente dans un modèle métier PLM existant [Gzara, 00] et de trois autres modèles [Eynard, 06] [Le Duigou, 10] [Paviot, 11]. Cette étape nous permet d'identifier les entités de notre métamodèle métier PLM (§ 5.2). La troisième étape définit la structure du métamodèle dans EMF (§5.3). Enfin, ce paragraphe présente une illustration de la création d'un modèle métier PLM à partir du métamodèle défini (§5.4).

5.1 1^{ère} étape : étude terrain des modèles métiers de membres du Club PLM

5.1.1 Objectif de l'étude

L'objectif principal de cette étude est d'identifier les caractéristiques que le métamodèle du modèle métier PLM doit posséder pour répondre aux besoins industriels. Pour cela, nous analysons les modèles métiers de différents industriels utilisant des logiciels PLM. Cette analyse permet d'identifier des points communs et des différences entre les modèles métiers PLM étudiés. Dans cette étude, les profils d'entreprises sont variés et les logiciels PLM utilisés sont Audros, Teamcenter et Windchill. L'objectif secondaire de l'étude est d'identifier les grandes tendances des modèles métiers PLM utilisés.

5.1.2 Résultats de l'étude

Notre étude a été réalisée à partir de quatre sources. La première source est constituée d'interviews basées sur un questionnaire⁴². La deuxième source est constituée d'une étude de présentations réalisées

⁴² Ces interviews ont été réalisées en mars-avril 2011 auprès de cinq entreprises du Club PLM.

au sein du Club PLM par deux sociétés⁴³. La troisième source est constituée d'une interview informelle d'une entreprise du Club PLM. Enfin, la quatrième source est constituée d'échanges réalisés au sein du Club PLM durant une réunion consacrée aux modèles métiers mis en œuvre dans des logiciels PLM⁴⁴.

Il est à noter que dans tous les cas étudiés, le modèle d'exigence est défini sous la forme d'un cahier des charges adressé au fournisseur du logiciel (éditeur ou intégrateur). Le modèle métier PLM et le modèle personnalisé du logiciel PLM sont très similaires étant donné que le modèle métier PLM est défini une fois que le logiciel PLM a été choisi. Ce modèle métier PLM intègre donc les spécificités (qu'elles soient des contraintes ou des opportunités) du logiciel choisi ; il est formalisé avec des outils de bureautique.

Suite à cette étude, trois points communs et six différences majeures entre les modèles ont été mis en lumière. Ces éléments sont formalisés par des concepts du langage UML : « classes », « attributs », « objets » et « relations ».

Le premier point commun entre les modèles concerne **la nature des produits**. Tous les modèles étudiés servent à gérer des représentations de produits mécaniques ou mécatroniques.

Le deuxième point commun concerne **les attributs**. Dans tous les modèles, la référence, la version, la désignation et le statut d'une classe sont des attributs particuliers. Les autres attributs, spécifiques à chaque entreprise, servent à définir des propriétés du produit. Ces propriétés facilitent les recherches et elles peuvent être exploitées pour générer des documents ou des sites Internet de manière automatique. Les valeurs associées aux attributs, la matière d'un produit par exemple, peuvent être contenues à l'intérieur d'un fichier, un plan par exemple. L'information existe alors de manière redondante dans le système PLM. L'attribut « statut » a des propriétés particulières. D'une part, ses valeurs sont pré-définies dans le modèle. Par exemple, l'administrateur définit que les valeurs possibles pour un statut sont « en création », « valide », « obsolète »... D'autre part, une valeur, « valide » par exemple, peut être attribuée à plusieurs classes, les « plans », les « cahier des charges », etc.

Le troisième point commun concerne **les relations**. Dans les modèles étudiés, il existe plusieurs types de relations et chacune a une sémantique bien particulière : « est documenté par », « est composé de », « relation de nomenclature CAO »... Une même relation, « est documenté par » par exemple, peut servir à structurer plusieurs classes. Elle peut relier une classe « produit » à une classe « cahier des charges » et la même classe « produit » à une classe « norme » par exemple. De plus, les relations peuvent porter des attributs. Par exemple, la relation de nomenclature porte l'attribut « quantité ».

La première différence concerne **les documents utilisés** pour représenter le produit de conception et **leur format**. Dans le cadre d'une offre à l'affaire, les entreprises interviewées sont sous-traitantes d'un donneur d'ordre et elles assurent la conception du produit. Ces entreprises créent un plan interne et un plan destiné au client. Le plan client est un document pour lequel chaque information est contractuelle. Il porte ainsi le minimum d'informations possible. Si le produit est un sous-ensemble du produit final, le client fournit le cahier des charges fonctionnel du sous-ensemble mais également la définition de l'environnement sous forme de modèle CAO 3D. Concernant les formats des fichiers, les fichiers natifs sont souvent doublés par des fichiers neutres qui sont ceux partagés (le format natif n'étant disponible que pour le métier qui l'a créé). Des « doublons » (données gérées dans la base de données et dans un fichier) sont également réalisés pour les échanges avec l'environnement. Par exemple, la nomenclature est gérée dans la base de données au sein de l'entreprise mais elle est générée sous forme de document pour les échanges avec les fournisseurs et les clients.

La deuxième différence concerne **la spécialisation du modèle**. Les classes utilisées sont différentes d'une entreprise à une autre. D'une part, les notions d'ensemble et de composant peuvent être dissociées (une classe « ensemble » et une classe « composant ») ou non (tout est « article »). Les composants achetés sont souvent gérés dans une classe à part (« composant acheté »). Lorsqu'il y a plusieurs sortes de plan pour un même produit (plan client, plan interne, etc.), tous ces plans peuvent être gérés dans une même classe « plan » ou dans plusieurs classes spécifiques « plan client », « plan

⁴³ Ces présentations se sont déroulées en septembre 2010 et en janvier 2011

⁴⁴ Cette réunion s'est déroulée le 23/06/2011.

interne », etc. D'autre part, certains membres du Club PLM utilisent des classes spécifiques à leurs produits et à leurs constituants avec des attributs bien particuliers. Par exemple, dans le cas d'une manufacture de montres, le modèle peut être constitué de classes « cadran », « bracelet », « mouvement », etc. Le fait de spécialiser ou non le modèle à ses produits dépend de plusieurs facteurs. *Le premier facteur* est la nature de l'offre. Pour une offre catalogue, le « terrain » est favorable car l'entreprise a une logique de gamme de produits. Le produit est décliné sous forme de variantes ayant des rôles précis dans la gamme : le produit-vedette, le produit d'appel, le produit suiveur, produits haut de gamme et bas de gamme [Cossette, 06], [Mayrhofer, 06], [Vernette, 91]. Pour une offre à l'affaire, des familles de produits sont identifiables si l'entreprise est spécialisée dans la conception et fabrication de produits particuliers (par exemple, des roulements). Par contre, si l'entreprise a basé son offre sur la réponse à une fonction, cela peut être plus compliqué. En effet, aucune famille de produits n'est facilement identifiable. *Le deuxième facteur* est le besoin de réutilisation de l'existant. La réutilisation est plus facile à mettre en œuvre au travers d'une conception dite « modulaire » lorsque le produit est un assemblage et que l'offre est basée sur un catalogue. Les nouveaux produits sont alors souvent construits sur la base d'un produit existant avec d'éventuelles évolutions sur un élément. Dans le Club PLM, certaines d'entreprises ont des produits avec un, deux, trois composants. Dans ces cas-là, seuls les « choix » de conception sont réutilisables. *Le troisième facteur* est la diversité des produits au sein de l'entreprise. Plus une entreprise est importante, plus elle fait des choix génériques au niveau de son modèle afin que ce dernier réponde aux besoins de tous.

La troisième différence concerne **la séparation ou non du modèle 3D CAO et de la notion d'article**. Quand un assemblage 3D CAO est stocké dans un logiciel PLM, la nomenclature est automatiquement créée. Dans le Club PLM, ceux qui utilisent un même objet pour le modèle 3D CAO et l'article sont ceux pour qui l'intégrité de l'assemblage CAO a une grande importance. Adopter ce choix n'est cependant pas possible dans le cas de produits mécatroniques pour lesquels les parties informatiques du produit n'ont pas de représentation CAO.

La quatrième différence concerne **le regroupement ou la dissociation de différents fichiers dans un même objet**. *Le premier regroupement* est celui du fichier natif et du fichier neutre correspondant. Ce regroupement est fait par la majorité des membres du Club PLM. Par exemple, le fichier natif issu de la CAO et le fichier en format neutre PDF correspondant sont encapsulés dans un même objet « plan ». Cependant, certains membres du Club PLM préfèrent attribuer chaque fichier à un objet différent. Dans ce cas, un objet « plan format natif » encapsule le fichier natif issu de la CAO et un objet « plan format neutre » encapsule le fichier neutre en format PDF. Ce choix est fait pour séparer les objets manipulés par tous (« plan format neutre ») des objets spécifiques au Bureau d'Etudes (« plan format natif »). *Le deuxième regroupement* concerne le modèle 3D CAO et le plan correspondant. La majorité des membres du Club PLM sépare ces deux notions avec un objet qui encapsule le modèle 3D et un objet qui encapsule le plan. Cependant, trois membres ont fait le choix d'encapsuler le modèle 3D et le plan associé à l'intérieur d'un même objet.

La cinquième différence concerne **la gestion des évolutions**. Les différences entre les modèles sont fortement influencées par le logiciel PLM utilisé. Certains membres du Club PLM conservent pour une même version d'un document les différents enregistrements du document. A l'inverse, d'autres membres ne stockent dans le logiciel PLM que l'enregistrement « en-cours » (chaque enregistrement écrase le précédent). Concernant les versions, certains utilisent des objets « master » qui regroupent toutes les versions d'un objet alors que d'autres ne gèrent que des objets versionnés. La figure 5.12 illustre cette différence au travers de l'exemple de la nomenclature d'une table. Dans cette figure, les liens de nomenclature sont définis en rouge. Lorsque des objets « master » sont utilisés, les structures sont construites entre l'objet A versionné et l'objet B master (à gauche dans la figure) ou entre l'objet A versionné et l'objet B versionné (à droite dans la figure). Dans le cas où la structure est construite entre un objet A versionné et un objet B master, la version applicable de l'objet B est définie par le statut. Dans notre exemple, la version du pied applicable est la version 3 car c'est la seule en statut « valide » (les versions 1 et 2 étant « obsolètes »).

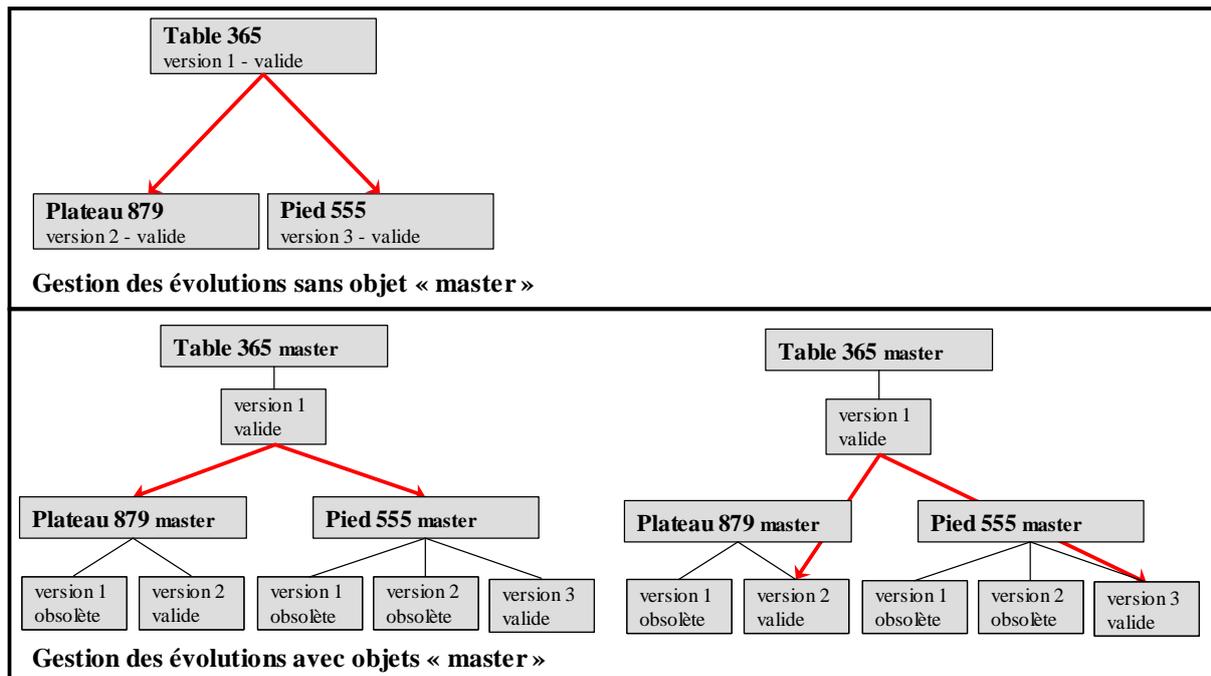


Figure 5.12 : Les trois manières de gérer les évolutions avec un système PLM

5.1.3 Conclusions de l'étude

Suite à cette étude, trois points communs et six différences majeures ont été identifiés entre les modèles étudiés.

Les modèles analysés ne traitent que de produits mécaniques ou mécatroniques. Ainsi, les conclusions que nous pouvons tirer de cette étude ne sont valables que pour les produits de ces natures. Pour élargir les conclusions à des produits chimiques par exemple, des études complémentaires sont nécessaires.

Cette étude a mis en lumière deux choix importants auxquels sont confrontés les industriels. Le premier consiste à choisir de baser la nomenclature du produit sur la nomenclature CAO (modèle appelé « CAD centric ») ou sur une nomenclature d'articles indépendants de la CAO (modèle appelé « item centric »). Le deuxième consiste à choisir une des trois manières identifiées pour gérer les évolutions.

L'étude a également montré deux formes de redondances dans les systèmes PLM. La première redondance est due à la création de fichiers en format standard en complément des fichiers en format natif. La deuxième est due à la présence d'une même information à l'intérieur d'un fichier et en tant qu'attribut d'un objet. La mise en œuvre de mécanismes pour maîtriser ces redondances est ainsi nécessaire. Il a aussi été mis en lumière qu'à un objet ne correspond pas forcément un fichier et réciproquement. Ainsi, plusieurs fichiers peuvent être encapsulés dans un même objet et un objet peut n'encapsuler aucun fichier.

5.2 2^{ème} étape : identification des entités du métamodèle métier PLM

Dans ce paragraphe, nous confrontons un modèle métier PLM [Gzara, 00] avec les résultats de l'étude réalisée auprès du Club PLM et avec trois autres modèles académiques [Eynard, 06] [Le Duigou, 10] [Pavio, 11]. L'objectif de cette étape est d'identifier les entités et les structures du métamodèle du modèle métier PLM utilisé dans notre cadre de métamodélisation.

Un métamodèle définit les entités qui servent à construire les modèles. Dans notre cas, trois sortes d'entités comparables au métamodèle d'UML vont exister : des « classes », des « relations » et des « attributs ». Les modèles PLM existants ne définissent pas les opérations des classes, ils sont

« statiques ». Les opérations ne sont pas prises en compte dans cette étude, elles seront intégrées ultérieurement. Suite à l'étude réalisée auprès du Club PLM, nous considérons que le métamodèle du modèle métier PLM doit intégrer des entités « fichier » et des entités « information contenue dans un fichier ». Ces entités permettent de mettre en lumière les redondances, d'identifier clairement sous quel format sont encapsulés les fichiers dans un objet et d'identifier les interactions entre les objets encapsulant des fichiers CAO.

Concernant les classes, les industriels utilisent des classes « master » avec des classes « versions associées au master » ou ils utilisent des classes « versionnés ». Dans les modèles académiques, le système PLM s'appuyant sur des classes « master » est privilégié mais certains travaux proposent des classes « versionnés » [Eynard, 06]. Notre métamodèle doit prendre en compte ces deux possibilités, il contient donc trois sortes de classes : *classe « master »*, *classe « versions associées au master »* et *classes « versionnées »*. Parmi les classes identifiées, *article* et *document* sont communs à tous les modèles, la classe *dossier* est présente dans un modèle [Gzara, 00]. Ces trois éléments sont définis dans notre métamodèle et ils sont regroupés sous la classe *élément*.

Concernant les relations, certains modèles étudiés proposent une relation d'agrégation pour définir les nomenclatures ; il n'y a pas de factorisation de la quantité [Gzara, 00] [Le Duigou, 10] [Eynard, 06]. Si un ensemble A est composé de quatre composants B, quatre relations existent. Un autre modèle propose également une relation d'agrégation mais associée à une classe-association avec une propriété définissant la quantité [Paviot, 11]. Les modèles industriels utilisent une relation d'association avec une classe-association pour déterminer la quantité. Dans notre métamodèle, nous définissons ainsi une entité *relation* à laquelle peuvent être associés des attributs. [Paviot, 11] propose des relations spécifiques entre les classes encapsulant les fichiers CAO : les relations de « CAD dependency ». Nous reprenons cette terminologie en proposant dans notre métamodèle trois sortes de relations : les *relations documentaires*, les *relations de dépendance CAO*, les *relations de structure produit non CAO*.

Dans les modèles industriels étudiés, une même relation, « est composé de » par exemple, peut servir à structurer plusieurs entités. Elle peut relier une entité « produit » à une entité « sous-ensemble », la même entité « sous-ensemble » à elle-même et la même entité « sous-ensemble » à une entité « composant standard ». Une entité « relation » est donc nécessaire dans notre métamodèle. La figure 5.13 montre qu'elle devrait être liée par une relation n-aire à l'entité « article ».

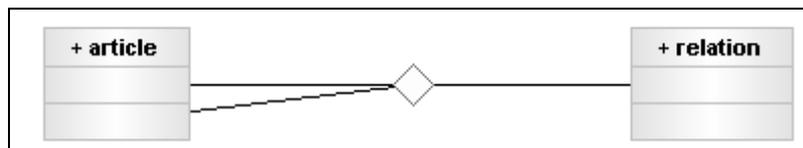


Figure 5.13 : Relation n-aire entre l'entité « article » et l'entité « relation »

Comme nous l'avons vu dans le paragraphe 3.2, EMF ne permet pas de modéliser les relations n-aire. Il est ainsi nécessaire de créer une **nouvelle entité « association »** (terme utilisé dans le métamodèle du modèle personnalisé du logiciel Audros). Au moment de l'instanciation, cette entité regroupe les identifiants des deux « articles » et de la « relation ».

Pour l'exemple de la figure 5.14, « produit2sous_ensemble » contient les identifiants de « produit », « sous-ensemble » et « est composé de ».

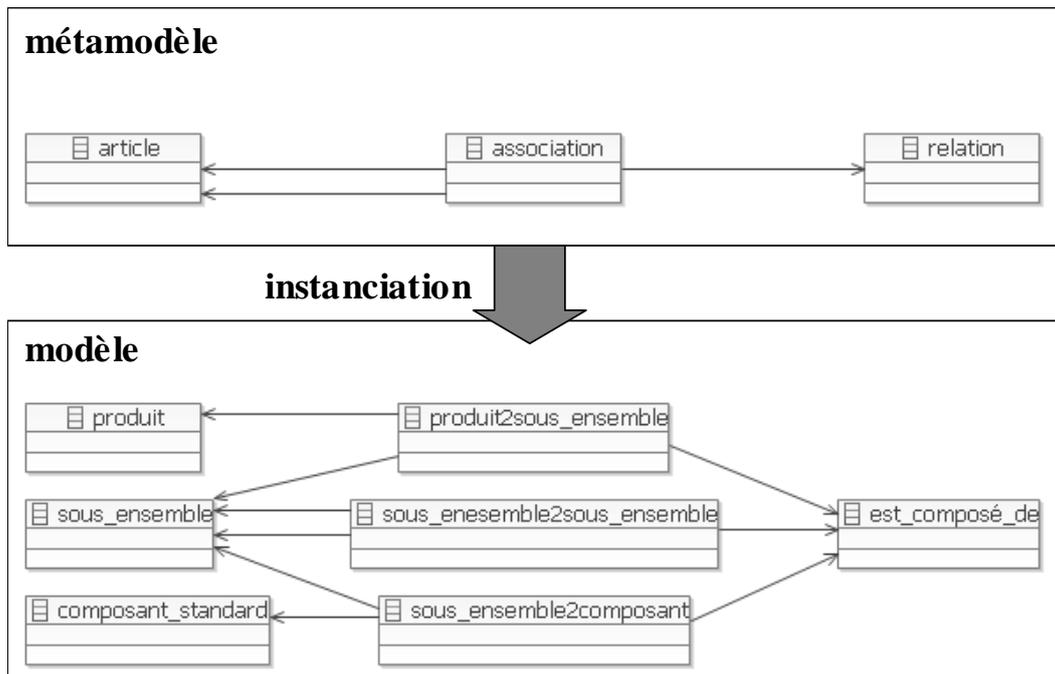


Figure 5.14 : Les entités article, association et relation de notre métamodèle métier

Concernant les attributs, certains modèles ne proposent que peu, voire pas d'attributs [Gzara, 00] [Le Duigou, 10] [Paviot, 11]. Un autre modèle propose les mêmes attributs que ceux identifiés dans les modèles industriels : *la référence*, *la version*, *la désignation* et *le statut* [Eynard, 06]. Nous définissons ainsi ces trois entités dans notre métamodèle et une entité *attributs spécifiques* pour les autres attributs. L'identification des différents statuts possibles est réalisée par une **entité à part**, « état » ([Gzara, 00] [Le Duigou, 10]).

Concernant les fichiers et leurs formats, la taille du fichier [Gzara, 00] et le nom du fichier [Eynard, 06] sont définis comme les propriétés d'une classe. Pour représenter de manière plus explicite et riche les relations entre objet et fichier encapsulé, nous proposons de déterminer une entité *fichier* pour laquelle il est possible de définir des attributs. Ces attributs permettent entre autres d'identifier le format du fichier : DWG, ASM, PDF, ... Les fichiers CAO ayant des particularités, nous déclinons l'entité fichier en *fichier natif CAO*, *fichier natif non CAO* et *fichier neutre*. Une relation est utilisée pour représenter la correspondance entre le fichier en format natif et le fichier au format neutre correspondant.

Enfin, une dernière entité définit les **informations contenues dans les fichiers**. L'utilité de cette entité est de représenter la redondance entre certains attributs de l'objet et certaines informations contenues dans le fichier encapsulé. Dans les modèles du Club PLM, cette redondance est souvent faite pour les références, version et statut d'un plan. Ces informations sont ainsi à la fois contenues dans le fichier du plan et à la fois définies comme attributs de l'objet encapsulant le plan.

Le tableau 5.1 présente un récapitulatif des différentes entités identifiées. Il établit également des correspondances avec les entités du modèle utilisé comme base de notre proposition [Gzara, 00]. Une de nos perspectives est de compléter ce tableau en y intégrant les correspondances avec les entités du protocole d'application 214 de la norme STEP [ISO, 10].

entité du métamodèle proposé	entité du modèle de Lilia Gzara
article master	article virtuel
version de l'article master	article virtuel versionné
article versionné	
document master	document
version du document master	document versionné
document versionné	
dossier master	dossier
version du dossier master	dossier versionné
dossier versionné	
référence	
statut	
version	
attribut spécifique	
état	état d'élément
relation	
association	
fichier_informatique	
information	

Tableau 5.1 : Correspondances entre les entités du métamodèle proposé et celles du modèle de [Gzara, 00]

Pour finaliser notre métamodèle, l'étape suivante consiste à structurer les entités identifiées entre elles. Ceci est réalisé dans le paragraphe suivant.

5.3 3^{ème} étape : *identification de la structure de notre métamodèle du modèle métier PLM et mise en œuvre dans EMF*

Dans ce paragraphe, l'objectif est de construire le métamodèle en définissant la structure entre les entités. Tout d'abord, un métamodèle contenant les entités principales identifiées est créé (§ 5.3.1). Puis deux points particuliers relatifs aux systèmes PLM sont détaillés à partir de ces entités. Le paragraphe illustre ensuite la gestion des versions avec ou sans élément master (cf. § 5.3.2). Enfin, le paragraphe 5.3.3 illustre la structuration d'éléments à partir de plusieurs natures de relations.

5.3.1 Structure des entités principales

Tout d'abord, rappelons que le métamodèle proposé ne concerne que les données (et pas les traitements) et qu'il n'est applicable que pour les produits mécatroniques. Rappelons également que son objectif est de constituer un support à la construction du modèle métier du système PLM d'une entreprise. Ainsi, l'entreprise peut le modifier selon ses besoins. Le métamodèle proposé permet de créer toutes les spécificités des modèles étudiés. Cette caractéristique tient lieu de validation pour notre métamodèle. Il est à noter que dans l'état actuel, le métamodèle proposé n'évite pas de faire des incohérences. La définition des contraintes fait ainsi partie de nos perspectives.

La figure 5.15 montre les principales entités du métamodèle proposé.

Dans EMF, une entité racine (« root » dans notre métamodèle) est nécessaire. Les entités basiques utilisées pour créer le modèle, « element », « relation » et « association » dans notre cas, doivent être reliées à cette entité par une relation de composition (« root2element », « root2relation » et « root2association » dans notre métamodèle).

Un « élément » est un « article », un « document » ou un « dossier ». Dans le métamodèle, l'entité élément est utilisée pour factoriser les propriétés d'article, document et dossier. Elle n'a pas vocation à être instanciée dans un modèle. Nous l'avons donc déclaré abstraite. Les caractères italiques qu'EMF utilise pour écrire « élément » traduisent ce choix. Un élément est décrit au minimum par un

« attribut » (la référence) et il peut encapsuler aucun, un ou plusieurs « fichiers informatiques ». Si l'objet est détruit, les attributs et le fichier sont également détruits. Les relations entre l'entité élément et les entités attribut et fichier_informatique sont ainsi des relations de composition⁴⁵. L'attribut statut est associé à différents « états ». Comme vu ci-dessus, un « élément » est associé à un autre « élément » (pouvant être lui-même) par une « relation ». Cette structure est créée grâce à une entité « association ». Une relation peut être décrite par des attributs. Ceci permet de définir les quantités dans les relations de nomenclatures entre les organes du produit.

Concernant la modélisation des redondances, un fichier informatique peut être le « format_neutre » d'un autre fichier informatique. Une « information » contenue dans le fichier peut être une copie d'un attribut de l'objet.

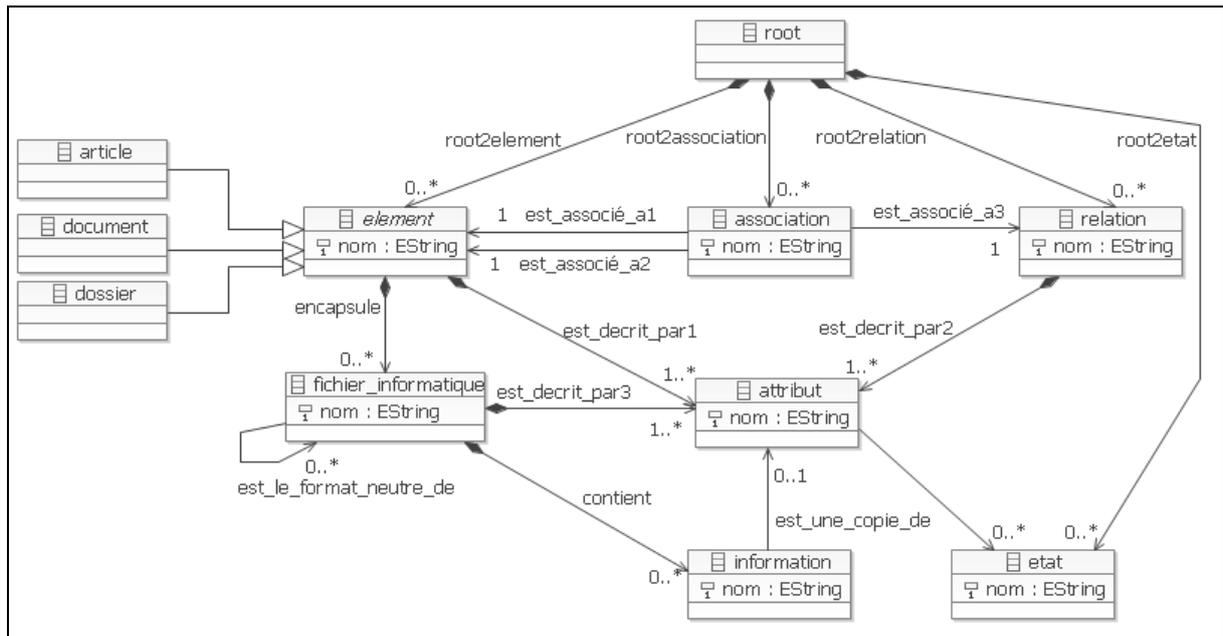


Figure 5.15 : Métamodèle métier de données proposé (produit mécatronique)

Les relations entre « element » et « attribut » et entre « relation » et « attribut » ont la même sémantique « est_decrit_par ». Pour les différencier, nous ajoutons un chiffre à leur nom : « est_decrit_par1 » et « est_decrit_par2 ».

Pour reprendre le vocabulaire d'EMF présenté dans le paragraphe 3.2, ce métamodèle est construit à partir de 11 EClass (root, article, document, attribut...), de 7 EAttributes (« nom » des EClass association, relation, attribut...), de 15 EReferences (root2element, est_associé_a1, encapsule...) et de la définition de 3 héritages (element est le Surtype des EClass article, document et dossier).

Le métamodèle ci-dessus présente une structure générique. Tout « élément » peut encapsuler des « fichiers », tout « élément » est décrit par un ou plusieurs « attributs », ... Cette structure peut être affinée en précisant les différents types d'entités et les conséquences sur la structure de ces entités. Dans les paragraphes suivants, nous étudions respectivement des spécificités pouvant être définies concernant les entités élément, attribut et fichier informatique (§ 5.3.2) puis des spécificités concernant les entités relations (§ 5.3.3).

Rappelons que nos travaux n'ont pas pour objectif de créer un métamodèle du modèle métier PLM complet. Leur but est de montrer comment construire un métamodèle en illustrant la démarche sur des sujets spécifiques aux systèmes PLM.

⁴⁵ Rappelons que cette relation est représentée par un losange noir dans le langage UML

5.3.2 Gestion des versions avec ou sans élément master

La figure 5.16 illustre la structure entre les « articles », les « attributs », les « états » et les « fichiers informatiques » dans le cas d'une gestion des versions avec des éléments versionnés. « L'article versionné » est décrit par des statuts et par des attributs référence, version, statut, désignation et attribut spécifique. L'attribut statut peut prendre une valeur dans un ensemble d'états possibles. L'article versionné peut encapsuler un ou plusieurs fichiers.

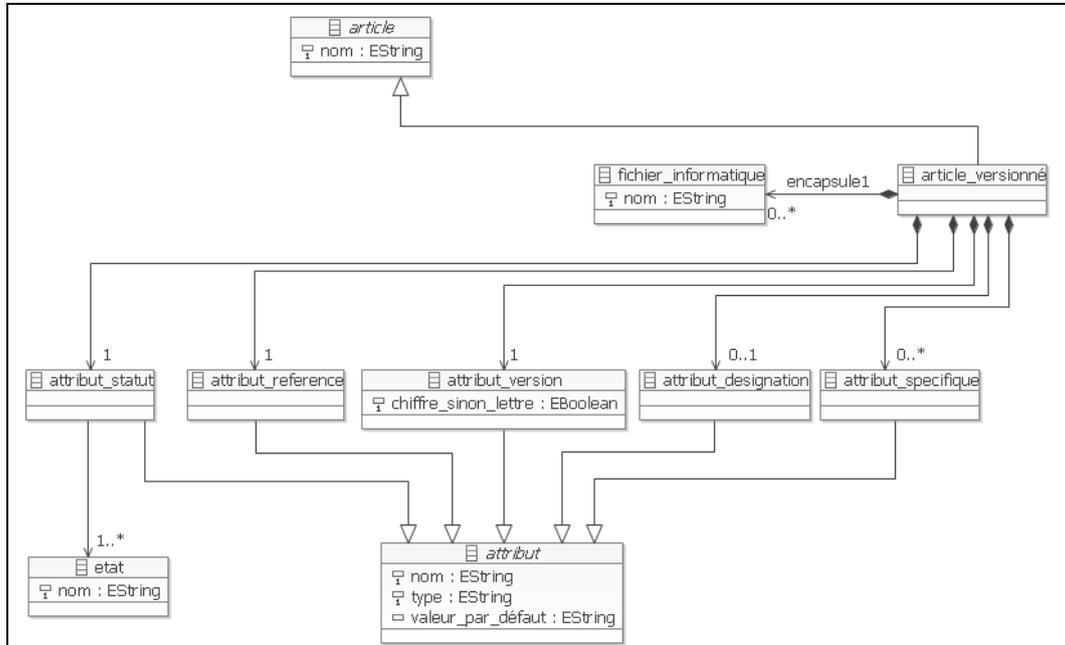


Figure 5.16 : Exemple de gestion des versions sans élément master

La figure 5.17 illustre les mêmes entités dans le cas d'une gestion des versions avec des éléments master et des éléments version de master. « L'article master » « a pour détail » une ou plusieurs « version_article_master ». La relation entre « article master » et les « versions de l'article master » est une relation de composition : une « version de l'article master » ne peut pas exister sans « l'article master » et un « article master » a au minimum une « version de l'article master ». L'attribut référence est porté par « l'article master » et les statuts et l'attribut version sont portés par la « version de l'article master ». Des attributs spécifiques peuvent exister tant pour « l'article master » que pour « la version de l'article master ». Le ou les fichiers informatiques sont encapsulés dans les objets « versions de l'article master ».

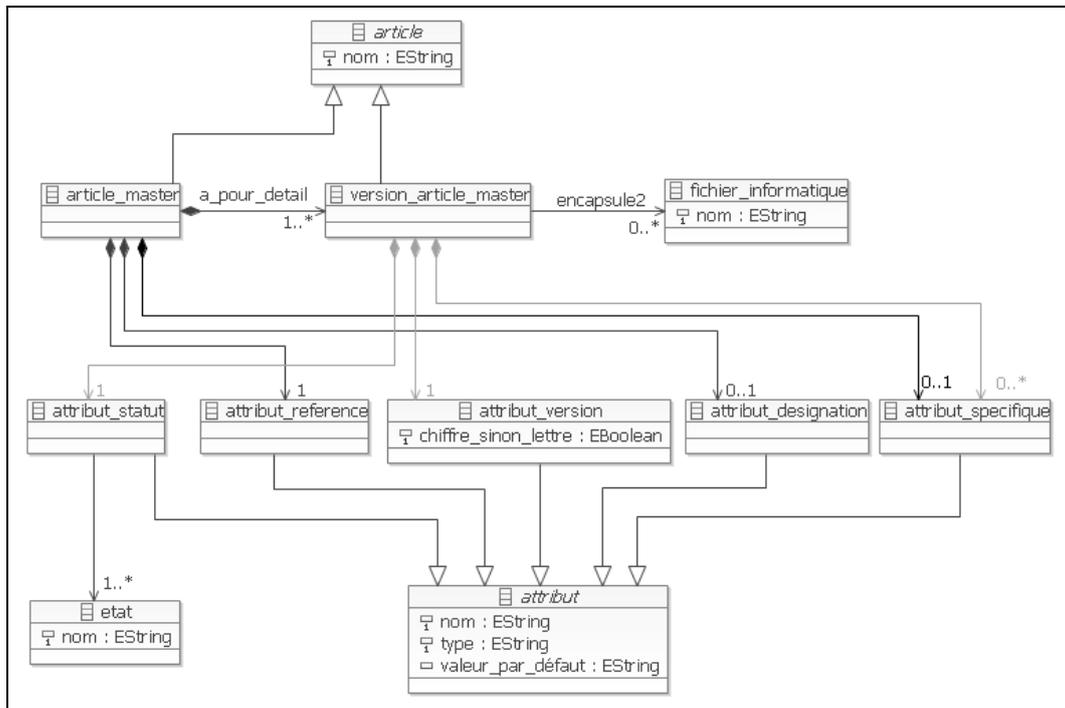


Figure 5.17 : Exemple de gestion des versions avec un élément master

5.3.3 Structuration des « éléments »

La figure 5.18 présente un exemple de structuration des éléments en définissant trois natures de relation. Les relations « structure_produit_non_CAO » permettent de construire des nomenclatures de produits indépendamment de la structure des fichiers CAO. Ceci correspond à un modèle « item centric ». Les relations « dépendance_CAO » mettent en évidence les interactions natives entre des objets « documents versionnés » encapsulant des fichiers CAO. Par exemple, une relation de cette nature peut identifier les interactions entre un objet encapsulant un modèle 3D et un objet encapsulant la mise en plan correspondante. Les relations « documentaire » permettent de relier les différents « documents » aux « articles ».

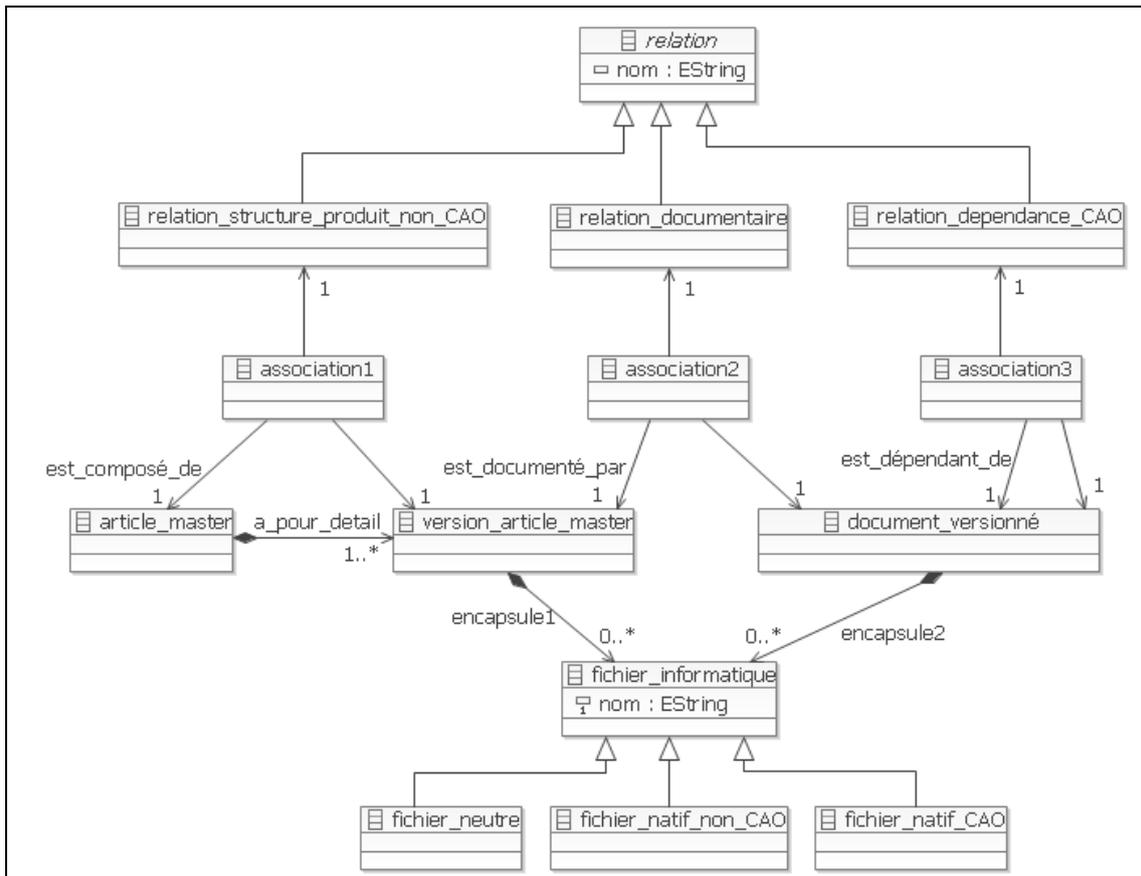


Figure 5.18 : Relations entre des entités « relation », « association », « élément » et « fichier »

5.3.4 Conclusion

Le métamodèle obtenu a été construit à partir des besoins identifiés dans le Club PLM. Il identifie les principales entités à considérer lors de la construction du modèle métier PLM. Il met également en lumière plusieurs points importants : l'utilisation ou non d'élément master, l'identification des dépendances entre les fichiers CAO, la redondance entre les fichiers en format natif et les fichiers en format neutre. Ces problèmes ont été illustrés par les exemples des figures 5.16, 5.17 et 5.18. Cependant, trop détailler le métamodèle peut être un frein à sa compréhension. Chaque entreprise doit ainsi adapter le métamodèle en choisissant les entités proposées répondant à ses besoins et en en créant d'autres si nécessaire.

5.4 Création d'un modèle métier à partir du métamodèle métier proposé dans EMF

Le premier objectif de ce paragraphe est d'illustrer la création d'un modèle métier PLM à partir du métamodèle. Le second objectif est de comparer le modèle métier PLM obtenu avec celui utilisé habituellement, c'est à dire un modèle réalisé au moyen d'un outil de bureautique.

Dans un premier temps, ce paragraphe présente les fonctions d'EMF qui permettent de créer des modèles à partir d'un métamodèle (§ 5.4.1). Ensuite, nous reprenons l'exemple du modèle métier PLM (cf. chapitre 3, § 3.1.2). Le modèle métier PLM correspondant à cet exemple est créé dans EMF à partir du métamodèle métier PLM défini pour réaliser la comparaison.

5.4.1 Création de modèles à partir d'un métamodèle dans EMF

Dans EMF, il existe deux possibilités de créer des modèles à partir d'un métamodèle. La première consiste à générer un environnement de modélisation à partir du métamodèle défini puis à créer les modèles dans cet environnement. La deuxième consiste à générer directement des instances à partir du métamodèle (cf. chapitre 4, § 2.1.1). Dans cette étude, nous nous appuyons sur cette deuxième possibilité.

Pour illustrer l'instanciation d'un métamodèle avec EMF, nous nous appuyons sur le métamodèle proposé dans la figure 5.15.

La première action consiste à se mettre sur l'EClass « root » et à créer une instance. La figure 5.19 montre le résultat obtenu à savoir : un fichier en format XMI avec un objet « root ». A partir de cet objet, EMF propose de créer des instances de toutes les EClass qui sont reliées à l'EClass « root » dans le métamodèle. Une liste propose ainsi les différentes EClass possibles en indiquant le nom de la relation entre l'EClass « root » et l'EClass possible plus le nom de l'EClass. La relation de composition définie entre l'EClass « root » et les différentes EClass possibles dans le métamodèle est traduite au moment de l'instanciation par une relation père-fils dans le modèle (« child »). Si l'objet root est supprimé, tous les objets dont il est le père le sont également.

Ainsi, pour créer une instance de l'EClass « Article », EMF propose de créer un nouveau « child » de « root » en indiquant le nom de la EReference « Root2element » et de l'EClass associée « Article ».

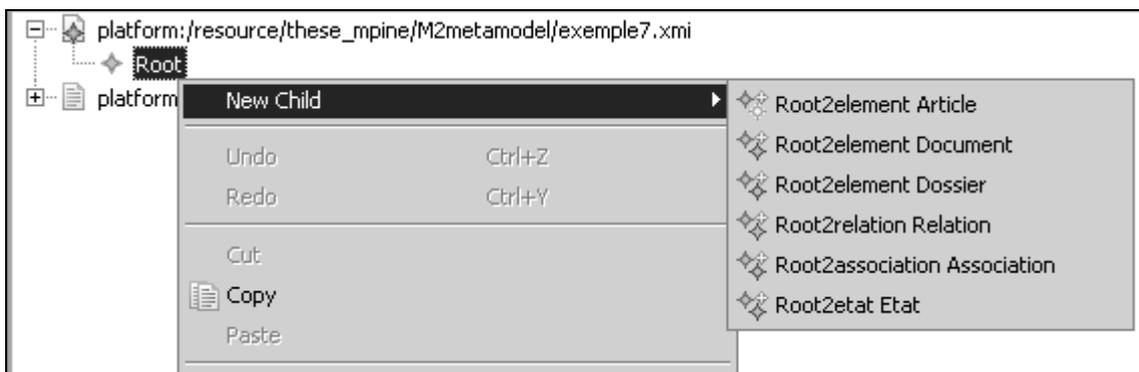


Figure 5.19 : Exemple d'instanciation d'un métamodèle avec EMF – première action

Ainsi, un objet « produit » est créé en instanciant l'EClass « Article » (cf. figure 5.20). Dans les propriétés de l'objet, nous retrouvons l'EAttribute « nom » défini dans le métamodèle. Dans l'arborescence, l'objet est décrit avec en premier le nom de l'EClass dont il est une instance (« Article » dans notre exemple) et en deuxième la valeur de la première de ses propriétés (« produit » dans notre exemple).

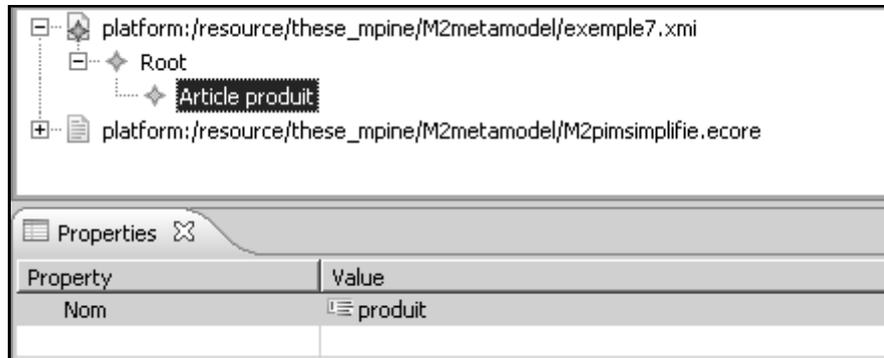


Figure 5.20 : Exemple d'instanciation d'un métamodèle avec EMF – instanciation d'une EClass

La figure 5.21 illustre l'exemple de la création d'une structure à partir des EReferences déclarées dans le métamodèle. Dans cet exemple, nous créons un objet « cahier des charges » en instanciant l'EClass « Document », un objet « est_documenté_par » en instanciant l'EClass « Relation » et un objet « produit2cahier_des_charges » en instanciant l'EClass « Association ». Dans les propriétés de l'objet produit2cahier_des_charges, nous retrouvons les trois EReferences « est_associé_a1 », « est_associé_a2 » et « est_associé_a3 » définies dans le métamodèle. En cliquant sur le champ valeur, EMF propose les objets potentiellement concernés par chaque EReference. Pour les EReferences « est_associé_a1 » et « est_associé_a2 », il propose les objets instanciés des EClass « article », « document » ou « dossier ». Pour l'EReference « est_associé_a3 », il propose les objets instanciés des EClass « relation ».

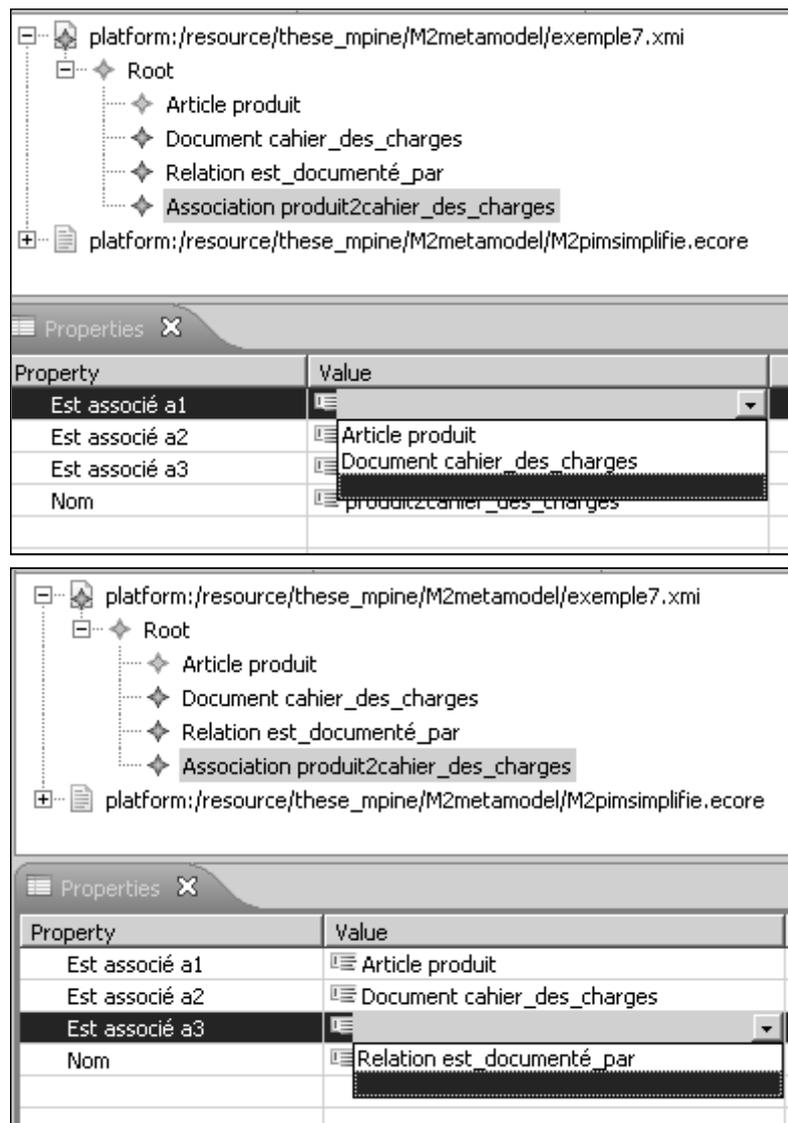


Figure 5.21 : Exemple d'instanciation d'un métamodèle avec EMF – création d'une structure par les EReferences

La figure 5.22 illustre les menus contextuels proposés pour chaque type d'objet. Conformément aux relations de composition dans le métamodèle, l'objet « produit » peut avoir pour fils des instances des EClass « attribut » et « fichier_informatique ».



Figure 5.22 : Exemple d'instanciation d'un métamodèle avec EMF – menu contextuel

La figure 5.23 illustre la possibilité d'attribuer plusieurs valeurs à une propriété. Dans l'exemple, nous avons créé des attributs référence, version et statut pour décrire les cahiers des charges. Nous avons également créé cinq états : en création, valide, obsolète, prototype et série. La figure illustre que nous attribuons trois valeurs possibles au statut des cahiers des charges : en création, valide et obsolète.

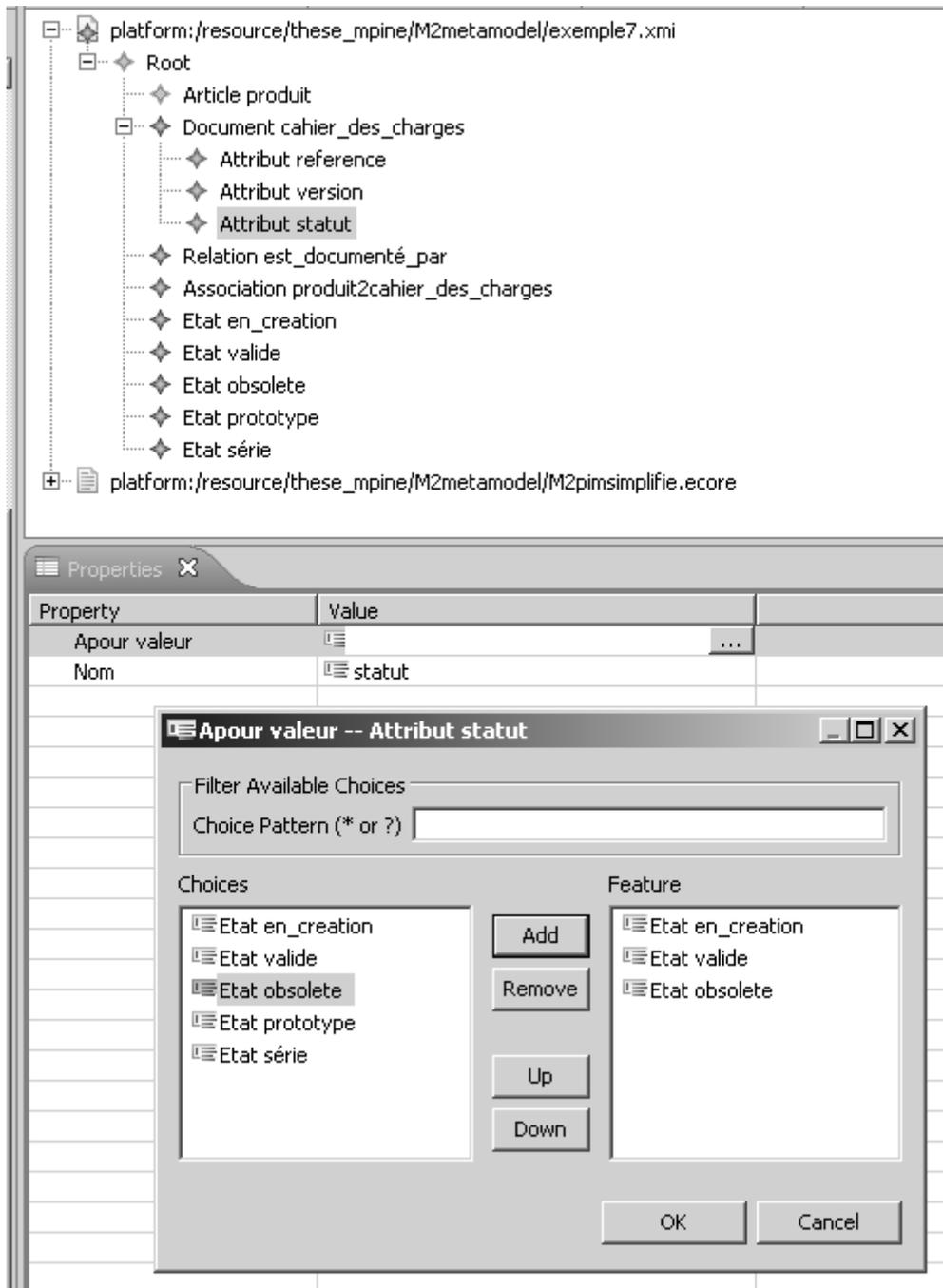


Figure 5.23 : Exemple d'instanciation d'un métamodèle avec EMF – plusieurs valeurs pour une propriété

Au travers de cet exemple, nous constatons que tout modèle créé à partir d'un métamodèle dans EMF est formalisé sous la forme d'une arborescence. Les niveaux de l'arborescence représentent les différentes relations de composition déclarées dans le métamodèle.

5.4.2 Exemple d'instanciation du métamodèle métier

Dans le paragraphe 2.1.3 du chapitre 3, nous avons fait le constat que les modèles métiers PLM des entreprises sont souvent formalisés avec des outils de bureautique. Dans ce paragraphe, nous proposons d'illustrer notre métamodèle métier au travers d'un exemple déjà présenté (cf. chapitre 3, § 3.1.2, repris dans la figure 5.24).

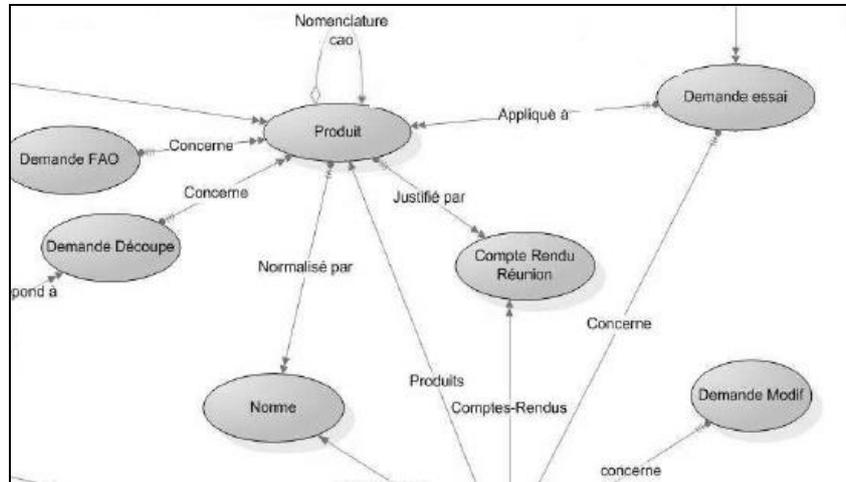


Figure 5.24 : Extrait du modèle métier de la société Duqueine mis en oeuvre au moyen de la personnalisation de leur logiciel PLM (Témoignage Journée du PLM 2008)

Cet exemple a deux objectifs : illustrer l'instanciation du métamodèle de la figure 5.15 en faisant une corrélation entre les entités du métamodèle et celles du modèle, comparer deux formalisations du modèle : celle réalisée avec un outil de bureautique et celle réalisée avec EMF.

La figure 5.25 montre le modèle obtenu suite à l'instanciation dans EMF. Seules quelques entités représentatives sont modélisées. L'entité « produit » est définie comme une instance de l'EClass « article ». Les entités « demande_FAO » et « demande_Decoupe » sont définies comme des instances de l'EClass « document ». Les relations « nomenclature_cao » et « concerne » sont définies par des instances de l'EClass « relation » et par des instances de l'EClass « association ». Des attributs « reference », « version » et « statut » décrivent « produit », « demande_FAO » et « demande_Decoupe ». Un attribut « quantité » décrit la relation « nomenclature_cao ». Dans le modèle de la figure 5.20, aucune mise en plan n'apparaît. Nous supposons qu'elles sont encapsulées dans le même objet que le produit qu'elles décrivent. Ainsi, le produit encapsule le fichier du modèle 3d mais également le fichier de la mise en plan dans notre modèle. Il encapsule également les fichiers au format neutre correspondant à ces deux fichiers.

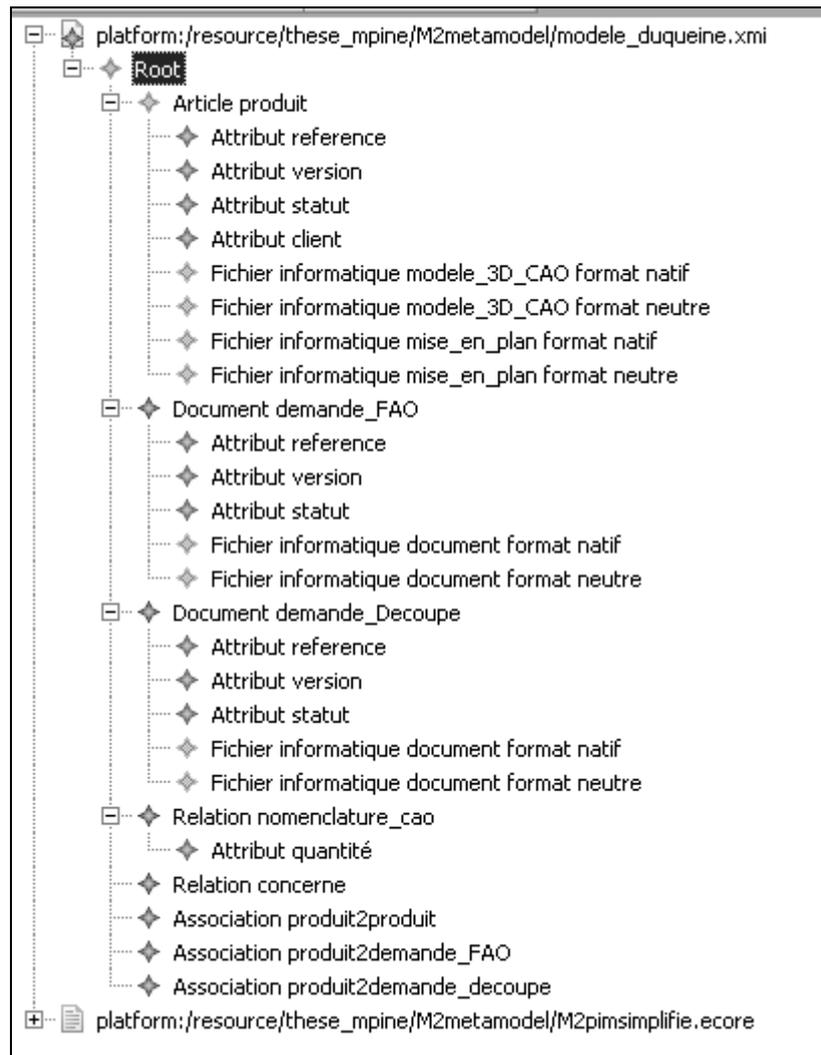


Figure 5.25 : Extrait de modèle métier de la société Duqueine créé à partir du métamodèle métier proposé

Cet exemple a montré comment instancier différentes entités du métamodèle métier proposé pour créer un modèle métier. Dans le modèle de la figure 5.24, une même relation, la relation « concerne », est représentée quatre fois. Le modèle créé à partir d'EMF est ainsi plus formel, donc plus exploitable, que celui existant. Il est également plus complet. Cependant, force est de constater que la lisibilité de celui de la figure 5.24 est plus grande pour deux raisons. La première raison est que l'aspect graphique met mieux en lumière les relations entre les entités. La deuxième raison est que la non-utilisation d'entités « association » et d'entités « relation » simplifie la lecture du modèle.

5.4.3 Conclusion

Ce paragraphe a présenté l'instanciation du métamodèle métier proposé avec EMF. A partir de l'entité racine (« root »), les différentes entités des modèles sont créées au fur et à mesure. Les relations de composition sont définies par des relations « père »-« fils » entre les entités. Les autres relations sont définies grâce à des propriétés contenues dans les entités. Les modèles obtenus sont présentés sous forme d'arborescences. Ils sont plus formels que ceux réalisés avec des outils de bureautique. Cependant, la structuration des entités entre elles n'est pas identifiable rapidement. Dans l'exemple de la figure 5.25, il faut aller dans les propriétés de l'entité « produit2produit » pour voir que l'entité « produit » est reliée à elle-même par la relation « nomenclature_cao ».

5.5 Conclusion

L'objectif de ce paragraphe était de définir un métamodèle du modèle métier PLM dans l'environnement EMF et d'expérimenter l'instanciation de ce métamodèle.

Suite à une étude auprès de membres du Club PLM, plusieurs conclusions ont pu être tirées. La première est que les modèles métiers PLM étudiés sont applicables pour des produits mécaniques ou mécatroniques. Ainsi, le métamodèle que nous construisons correspond à cette nature de produit. Pour d'autres natures de produits, chimique par exemple, il est nécessaire de faire des études complémentaires. Cette étude a également permis d'identifier des points importants dans la création du modèle métier PLM. A partir de ces points, nous avons défini les principales entités de notre métamodèle du modèle métier PLM. Enfin, nous avons expérimenté l'instanciation de ce métamodèle.

Plusieurs conclusions ont été faites tout au long de la construction et de l'instanciation du métamodèle du modèle métier PLM. La première montre que le nombre d'éléments intégrés dans le métamodèle peut être un frein à sa compréhension. Ainsi, chaque entreprise doit adapter le métamodèle en choisissant les entités répondant à ses besoins et en en créant d'autres si nécessaire. La deuxième est que le graphisme des modèles ne permet pas d'identifier rapidement les relations entre les entités. La mise en oeuvre d'outils complémentaires comme GMF pourrait être une réponse à ce problème.

Enfin, plusieurs perspectives ont été listées. La première est d'étendre cette étude à la partie traitement du système PLM. La deuxième est d'étudier comment mettre en oeuvre des contraintes dans le métamodèle afin d'éviter les incohérences au moment de l'instanciation. La dernière consiste à définir les correspondances entre les entités de notre métamodèle métier et les entités du protocole d'application 214 de la norme STEP [ISO, 10].

6 Définition du métamodèle du modèle personnalisé d'un logiciel PLM dans EMF

Une démarche pour construire le métamodèle de personnalisation d'un logiciel PLM relatif aux données dans un environnement de modélisation a été proposée dans le chapitre précédent (cf. chapitre 4, § 5). Dans ce paragraphe, cette démarche est appliquée sur le logiciel PLM Audros. Cette démarche consiste à traduire le métamodèle du modèle personnalisé d'Audros défini en langage relationnel en un métamodèle construit à partir du méta-métamodèle Ecore dans l'environnement EMF.

6.1 Définition du métamodèle de personnalisation d'Audros dans EMF

Dans le cadre de nos travaux, nous nous sommes concentrés sur l'aspect « données » du métamodèle du modèle personnalisé du logiciel PLM. L'expérimentation que nous avons réalisée a été faite sur le logiciel PLM Audros. Pour des raisons de confidentialité, nous ne montrons qu'un extrait du métamodèle obtenu.

Ce chapitre définit qu'il est nécessaire d'établir des correspondances entre le métamodèle du modèle personnalisé du logiciel Audros formalisé en Ecore et celui implémenté dans la base de données Oracle (cf. § 4). Ce paragraphe complète la démarche proposée dans le chapitre précédent en ajoutant une cinquième étape pour identifier ces correspondances (cf. chapitre 4, § 5).

La figure 5.26 illustre les différentes transformations pour passer du métamodèle 1 non normalisé formalisé en langage relationnel au métamodèle 4 normalisé construit à partir du méta-métamodèle Ecore.

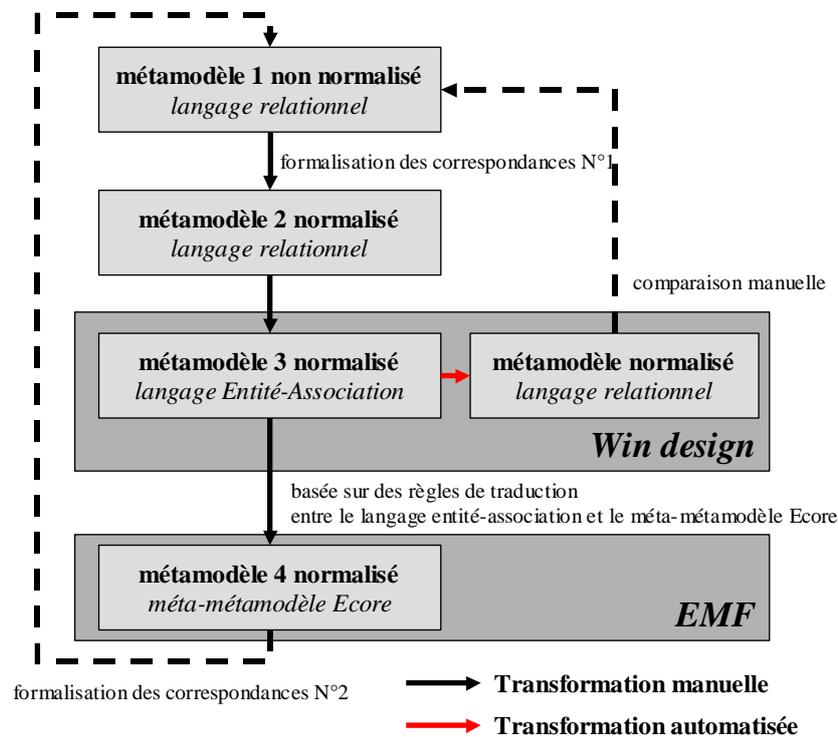


Figure 5.26 : Création du métamodèle de données de personnalisation du logiciel PLM Ecore à partir du métamodèle relationnel

La démarche proposée est ainsi constituée de cinq étapes. Son objectif est de créer le métamodèle 4 normalisé en Ecore et d'identifier les correspondances entre ce métamodèle et le métamodèle 1 non normalisé. Le métamodèle 4 répond à deux besoins. Le premier est de formaliser le fonctionnement du logiciel PLM afin d'en supporter la compréhension. Le deuxième est d'être un support pour la mise en œuvre de transformations de modèles. Dans notre sujet d'étude, le métamodèle 4 normalisé est le métamodèle d'arrivée de la transformation « métamodèle métier - métamodèle Audros 4 normalisé ». Il est également le métamodèle de départ pour la transformation « métamodèle Audros 4 normalisé - métamodèle Audros 1 non normalisé » (cf. figure 5.27). Les entités du métamodèle Audros 4 doivent ainsi être définies pour supporter au mieux ces deux transformations. Lors de la modélisation, la nécessité de faire des choix de modélisation en faveur de l'un ou l'autre de ces deux besoins (support à la compréhension et support aux transformations) peut apparaître.

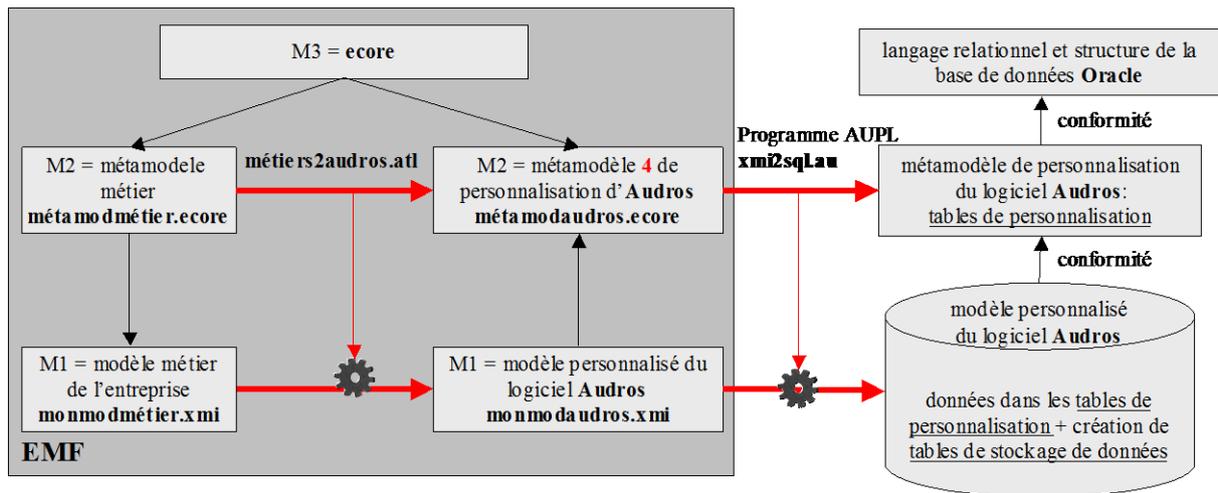


Figure 5.27 : Transformations dont le métamodèle 4 fait partie

La première étape de la démarche proposée consiste à étudier le **métamodèle 1 non normalisé** formalisé sous forme de tables relationnelles. Dans un logiciel PLM, certaines tables servent au fonctionnement du logiciel, certaines tables servent à la personnalisation du modèle et certaines tables servent au stockage des données. Notre objectif étant de personnaliser le modèle, il faut identifier les tables concernant la personnalisation et les tables étant impactées par la personnalisation. Pour cela, nous utilisons cinq sources. La première est constituée du métamodèle relationnel fourni par l'éditeur. La deuxième est constituée de tests de personnalisation d'une base de données du logiciel PLM Audros. La troisième est constituée d'un fichier identifiant les clés des tables. La quatrième est constituée d'un programme de personnalisation automatique d'une base de données à partir d'un fichier excel. La cinquième est constituée des connaissances de différents experts du logiciel Audros. Lors de cette étude, nous avons identifié qu'Audros crée des tables spécifiques de stockage au moment de la personnalisation.

Ainsi, la nécessité d'un *premier choix* apparaît : faut-il ou non identifier ces tables dans le modèle instancié à partir du métamodèle Audros 4 normalisé? Notre étude est centrée sur la personnalisation du logiciel PLM. Par conséquent, modéliser les tables de stockage a peu d'intérêt. Ainsi, nous ne représentons pas ces tables dans les modèles d'EMF. Leur création est définie dans le programme de transformation xmi2sql.au (cf. figure 5.27) qui définit le modèle personnalisé dans la base de données à partir d'un modèle xmi créé dans EMF.

La deuxième étape consiste à reconstruire le métamodèle 1 pour obtenir un **métamodèle 2 normalisé** afin d'avoir une structure de données cohérente et non redondante. Pour cela, l'analyse de la sémantique des attributs utilisés dans les tables et l'identification des clés ont tout d'abord été réalisées. En effet, le métamodèle relationnel de la base de données n'est pas normalisé et certains attributs des tables sont dupliqués pour des raisons de performance. Suite à cette étude, des regroupements de plusieurs tables dans une seule table ont également été identifiés. Ceci est traduit dans le métamodèle 3 normalisé par trois entités et deux relations d'héritage. A l'inverse, une table qui aurait dû être unique a été dupliquée en deux tables. Ainsi, la nécessité d'un *deuxième choix* apparaît : faut-il modéliser une seule entité (choix mieux adapté pour la compréhension) ou deux entités (choix plus proche du métamodèle 1 relationnel) ? Nous ferons le choix de modéliser deux entités dans le métamodèle 3 normalisé. Ce choix n'est pas un frein à la compréhension du fonctionnement et il simplifie la mise en œuvre de la transformation entre le métamodèle 1 et le métamodèle 4.

La troisième étape consiste à construire le **métamodèle 3** en langage entité-association. Outre les entités et les associations, ce métamodèle précise les différentes contraintes liées au fonctionnement du logiciel PLM (cardinalités, contraintes d'inclusion, d'exclusion, de totalité, etc.). L'environnement de modélisation Win Design [Win Design, 13] utilisé permet de générer automatiquement le métamodèle logique (métamodèle 2) à partir du métamodèle conceptuel (métamodèle 3). Il est ainsi possible de valider la rétro-conception en comparant le métamodèle relationnel obtenu avec le métamodèle initial.

La quatrième étape consiste à construire le **métamodèle 4 Ecore** dans EMF à partir du métamodèle 3 créé dans Win Design. Les différentes contraintes (exclusion, inclusion...) proposées dans Win Design ne sont pas disponibles dans EMF. Pour pouvoir les mettre en œuvre, il faudrait installer un plugin supportant le langage OCL (Object Constraint Language) et traduire les contraintes en OCL. La nécessité d'un *troisième choix* apparaît : faut-il formaliser ou non ces contraintes dans le métamodèle 4 ? La formalisation permet une meilleure compréhension du logiciel. Les contraintes étant formalisées dans le métamodèle 3 E-A, ce besoin est satisfait. Le deuxième besoin est de supporter la mise en œuvre des transformations de modèles. Dans notre sujet d'étude, le métamodèle 4 est le métamodèle d'arrivée de la transformation ATL. Le modèle personnalisé du logiciel PLM est créé de manière automatique par le programme ATL à partir du modèle métier de départ, qui est une instance du métamodèle métier. Dans EMF, l'exécution d'un programme ATL n'indique pas de message d'erreur si des données sont perdues dans la transformation suite à une contrainte dans le métamodèle d'arrivée. Les contraintes doivent ainsi être prises en compte par le modélisateur lors de la définition du programme ATL. Il n'y a ainsi pas d'intérêt à les formaliser en langage OCL dans le métamodèle 4. Avant de traduire le métamodèle 3, il est nécessaire de définir des règles de traduction entre le langage entité association et le méta-métamodèle Ecore. Comme exemple de règles de traduction, nous pouvons citer la proposition décrite dans le paragraphe 3.2.1 pour traduire les relations n-aire

Durant cette étape, nous avons fait deux choix de modélisation pour faciliter les transformations.

Tout d'abord, nous avons séparé la représentation des attributs intrinsèques à Audros (référence, version, etc.) de celle des autres attributs. Nous avons ainsi des entités « attribut_reference », « attribut_version »... et une entité « attribut_spécifique ». Cette distinction facilite à la fois la transformation entre le métamodèle métier et ce métamodèle et la transformation entre ce métamodèle et le métamodèle 1.

Une amélioration possible du métamodèle 4 a également été identifiée lors de la création du programme de transformation entre le métamodèle 4 et le métamodèle 1. La figure 5.28 illustre un extrait du métamodèle 4 initial avec deux EClass : « classe_d_objets » et « classe_de_liens ».

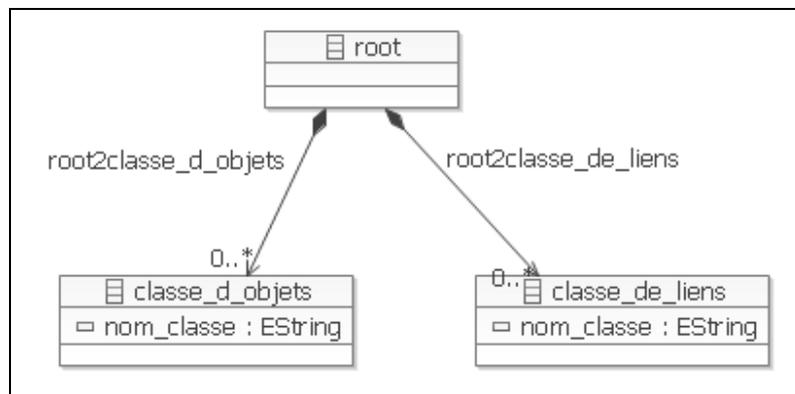


Figure 5.28 : Extrait du métamodèle 4 initial avec deux EClass

Nousinstancions ce métamodèle en créant deux classe_d_objets « produit » et « document » et deux classe_de_liens « est_documente_par » et « est_compose_de » (cf. figure 5.29).

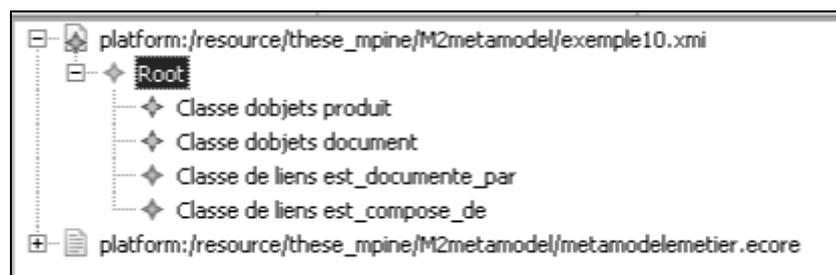


Figure 5.29 : Instanciation du métamodèle de la figure 5.28

Le fichier correspondant au modèle de la figure 5.29 est un fichier au format XMI (cf. figure 5.30). Ce fichier sert de donnée d'entrée au programme qui personnalise le logiciel PLM en écrivant dans la base de données.

```

exemple10.xmi - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="ASCII"?>
<metamodelemetier:root xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:metamodelemetier="http://metamodelemetier/1.0"
xsi:schemaLocation="http://metamodelemetier/1.0 metamodelemetier.ecore">
  <root2classe_d_objets nom_classe="produit"/>
  <root2classe_d_objets nom_classe="document"/>
  <root2classe_de_liens nom_classe="est_documente_par"/>
  <root2classe_de_liens nom_classe="est_compose_de"/>
</metamodelemetier:root>

```

Figure 5.30 : Fichier XMI correspondant au modèle de la figure 5.29

On constate dans cette figure que les instances des classe_d_objets et de classe_de_liens ne sont pas séparées. Pour faciliter le traitement de ce fichier par le programme xmi2sql.au, nous ajoutons dans notre métamodèle des EClass qui regroupent les entités des EClass classe_d_objets et classe_de_liens grâce à une relation de composition. Dans notre exemple, « regroupement_classe_d_objets » est composé de toutes les « classe_d_objets » et « regroupement_classe_liens » est composé de toutes les « classe_liens » (cf. figure 5.31).

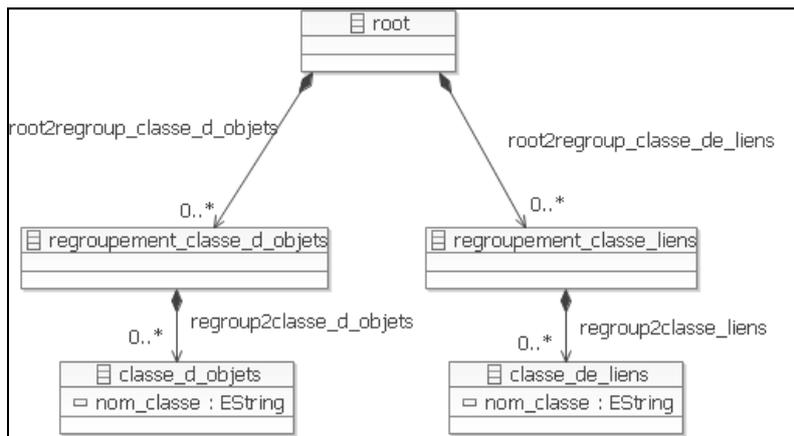


Figure 5.31 : Métamodèle de la figure 5.28 amélioré

Nousinstancions ce métamodèle en créant les quatre entités précédentes (« produit », « document », « est_documente_par » et « est_compose_de ») ainsi que deux entités « regroupement classe dobjets » et « regroupement classe liens » (cf. figure 5.32).

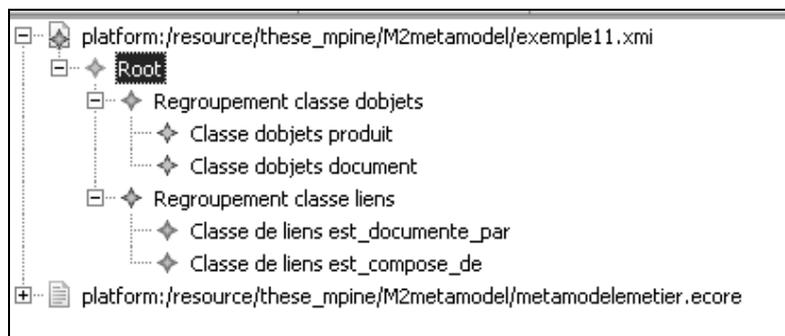


Figure 5.32 : Instanciation du métamodèle de la figure 5.31

Dans le fichier XMI (cf figure 5.33), on constate que les instances des classe_d_objets et des classe_de_liens sont séparées et que toutes les instances d'une même EClass sont regroupées ensemble. Chaque EClass du métamodèle audros correspond à une table de la base de données relationnelle. Ainsi, cette structuration permet un traitement plus facile du fichier XMI par le programme AUPL de transformation.

```

<?xml version="1.0" encoding="ASCII"?>
<metamodelemetier:root xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:metamodelemetier="http://metamodelemetier/1.0"
xsi:schemaLocation="http://metamodelemetier/1.0 metamodelemetier.ecore">
  <root2regroup_classe_d_objets>
    <regroup2classe_d_objets nom_classe="produit"/>
    <regroup2classe_d_objets nom_classe="document"/>
  </root2regroup_classe_d_objets>
  <root2regroup_classe_de_liens>
    <regroup2classe_liens nom_classe="est_documente_par"/>
    <regroup2classe_liens nom_classe="est_compose_de"/>
  </root2regroup_classe_de_liens>
</metamodelemetier:root>

```

Figure 5.33 : Fichier XMI correspondant au modèle de la figure 5.32

Enfin, **la cinquième étape** consiste à définir les correspondances entre le métamodèle 4 audros en Ecore et le métamodèle 1 Audros de la base de donnée afin de construire le programme AUPL xmi2sql.au. Cette formalisation des correspondances N°2 s'appuie sur celle déjà effectuée (formalisation des correspondances N°1) lors de la normalisation du métamodèle (cf. figure 5.26).

La démarche proposée requiert une expertise du logiciel étudié. Ainsi, notre expérimentation a été validée par des experts du logiciel Audros. Avoir plusieurs sources pour analyser le métamodèle du logiciel est primordial. En effet, certaines règles de gestion du logiciel ne sont pas perceptibles lors de l'étude de la base de données. Par exemple, nous avons pu identifier que la gestion avec un thesaurus n'est possible que si l'attribut de l'objet est de type chaîne, ce qui n'est pas visible dans les tables du modèle relationnel.

Les logiciels PLM sont des logiciels complexes. Il est ainsi nécessaire de valider avec un expert du logiciel PLM concerné la démarche proposée avant de la mettre en œuvre. Elle sera ensuite appliquée telle quelle ou adaptée selon les spécificités du logiciel PLM.

6.2 Conclusion

Ce paragraphe a défini le métamodèle du modèle personnalisé d'Audros relatif aux données. Contrairement à la construction du métamodèle métier, ce travail de construction n'a pas été un travail de métamodélisation. En effet, les entités et la structure du métamodèle Ecore obtenu correspondent à une traduction et une adaptation des tables de la base de données d'Audros. Le point délicat dans la construction du métamodèle de personnalisation du logiciel PLM est d'acquérir une bonne connaissance du logiciel PLM afin de construire un métamodèle correct et complet. Le métamodèle de personnalisation répond à deux besoins. Le premier est de formaliser le fonctionnement du logiciel PLM afin d'en supporter la compréhension. Ceci permet également de capitaliser les savoir-faire autour du logiciel. Le deuxième est d'être un support pour la mise en œuvre de transformations de modèles. Ainsi, nous avons illustré le choix d'entités facilitant les transformations dans le métamodèle.

7 Conclusion

Dans ce chapitre, nous avons expérimenté l'utilisation d'un environnement de modélisation pour supporter la création de métamodèles et de modèles constituant le cadre de modélisation proposé. Notre sujet d'étude était le passage entre le métamodèle métier et le métamodèle de personnalisation relatifs aux données dans le logiciel PLM. Un focus a été tout particulièrement fait sur la formalisation

des métamodèles. La représentation du produit analysée était la représentation de conception. Le logiciel PLM étudié était le logiciel Audros.

Dans le paragraphe 3, une étude rapide de divers environnements de modélisation a été faite. Deux catégories d'environnement ont été identifiées. La première catégorie regroupe des environnements basés sur l'utilisation de langages standards, Win Design par exemple. La deuxième catégorie regroupe des environnements basés sur la définition de métamodèles, EMF par exemple. Seule cette catégorie permet de mettre en œuvre des métamodèles et des mécanismes de transformation entre les modèles. Notre choix pour outiller notre expérimentation s'est ainsi porté sur EMF.

Dans le paragraphe 4, nous avons identifié la nature des différents métamodèles, modèles et transformations à mettre en œuvre suite aux choix d'EMF et d'Audros.

Dans le paragraphe 5, nous avons défini un métamodèle du modèle métier PLM et nous avons expérimenté son instanciation. Cette formalisation a permis de mettre en lumière l'intérêt d'utiliser un métamodèle adapté au domaine d'étude. Un métamodèle métier détaillé permet de capitaliser les savoirs faire relatifs à la création du modèle et relatifs aux transformations vers d'autres modèles. D'une part, lors de la création du modèle, les entités proposées par le métamodèle et les contraintes formalisées guident le modélisateur. D'autre part, la richesse des métamodèles permet de définir des transformations précises et automatisables.

Dans le paragraphe 6, nous avons formalisé le métamodèle de personnalisation d'Audros dans EMF. Cette formalisation permet d'acquérir de robustes connaissances du logiciel via une analyse approfondie de son fonctionnement.

Suite à cette expérimentation, une première constatation concerne la difficulté liée à l'évolution des métamodèles. En effet, si le métamodèle évolue, les modèles créés à partir de ce dernier peuvent devenir obsolètes. Le métamodèle métier peut évoluer suite à une connaissance plus approfondie du domaine qui conduit à identifier de nouvelles entités ou des contraintes pertinentes à intégrer dans le métamodèle. Dans le cas du métamodèle de personnalisation du logiciel PLM, il est nécessaire de valider que les nouvelles versions du logiciel PLM ne remettent pas en cause le métamodèle défini. Dans le cas contraire, il est nécessaire de mettre à jour le métamodèle. Si les métamodèles évoluent, les programmes de transformations définis doivent évoluer également.

Plusieurs constatations concernent aussi l'utilisation des environnements de modélisation. La première constatation est qu'il est important de bien appréhender le méta-métamodèle proposé par l'environnement (Ecore dans le cas de d'EMF). Cette compréhension est nécessaire pour construire au mieux les métamodèles et pour être capable de mettre en œuvre des transformations entre ces métamodèles. La deuxième constatation est que le manque d'aspect graphique rend la prise en main de l'environnement peu intuitive pour des personnes de profil non « informaticien ». Cependant, le graphisme ne doit en aucun cas être un frein à la bonne compréhension des entités manipulées. La figure 5.5 du paragraphe 3.2.1 illustre ainsi que le format graphique Ecorediag proposé par EMF peut prêter à confusion sur les entités manipulées. En effet, il laisse à penser que les relations entre deux EClass sont portées par une entité autonome alors qu'elles sont portées par une EReference dans une des deux EClass (ou dans les deux) concernée par la relation. Enfin, la troisième constatation est qu'il est nécessaire d'être attentif à la stabilité des métamodèles et des modèles créés lors des évolutions de l'environnement de modélisation utilisé.

Conclusion et perspectives

Conclusion

La raison d'être de l'entreprise manufacturière est de fournir le bon produit sur le bon marché au bon moment et avec le bon coût. Cette raison d'être constitue ainsi un premier but stratégique à atteindre pour l'entreprise. Pour survivre dans un contexte d'économie de marché et de concurrence accrue, l'entreprise doit s'adapter aux évolutions de son environnement. Elle doit également être proactive en innovant constamment. Pour cela, elle définit un deuxième but stratégique : développer son agilité opérationnelle. Afin d'atteindre ces deux buts, l'entreprise met en œuvre divers projets d'entreprise. Parmi ces derniers, le projet PLM vise à maîtriser le produit au moyen de la construction du système PLM. Ce système d'information supporte la création, la circulation, l'utilisation et l'évolution du patrimoine informationnel de représentation du produit. Si la construction du système PLM est en adéquation avec le premier but stratégique identifié, il semble en contradiction avec le développement de l'agilité opérationnelle. En effet, l'agilité opérationnelle prône le changement alors que la construction du système PLM vise à augmenter la robustesse des savoirs relatifs aux produits. Le besoin qui est ainsi mis en lumière est comment construire un système PLM en adéquation avec les deux buts identifiés ? La réponse à ce besoin a constitué le problème de ces travaux de thèse. L'hypothèse formulée dans ce mémoire a été qu'il est nécessaire d'avoir une méthode adaptée pour cette construction. Pour pouvoir supporter l'agilité recherchée, cette méthode doit limiter les séparations entre les perceptions informationnelle et organisationnelle de l'entreprise.

Il est à noter qu'afin de construire des propositions en adéquation avec les besoins du terrain, toutes les analyses contenues dans ce mémoire ont été réalisées à partir d'états de l'art de travaux académiques mais également à partir d'études « terrain » réalisées au sein du Club PLM.

Une méthode de construction d'un système d'information définit un ensemble de modèles et un processus. Tout d'abord, nous nous sommes concentrés sur la première étape de la vie du système PLM, c'est à dire l'élaboration de son cahier des charges. Un état de l'art des travaux relatifs à l'ingénierie des besoins, un état des lieux des spécificités relatives au système PLM et une étude réalisée au sein du Club PLM nous ont permis d'appréhender les tenants et les aboutissants de l'élaboration d'un cahier des charges. Un **processus adapté à l'élaboration du CDC du système PLM** et intégrant les perceptions informationnelle et organisationnelle de l'entreprise a alors été proposé. Dans ce processus, les différents modèles dont l'utilisation est préconisée ont été identifiés et positionnés.

La suite des travaux a alors été consacrée à la modélisation du système PLM au travers d'une nouvelle hypothèse : l'utilisation d'un cadre de modélisation adapté fait partie des solutions possibles pour réaliser une construction du système PLM en adéquation avec les deux buts de l'entreprise identifiés. Notre proposition a ainsi consisté à construire ce cadre et à illustrer sa mise en application. Le **cadre de modélisation** proposé guide le modélisateur dans la construction du système PLM au moyen de deux analyses. La première analyse perçoit de manière conjointe le système PLM et le produit afin de définir les modèles nécessaires à la construction du système PLM. La deuxième analyse perçoit de manière conjointe le système PLM et les systèmes avec lesquels il est en interaction afin de définir les relations d'interface. Pour supporter l'agilité opérationnelle et pour éviter les distorsions entre les différentes perceptions du produit tout au long de sa vie, un paradigme supplémentaire a été utilisé, le **paradigme d'ambivalence**. Ce paradigme promeut un raisonnement synergique entre les familles d'acteurs travaillant autour d'une même réalité. Nous avons proposé l'application de ce paradigme sur deux réalités : l'entreprise et le produit. Pour l'entreprise, l'objectif à atteindre est de développer la synergie entre les techniciens de l'information et les responsables métiers. Pour le produit, la synergie est recherchée entre les responsables métiers des différentes étapes du cycle de vie du produit. La mise en application d'un raisonnement synergique implique deux changements organisationnels. Le premier changement consiste à casser le fonctionnement client fournisseur instauré entre les services. Le

deuxième changement implique que chaque famille d'acteurs doit enrichir l'autre de par ses connaissances. Ces deux changements sont supportés entre autres par la mise en œuvre d'organisations peu hiérarchisées et constituées d'équipes transdisciplinaires.

Une fois le cadre défini, son utilisation a été illustrée. Concernant la construction du système PLM, les travaux existants n'explicitent pas la mise en application des modèles métiers dans un logiciel PLM. Ainsi, pour compléter ces travaux, **des processus permettant de construire le métamodèle métier et le métamodèle de personnalisation du logiciel PLM** ont été proposés. Concernant les distorsions entre les différentes perceptions du produit, un travail sur les modèles servant à créer les nomenclatures de conception et les nomenclatures pour fabrication du produit a été présenté. Il a alors été défini que l'application du paradigme d'ambivalence peut être réalisée sur deux représentations du produit au moyen **de règles de cohérence support à la transformation** ou **de règles de cohérence support à la vérification**.

La construction d'un métamodèle métier et d'un métamodèle de personnalisation d'un logiciel PLM dans un **environnement de modélisation existant** a alors été présenté. L'objectif de cette construction n'était pas de fournir des métamodèles mais d'illustrer le processus de construction et de mettre en lumière l'intérêt de formaliser les métamodèles.

Perspectives

Nos perspectives de recherche visent d'une part à compléter les propositions réalisées dans ces travaux et d'autre part à faire de nouvelles études fondées sur de nouvelles hypothèses permettant de répondre au problème identifié.

Perspectives complétant les propositions de ces travaux

Concernant la poursuite des travaux réalisés, deux catégories de perspectives sont identifiables. La première catégorie est constituée de travaux « conceptuels » relatifs à la définition du cadre de modélisation et à l'application du paradigme d'ambivalence. La deuxième catégorie est constituée de travaux relatifs à la mise en œuvre du cadre dans un environnement de modélisation.

Perspectives de travaux « conceptuels »

Durant nos travaux, un manque de maturité des entreprises concernant les techniques de modélisation a été constaté. Ainsi, pour que le cadre défini puisse être utilisé par ces dernières, une première perspective identifiée consiste à définir un module de formation continue intégrant les bases de la modélisation et les résultats de nos travaux. Suite à sa mise en application au sein de plusieurs entreprises, la pertinence du cadre proposé vis à vis du problème identifié pourrait être validée.

La deuxième perspective consiste à rendre complet le cadre. Pour cela, il est nécessaire d'introduire les modèles relatifs au pôle d'observation comportemental pour le produit et pour le système PLM (cf. paradigme systémique décrit dans le chapitre 2). Les mécanismes de transformations entre le produit et ses interfaces doivent également être définis.

Pour mieux appréhender les concepts métiers permettant de construire les représentations du produit, un travail sur les ontologies relatives au PLM fait aussi partie de nos perspectives. Pour créer des modèles métiers répondant aux besoins, il est en effet nécessaire de comprendre le sens des représentations manipulées. Les questions qui se posent sont : à qui servent ces représentations ? A quoi servent ces représentations ? Une fois les concepts définis, il est possible d'identifier les caractéristiques à introduire dans les modèles métiers afin de faciliter la mise en œuvre de synergies et l'implémentation de règles de cohérence.

Perspectives de travaux relatifs à l'utilisation d'un environnement de modélisation

La mise en œuvre du cadre dans un environnement de modélisation a permis d'identifier plusieurs perspectives. La première vise à compléter l'étude réalisée par des études qui mettraient en œuvre une

expérimentation complète du cadre. Pour cela, il faudrait d'une part finaliser les transformations définies, c'est à dire le programme ATL pour transformer le modèle métier en modèle Audros et le programme AUPL pour personnaliser Audros à partir du modèle Audros en Ecore. D'autre part, il faudrait expérimenter sur un cas concret industriel le passage entre deux représentations du produit, la représentation de conception et la représentation pour fabrication par exemple. La deuxième perspective vise à expérimenter l'utilisation d'autres environnements de modélisation ou de plug-in d'EMF en intégrant des critères d'évaluation de l'environnement liés au graphisme. Enfin, la troisième perspective a pour objectif de valider l'aspect expérimentation de nos travaux. Pour cela, la généralité de la démarche proposée pour définir le métamodèle plate-forme doit être établie via des tests sur d'autres logiciels PLM. Il est également nécessaire d'identifier les problèmes pouvant être rencontrés dans le temps suite à l'évolution des métamodèles, des modèles et de l'environnement de modélisation lui-même. En effet, l'environnement de modélisation mis en oeuvre doit fonctionner à un instant « t », mais il doit également rester fonctionnel malgré les différentes évolutions listées.

Perspectives fondées sur de nouvelles hypothèses

Tout au long de ce mémoire, la multiplicité des domaines impliqués par la mise en oeuvre d'un système PLM a été mise en lumière. Dans ces perspectives, nous nous intéressons plus particulièrement à trois domaines relatifs au problème identifié.

Le premier domaine concerne la maturité des entreprises, et plus précisément, la maturité PLM. Dans ces travaux, il a été défini qu'il est nécessaire que les perceptions informationnelle et organisationnelle de l'entreprise progressent de manière corrélée. Il a également été mis en lumière que le développement de l'agilité opérationnelle nécessite l'adoption d'un raisonnement synergique entre les métiers des différentes étapes du cycle de vie. Dans le chapitre 3, des modèles de maturité PLM ont été présentés. Ces modèles déterminent les situations à atteindre pour chaque niveau de maturité identifié. Ils permettent ainsi à l'entreprise d'évaluer sa situation existante et de définir la situation souhaitée. Une fois cette évaluation et cette définition réalisées, les actions à mettre en oeuvre pour atteindre la situation souhaitée sont identifiables. Suite à des échanges avec des industriels, la méconnaissance des problèmes usuellement rencontrés dans les projets PLM a été déterminée comme un frein majeur à l'atteinte de la situation souhaitée. Ainsi, un tableau complémentaire au modèle de maturité pourrait être créé. Ce tableau recenserait les problèmes usuels rencontrés pour atteindre chaque situation identifiée et les solutions possibles pour répondre à ces problèmes. Une autre perspective relative aux modèles de maturité concerne la mise en relation de ces modèles avec les définitions génériques des besoins relatifs au système PLM. L'objectif de ces travaux consisterait à déterminer les besoins correspondant à chaque niveau de maturité. Ceci permettrait de définir l'antécédence préconisée pour la mise en oeuvre des différentes fonctions du système PLM.

Le deuxième domaine identifié concerne la construction du système PLM selon une méthode agile [Cunningham et al., 01]. Cette méthode permet de minimiser les séparations d'échelles entre les perceptions organisationnelle et informationnelle. Elle définit le système sous forme de fragments livrés de manière régulière. La définition des fragments pourrait être réalisée à partir des fonctions définies dans les définitions génériques des besoins et à partir des antécédents identifiés entre ces fonctions. Cependant, il est à noter que cette méthode implique de limiter la formalisation et la documentation. Ainsi, ses principes se révèlent antagonistes à la formalisation des modèles préconisée dans nos travaux.

Cet antagonisme se retrouve dans les objectifs de l'approche Lean (troisième domaine identifié). En effet, cette approche a pour objectif de rendre l'entreprise apprenante mais également agile. Cependant, comment réutiliser les savoirs et garantir la qualité sans un minimum de formalisation, identifié comme un frein à l'agilité ? En partant du constat que l'approche Lean a pour but l'agilité et que l'approche PLM a pour but la pérennisation des savoirs, nous proposons d'appliquer le paradigme d'ambivalence à ces deux approches afin d'identifier des synergies potentielles. Ainsi, la figure 6.1, issue de travaux préliminaires sur ce sujet, présente les buts que nous attribuons à chacune des deux approches [Pinel et al., 13] [Pinel et al., 12c]. Ces hiérarchies de buts constituent les bases pour nos futurs travaux.

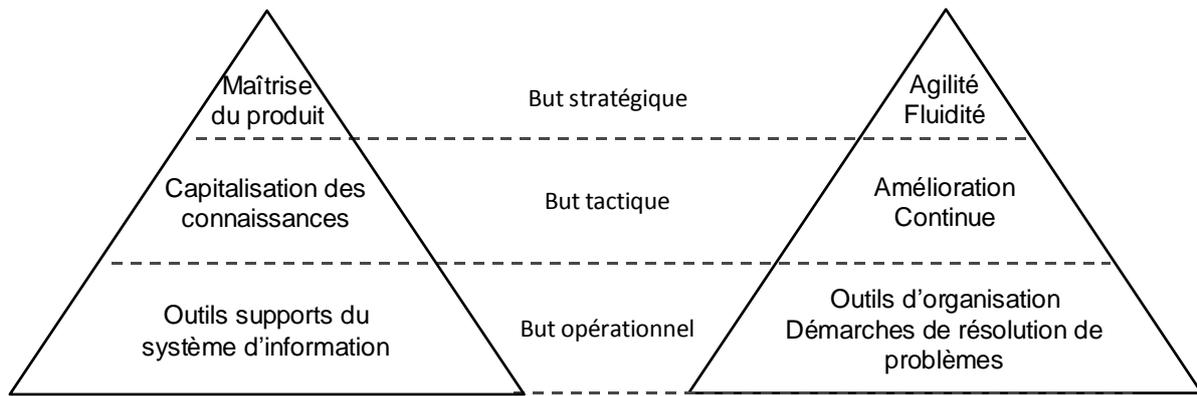


Figure 6.1 : Buts stratégiques, tactiques et opérationnels des projets Lean et PLM [Gautrot et al., 2010]

Références bibliographiques

A

- [AFIS, 12] sous la direction de Fiorèse S. et Meinadier J.-P., ouvrage collectif de l'AFIS (Association Française d'Ingénierie Système), *Découvrir et comprendre l'ingénierie système*, Ed. Cépaduès, 2012.
- [AFNOR, 91] AFNOR, *NF X50-151 - Analyse de la Valeur, Analyse Fonctionnelle*, 1991.
- [AFNOR, 87] AFNOR, *NF X50-300 - Organisation et gestion de la production industrielle - Sous-traitance industrielle - Vocabulaire*, 1987.
- [Akoka et al., 06] sous la direction de Akoka J., Comyn-Wattiau I., *Encyclopédie de l'informatique et des systèmes d'information*, Ed. Vuibert, 2006.
- [Altersohn, 97] Altersohn C., *La sous-traitance à l'aube du XXI^e siècle*, Ed. L'Harmattan, 1997.
- [Anton, 96] Anton A., *Goal-Based Requirements Analysis*, Proceeding of ICRE'96, pp. 136-144, 1996.
- [Ameri et al., 05] Ameri, F., Dutta, D., *Product Lifecycle Management: Closing the Knowledge Loops*, Computer-Aided Design & Applications, vol. 2, N° 5, pp. 577-590, 2005.
- [Artto et al., 05] Artto K., Wikström K., *What is project business ?*, International Journal of Project Management, vol. 23, N° 5, pp. 343-353, 2005.
- [Atlantic Systems Guild Inc., 05] Atlantic Systems Guild Inc., *Plan de cahier des charges et spécification des exigences non fonctionnelles avec Volere*, 2005.
http://www.volere.co.uk/pdf%20files/template_fr.pdf
- [Audibert, 09] Audibert L., *UML 2 de l'apprentissage à la pratique*, Ed. Ellipses, 2009.
- [Audros, 13] Audros, <http://www.audros.fr/cms/easysite/audros/accueil>, 2013.

B

- [Bacha, 02] Bacha R., *De la gestion des données techniques pour l'ingénierie de production – Référentiel du domaine et cadre méthodologique pour l'ingénierie des systèmes d'information techniques en entreprise*, Thèse de doctorat de l'Ecole Centrale Paris, 2002.
- [Batenburg et al., 05] Batenburg R., Helms R.W., Versendaal J., *The maturity of product lifecycle management in Dutch organizations: A strategic alignment perspective*, Proceedings of the International Conference on Product Lifecycle Management (PLM'05), pp.436-450, 2005.
- [Beck et al., 01] Kent Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., C. Martin R., Mellor S., Schwaber K., Sutherland J., Thomas D., *Manifeste pour le développement Agile de logiciels*, 2001.
Disponible sur : <http://agilemanifesto.org/iso/fr/>
- [Bell et al., 76] Bell T.E., Thayer T.A., *Software Requirements: Are They Really a Problem ?*, Proceedings ICSE-2: 2nd International Conference on Software Engineering, San Francisco, pp. 61-68, 1976.
- [Bernus et al., 05] Bernus P., Mertins K., Schmidt G., *Handbook on Architectures of Information Systems*, International Handbooks on Information Systems, Ed. Springer, 2005.

- [Bezivin et al., 02] Bézivin J., Blanc X., *MDA : vers un important changement de paradigme en génie logiciel*, Développeur Référence, 2002.
- [Bissay, 10] Bissay A., *Du déploiement d'un système PLM vers une intégration des connaissances*, Thèse de doctorat de l'Université Lyon 2, 2010.
- [Blanc, 05] Blanc X., *MDA en action, Ingénierie logicielle guidée par les modèles*, Ed. Eyrolles, 2005.
- [Bolusset, 02] Bolusset T., Oquendo F., *Formal Refinement of Software Architectures Based on Rewriting Logic*, International Workshop on Refinement of Critical Systems, 2002.
- [Bonjour, 06] Bonjour E., Dulmet M., *Pilotage des activités de conception par l'Ingénierie Système*, Chapitre 5 de l'ouvrage collectif [Roucoules et al., 06].
- [Booch et al., 00] Booch G., Rumbaugh J., Jacobson I., *Le guide de l'utilisateur UML*, Ed. Eyrolles, 2000.
- [Braesch et al., 95a] Braesch C., Haurat A., Beving J.-M., *L'entreprise-système*, Chapitre de [Braesch et al. 95b]
- [Braesch et al. 95b] Braesch C., Haurat A., *La modélisation systémique en entreprise*, Ed. Paris Hermès, 1995.
- [Brion, 11] Brion S., *Stratégie et organisation de l'innovation*, Cours école doctorale SISEO, IAE Annecy le Vieux, 2011.
- [Brown et al., 98] Brown S., Eisenhardt K., *Competing on the Edge: Strategy As Structured Chaos*, Ed. Harvard Business School Press, 1998.

C

- [Capraro et al., 02] Capraro M., Baglin G., *L'entreprise étendue et le développement des fournisseurs*, Ed. Presses universitaires de Lyon, 2002.
- [Cauvet et al., 01] Ouvrage collectif sous la direction de Cauvet C., Rosenthal-Sabroux C., *Ingénierie des systèmes d'information*, Ed. Hermès Science publ. Lavoisier, 2001.
- [Chapurlat, 07] Chapurlat V., *Vérification et validation de modèles de systèmes complexes: application à la Modélisation d'Entreprise*, Habilitation à diriger les recherches, Université de Montpellier II, 2007.
- [Chelli, 03] Chelli H., *Urbaniser l'entreprise et son système d'information – Guide des entreprises agiles*, Ed. Vuibert, 2003.
- [Chen et al., 97] Chen D., Vallespir B., Doumeings G., *GRAI integrated Methodology and its mapping onto Generic Enterprise Reference Architecture and Methodology*, Computers in Industry, vol. 33, N° 2-3, pp. 387-394, 1997.
- [Cherfi, 02] Ouvrage collectif sous la direction de Cherfi Z., *La qualité – Démarche, methods et outils*, Ed. Hermès Science publ. Lavoisier, 2002.
- [CIMData, 02] CIMData, *Product Lifecycle Management – Empowering the Future of Business*, CIMData Report, 2002.
- [CNRTL, 13] Centre National de Ressources Textuelles et Lexicales, <http://www.cnrtl.fr/>, 2013.
- [Cockburn, 00] Cockburn A., *Writing Effective Use Cases*, Ed. Addison-Wesley, 2000.
- [Constantinidis, 10] Constantinidis Y., *Expression des besoins pour le système d'information guide d'élaboration du cahier des charges*, Ed. Eyrolles, 2010.

- [Cossette, 06] Cossette C., *La publicité de A à Z: dictionnaire technique français-anglais*, Ed. Les Presses de l'Université Laval, 2006.

D

- [D'Aveni, 94] D'Aveni R., *Hypercompetition: Managing the Dynamics of Strategic Maneuvering*, Ed. New York: The Free Press, 1994.
- [Dardenne et al., 93] Dardenne A., van Lamsweerde A., Fickas S., *Goal-directed requirements acquisition*, Science of Computer Programming, vol. 20, pp. 3-50, 1993.
- [Devalan, 10] Devalan P., *Introduction à la mécatronique*, Techniques de l'Ingénieur, Référence BM8000, 2010.
- [Demoly, 10] Demoly F., *Conception intégrée et gestion d'informations techniques : application à l'ingénierie du produit et de sa séquence d'assemblage*, Thèse de doctorat de l'Université de Technologie de Belfort-Montbéliard, 2010.
- [Doumeingts, 85] Doumeingts G., *How to decentralize decisions through GRAI model in production management*, Computers in Industry, vol. 6, N° 6, pp. 501-514, 1985.
- [Dutta et al., 05] Dutta D., Wolowicz J., *An Introduction to Product Lifecycle Management*, Proceedings of the 12th ISPE International Conference on Concurrent Engineering: Research and Applications, Ft Worth/Dallas - USA, 25-29 July, 2005.

E

- [Eclipse, 13a] Eclipse, <http://www.eclipse.org/modeling/emf/>, 2013.
- [Eclipse, 13b] Eclipse, http://wiki.eclipse.org/ATL/User_Guide_-_The_ATL_Language, 2013.
- [Eclipse, 13c] Eclipse, <http://www.eclipse.org/>, 2013.
- [Eynard, 06] Eynard B., Gallet T., Roucoules L., Ducellier G., *PDM system implementation based on UML*, Mathematics and Computers in Simulation, N° 70, pp. 330-342, 2006.
- [Eynard, 04] Eynard B., Gallet T., Nowaka P., Roucoules L., *UML based specifications of PDM product structure and workflow*, Computers in Industry, vol. 55, pp. 301-316, 2004.

F

- [Favre et al., 06] Ouvrage collectif sous la direction de Favre J.-M., Estublier J., Blay-Fornarino M., *L'ingénierie dirigée par les modèles - Au-delà du MDA*, Ed. Hermès Science publ. Lavoisier, 2006.
- [Favre, 04] Favre, J.M., *Towards a Basic Theory to Model Model Driven Engineering*, Workshop on Software Model Engineering, joint event avec UML2004, Lisboa, 2004.
- [Fougeres et al., 97] Fougères A.-J., Trigano P., *Rédaction de spécifications formelles: élaboration à partir des spécifications écrites en langage naturel*, Cognito - Cahiers Romains de Sciences Cognitives, vol. 8, pp. 29-36, 1997.

G

- [Gabay, 98] Gabay J., *Merise Vers OMT et UML*, Ed. InterEditions, 1998.

- [Gautrot et al., 10] S. Gautrot, M. Pralus, L. Gzara, J. Bardet, *Lean et PLM : maîtriser la valeur ajoutée - Lean et PLM : deux piliers complémentaires pour l'entreprise ?*, Expo Progiciels, 15ème édition, Annecy-le-Vieux – France, 2010.
- [Genelot, 01] Genelot D., *Manager dans la complexité*, 3ème édition, Ed. Insep Consulting, 2001.
- [Gnahou et al., 10] Gnaho C., Semmak F., *Une extension SysML pour l'ingénierie des exigences dirigée par les buts*, Actes du 28ème congrès INFORSID, pp. 277-292, Marseille, mai, 2010.
- [Grieves, 06] Grieves M., *Product lifecycle management: driving the next generation of lean thinking*, Ed. McGraw-Hill, 2006.
- [Gzara, 00] Gzara L., *Les patterns pour l'ingénierie des Systèmes d'Information Produit*, Thèse de doctorat de l'Institut National Polytechnique de Grenoble, 2000.

H

- [Hamel et al., 90] Hamel G., Prahalad C. K., *The Core Competence of the Corporation*, Ed. Harvard Business Review, vol. 68, N° 3, pp. 79-93, 1990.
- [Huget, 10] Huget M.-P., *From models to code*, Cours école doctorale internationale – Information et Communication Technologies, Polytech' Savoie, 2010.

I

- [IEEE, 93] IEEE, St 830-1993, *Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel*, 1993.
- [IEEE, 90] IEEE, St. 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, 1990.
- [IFIP-IFAC Task Force, 99] IFIP-IFAC Task Force, *Generalised Enterprise Reference Architecture and Methodology*, version 1.6.3, 1999. Disponible sur <http://www.ict.griffith.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/GERAMv1.6.3.pdf>
- [IGL, 93] IGL, *SADT, un langage pour communiquer*, Ed. Eyrolles, Paris, 1993.
- [Iraqi-Houssaini et al., 11] Iraqi-Houssaini M., Kleiner, M., Roucoules, L.: *Model-Based (Mechanical) Product Design*. Models 2011, LNCS, vol. 6981, pp. 548-562, Ed. Springer Verlag Berlin Heidelberg, 2011.
- [ISO, 13] ISO, <http://www.iso.org/iso/fr/home/standards.htm>, 2013.
- [ISO, 10] ISO, *ISO 10303-214:2010, Industrial automation systems and integration -- Product data representation and exchange -- Part 214: Application protocol: Core data for automotive mechanical design processes*, 2010.
- [ISO, 08] ISO, *ISO/CEI 15288:2008, Systems and software engineering – System life cycle processes*, 2008.
- [ISO, 00] ISO, *ISO 10303-214:2000, Industrial automation systems and integration -- Product data representation and exchange -- Part 214: Application protocol: Core data for automotive mechanical design processes*, 2000.

J

- [Jézéquel et al., 06] Jézéquel J.-M., Gérard S., Mraidha C., Baudry B., *Le génie logiciel et l'IDM : une approche unificatrice par les modèles*, Chapitre 3 de [Favre et al., 06]

- [Jolliet et al., 05] Jolliet O., Saadé M., Crettaz P., *Analyse du cycle de vie: comprendre et réaliser un écobilan*, Ed. Presses Polytechniques et Universitaires Romandes, 2005.
- [Jun et al., 07] Jun H.-B. ; Kiritsis D.; Xirouchakis P.; *Research issues on closed-loop PLM*, Computers in Industry, vol. 58, N° 8-9, pp. 855-868, 2007.

K

- [Kadima, 05] Kadima H., *MDA, Conception orientée objet guidée par les modèles*, Ed. Dunod, 2005.
- [Kamsu Foguem et al., 06] Kamsu Foguem B., Chapurlat V., *Requirements modelling and formal analysis using graph operations*, International Journal of Production Research, vol. 44, N° 17 pp. 3451-3470, 2006.
- [Kiritsis et al., 03] Kiritsis D., Bufardi A., Xirouchakis P., *Research issues on product lifecycle management and information tracking using smart Embedded systems*, Advanced Engineering Informatics, vol. 17, N° 3-4, pp. 189-202, 2003.
- [Kleppe et al., 03] Kleppe A., Warmer S., Bast W., *MDA Explained. The Model Driven Architecture: Practice and Promise*, Ed. Addison-Wesley, 2003.
- [Kosanke et al., 99] Kosanke, K., Zelm, M., *CIMOSA modelling processes*, Computers in Industry, vol. 40, pp. 141-153, 1999.
- [Kotler, 96] Kotler P., Armstrong G., *Principles of Marketing European ed.*, 7th Edition, Ed. Prentice Hall, 1996.
- [Kuhn, 08] Kuhn T., *La Structure des révolutions scientifiques*, Ed. Flammarion, 2008.

L

- [Ladrière, 68] Ladrière J., *Représentation et connaissance, dans Encyclopedia Universalis*, 1968.
- [van Lamsweerde, 09] van Lamsweerde A., *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Ed. John Wiley & Sons, 2009.
- [van Lamsweerde, 00] van Lamsweerde A., *Requirements Engineering in the Year 00: A Research Perspective*, Proceedings 22nd Int. Conf. Software Engineering, Limerick, Ed. ACM Press, 2000.
- [Larousse, 13] Larousse, <http://www.larousse.fr/dictionnaires/francais>, 2013.
- [Le Dain et al., 07] Le Dain M.-A., Calvi R., Cheriti S., *How to evaluate the suppliers' performance in collaborative design?*, International Conference On Engineering Design, ICED'07, 28 - 31 Août, Paris, France, 2007.
- [Le Duigou, 10] Le Duigou J., *Cadre de modélisation pour les systèmes PLM en entreprise étendue - Application aux PME mécaniciennes*, Thèse de doctorat l'École Centrale de Nantes, 2010.
- [Le Moigne, 99] Le Moigne J.-L., *La modélisation des systèmes complexes*, Ed. Dunod, 1999.
- [Le Moigne, 90] Le Moigne J.-L., *La théorie du système général - Théorie de la modélisation*, 3ème édition, Ed. Presses Universitaires de France, 1990.
- [Ledeczi et al., 01] Ledeczi A., Maroti M., Bakay A., Karsai G., Garrett J., Thomason C., Nordstrom G., Sprinkle J., Volgyesi P., *The Generic Modeling Environment*, Proceedings of WISP'2001, May, Budapest, Hungary, 2001.
Plus d'informations sur : <http://www.isis.vanderbilt.edu/projects/GME>

- [Lombard, 06] Lombard M., *Contribution de la Modélisation Informationnelle aux Processus de Conception et Réalisation de Produits Manufacturiers : vers une Ontologie Métier*, Habilitation à Diriger des Recherches, Centre de Recherche en Automatique de Nancy, 2006.
- [Lycett et al., 04] Lycett M., Rassau A., Danson J., *Programme management: a critical review*, International Journal of Project Management, vol. 22, N° 4, pp. 289-299, 2004.
- [Lyonnet, 10] Lyonnet B., *Amélioration de la performance industrielle : vers un système de production Lean adapté aux entreprises du pôle de compétitivité Arve Industries Haute-Savoie Mont-Blanc*, Thèse de doctorat de l'Université de Savoie, 2010.

M

- [Mallek et al., 11a] Mallek S., Daclin N., Chapurlat V., *Formalisation and verification of interoperation requirements on collaborative processes*, Proceedings of IFAC World Congress, Milano, 2011.
- [Mallek et al., 11b] Mallek S., Daclin N., Chapurlat V., *An approach for interoperability requirements specification and verification*, Proceedings of International Working Conference on Enterprise Interoperability, Stockholm, 2011.
- [Mayrhofer, 06] Mayrhofer U., *Marketing*, 2ème édition actualisée, Ed. Bréal, 2006.
- [Mc Call et al., 77] McCall J., Richards P., Walters G., *Factors in software quality*, Rome Air Development Center, 1977.
- [Maurino, 94] Maurino M., *La gestion des données techniques: technologie du concurrent engineering*, Ed. Masson, 1994.
- [Meinadier, 98] Meinadier J.-P., *Ingénierie et integration des systèmes*, Ed. Hermès, 1998.
- [Melèse, 90] Melèse J., *Approches systémiques des organisations, vers l'entreprise à complexité humaine*, Ed. de l'Organisation, Paris, 1990.
- [Merminod, 07] Merminod V., *TIC et performance du développement de nouveaux produits : le cas de PLM au sein du Groupe SEB*, Thèse de doctorat l'Université de Savoie, 2007.
- [Merlo et al., 05] Merlo C., Eynard B., Girard P., Odinet A., Gallet T., *Compared implementations of PDM systems based on UML specifications*, Int. J. Product Lifecycle Management, vol. 1, N° 1, pp. 52-69, 2005.
- [Merlo et al., 04] Merlo C., Girard P., *Information system modelling for engineering design coordination*, Computers in Industry, vol. 55, pp. 317-334, 2004.
- [Morenton, 06] Morenton P., *CAO : logiciel CATIA*, Techniques de l'Ingénieur, Référence AG2535, 2006.
- [Morin, 80] Morin E., *La méthode, t. 2 : La vie de la vie*, Paris, Ed. du Seuil, 1980.
- [Morley, 08] Morley C., *Management d'un projet système d'information, Principes, techniques, mise en oeuvre et outils*, 6ème édition, Ed. Dunod, 2008.
- [Mylopoulos et al., 99] Mylopoulos J., Chung L., Yu E., *From object-oriented to goal-oriented requirements analysis*, Communications of the ACM, vol. 42, pp. 31-37, 1999.

N

- [Nanci et al., 01] Nanci D., Espinasse B., *Ingénierie des systèmes d'information : Merise deuxième génération*, 4ème édition, Ed. Vuibert, 2001.

- [Nebut et al., 05] Nebut C., Fleury F., *Une méthode de formalisation progressive des exigences basée sur un modèle simulable*, L'objet, vol. 11, fleury.com, pp. 145-158, 2005.
- [Noël et al., 08] Noël F., Roucoules L., *The PPO design model with respect to digital enterprise technologies among product life cycle*, International Journal of Computer Integrated Manufacturing, vol. 21, N° 2, pp. 139-145, 2008.
- [Noran, 03] Noran O. S., *An analysis of the Zachman Framework for Enterprise Architecture from the GERAM perspective*, Annuals Reviews in Control, vol. 27, N° 2, pp. 163-183, 2003.
- [Nuseibeh et al., 00] Nuseibeh B., Easterbrook S., *Requirements engineering: a roadmap*, Proceedings of the Conference on The Future of Software Engineering ICSE'00, pp. 35-46, 2000.

O

- [OMG, 12] OMG (Object Management Group), *Systems Modeling Language*, version 1.3, <http://omg.org/spec/SysML/1.3/PDF>, 2012.
- [OMG, 11] OMG (Object Management Group), *Product Lifecycle Management Services*, version 2.1, <http://www.omg.org/spec/PLM/2.1/>, 2011.

P

- [Paviot, 10] Paviot T., *Méthodologie de résolution des problèmes d'interopérabilité dans le domaine du Product Lifecycle Management*, Thèse de doctorat de l'École Centrale Paris, 2010.
- [Perrotin et al., 99] Perrotin R., Loubère J.M., *Nouvelles stratégies d'achat: sous-traitance, coopération, partenariat?*, Ed. d'Organisation, 1999.
- [Pinel et al., 13] Pinel M., Braesch C., Pralus M., Tabourot L., Bonnal P., *Lean et PLM : deux piliers complémentaires à l'agilité de l'entreprise*, The journal of modern project management, vol. 1, pp. 67-82, 2013. Article sélectionné suite à [Pinel et al., 12c].
- [Pinel et al., 12a] Pinel M., Braesch C., Tabourot L., Berger A., *Operational framework based on modeling languages to support product repository implementation*, Product Lifecycle Management Towards Knowledge-Rich Enterprises, pp. 257-266, Ed. Springer, 2012. Article sélectionné suite à [Pinel et al., 12b].
- [Pinel et al., 12b] Pinel M., Braesch C., Tabourot L., Berger A., *Operational framework based on modeling languages to support product repository implementation*, 9ème International Conference On Product Lifecycle Management, Montréal – Canada, 2012.
- [Pinel et al., 12c] Pinel M., Braesch C., Pralus M., Tabourot L., Bonnal P., *Lean et PLM : deux piliers complémentaires à l'agilité de l'entreprise*, Conférence Internationale en Gestion De Projet, Trois-Rivières – Canada, 2012.
- [Pinel et al., 11a] Pinel M., *Cadre opérationnel basé sur MDA (Model Driven Architecture) pour la mise en œuvre du référentiel produit*, Journées STP du GDRMACS, GT Easy-DIM, Tarbes, 2011.
- [Pinel et al., 11b] Pinel M., Genez C., Berger A., Gzara L., Braesch C., *L'approche PLM, comment remettre le produit au coeur de l'entreprise?*, Journal Technologique des Pays de Savoie (Jitec), N°150, 2011.
- [Pinel et al., 11c] Pinel M., Gzara L., Pralus M., Braesch C., Bignon J., Tabourot L., *PLM and Lean: two complementary pillars for the company*, International Conference

on Industrial Engineering and Systems Management (IESM 2011), Metz – France, 2011.

- [Pingaud, 05] Pingaud, H., *Modélisation d'entreprise*, Cours Master Productique de l'Université Paul Sabatier de Toulouse, 2005.
- [Pol et al., 05] Pol G., Jared G., Merlo C., Legardeur J., *Prerequisites for the implementation of a product data and process management tool in SME*, International Conference on Engineering Design, ICED'05, Australie, 2005.
- [Polizzi, 04] Polizzi M., *La chaîne numérique*, Revue Technologie, N° 134, novembre-décembre, 2004.
- [Porter, 03] Porter M., *L'avantage concurrentiel*, Ed. Dunod, 2003.

R

- [Reix, 11] Reix R., *Systèmes d'information et management des organisations*, 6ème édition, Ed. Vuibert, 2011
- [Rieu, 99] Rieu D., *Ingénierie des Systèmes d'Information*, Mémoire d'Habilitation à Diriger les Recherches, Institut National Polytechnique de Grenoble, 1999.
- [Rolland et al., 01] Rolland C., Grosz G., *De la modélisation conceptuelle à l'ingénierie des besoins*, Chapitre 4 de l'ouvrage collectif [Cauvet et al., 01].
- [Rolland, 03] Rolland C., *L'ingénierie des Besoins : L'Approche L'Ecriteiro*, Journal Techniques de l'Ingénieur (TI), 2003.
- [Roques, 09] Roques P., *UML 2 par la pratique études de cas et exercices corrigés*, 7ème édition, Ed. Eyrolles, 2009.
- [Roucoules et al., 06] Ouvrage collectif sous la direction de Roucoules L., Yannou B., Eynard B., *Ingénierie de la conception et cycle de vie des produits (Traité IC2, série Productique)*, Ed. Hermès Science publ. Lavoisier, 2006.
- [Rumbaugh et al., 04] Rumbaugh J., Jacobson I., Booch G., *UML2 : Guide de référence*, Ed. CampusPress, Paris, 2004.

S

- [Saaksvuori et al., 08] Saaksvuori A., Immonen A., *Product Lifecycle Management*, 3ème édition, Ed. Springer, 2008.
- [Sherehiy et al., 07] Sherehiy B., Karwowski W., K. Layer J., *A review of enterprise agility: Concepts, frameworks, and attributes*, International Journal of Industrial Ergonomics, vol. 37, pp. 445-460, 2007.
- [Söderlund, 04] Söderlund J., *On the broadening scope of the research on projects: a review and a model for analysis*, International Journal of Project, vol. 27, N° 8, pp. 655-667, 2004.
- [Sommerville et al., 97] Sommerville I., Sawyer P., *Requirements Engineering: A Good Practice Guide*, Ed. Wiley & Sons, 1997.
- [Sommerville, 92] Sommerville I., *Le génie logiciel*, Ed. Addison-Wesley France, 1992.
- [Standish, 95] The Standish Group, *The Standish Group report Chaos*, <http://www.projectsmart.co.uk/docs/chaos-report.pdf>, 1995.
- [Steinberg, 08] Steinberg D., *Fundamentals of the Eclipse Modeling Framework*, 2008. http://www.eclipse.org/modeling/emf/docs/presentations/EclipseCon/EclipseCon2008_309T_Fundamentals_of_EMF.pdf
- [Steinberg, 07] Steinberg D., *Fundamentals of the Eclipse Modeling Framework*, 2007.

<http://www.eclipse.org/modeling/emf/docs/presentations/EclipseWorld/EclipseWorld2007.pdf>

- [Stark, 05] Stark J., *Product Lifecycle Management: 21st century paradigm for product realisation*, Ed. Springer, 2005.
- [Sud et al., 03] Sud R., Arthur J., *Requirements Management Tools - A Qualitative Assessment*, Department of Computer Science report, Virginia Tech, Blacksburg, VA 24060 USA, 2003.
- [Sudarsan et al., 05] Sudarsan R., Fenves S. J., Sriram R. D., Wang F., *A product information modeling framework for product lifecycle management*, Computer-Aided Design, pp. 1399-1411, 2005.

T

- [Terzi et al., 07] Terzi S., Ball P.D., Bouras A., Dutta D., Garetti M., Gurumoorthy B., Han, S., Kiritsis D., *A new point of view on Product Lifecycle Management*, Proceedings of the 5th International Conference on Product Lifecycle Management PLM'08, Seoul - South Korea, 9-11 July, 2007.
- [Terzi, 05] Terzi S., *Gestion du Cycle de Vie des Produits : Définitions, Problèmes Ouverts et Modèles de Référence*, Thèse de doctorat de l'Université Henri Poincaré, Nancy-I et de Politecnico di Milano, 2005.
- [Tysebaert, 01] Tysebaert J.-M., *Les logiciels de gestion hautement intégrés: préparation par l'ingénierie de métier*, Ed. Technip, 2001.

U

- [Ungerer, 02] Ungerer M., Buchanan K., *Usage Guide for the STEP PDM Schema V1.2*, 2002. http://www.steptools.com/support/stddev_docs/express/pdm/index.html

V

- [Vallespir et al., 03] Vallespir B., Braesch C., Chapurlat V., Cretani D., *L'intégration en modélisation d'entreprise : les chemins d'UEML*, 4ème Conférence Francophone en MOdélisation et SIMulation MOSIM'03, Toulouse – France, pp. 140-145, 2003.
- [Vanwormhoudt, 12] Vanwormhoudt G., *Vérification de modèles avec EMF et OCL*, documentation en ligne sur le site EMF OCL Plugin, 2012. <http://www.telecom-lille1.eu/people/Vanwormhoudt/siteEMFOCL/documents/EMFOCLpresentation.pdf>
- [Vernette, 91] Vernet E., *Marketing fondamental*, Ed. Eyrolles Université, 1991.
- [Vicien et al., 11] Vicien G., Merlo C., *Modélisation pour l'étude de l'interopérabilité d'entreprise en conception de produits*, 9e Congrès International de Génie Industriel, Canada, 2011.
- [Visual Paradigm, 13] Visual Paradigm, <http://www.visual-paradigm.com/product/compare/>, 2013.

W

- [Wieggers, 03] Wieggers K., *Software Requirements*, 2nd edition, Ed. Microsoft Press, 2003.

- [Wiegers, 00] Wieggers K., *When Telepathy Won't Do: Requirements Engineering Key Practices*, Cutter IT Journal, 2000.
dispo sur <http://www.processimpact.com/articles/telepathy.html>
- [Williams, 94] Williams T., *The Purdue Enterprise Reference Architecture and Methodology (PERA)*, Computers in Industry, vol. 24, Issues 2-3, pp. 141-158, 1994.
- [Win Design, 13] Win Design, <http://www.win-design.com/fr/index.htm>, 2013.

Z

- [Zachman, 08] Zachman J., <http://www.zachman.com/about-the-zachman-framework>, 2008.
- [Zancul, 09] Zancul E., *Gestão do ciclo de vida de produtos: seleção de sistemas PLM com base em modelos de referência*, Thèse de doctorat de l'Escola de Engenharia de São Carlos, en portugais, 2009.
- [Zhang, 11] Zhang D., *Towards theory building in agile manufacturing strategies - Case studies of an agility taxonomy*, International Journal of Production Economics, vol. 131, pp. 303-312, 2011.

Introduction du cycle de vie produit dans les entreprises de sous-traitance comme vecteur d'agilité opérationnelle et de maîtrise du produit

Résumé

Pour faire face à un environnement en perpétuelle évolution, les entreprises doivent changer et parfois en profondeur. Ces évolutions en principe voulues et contrôlées se font au moyen de projets dits « d'entreprise ». Parmi les buts qu'il s'agit d'atteindre par le biais de ces projets, deux buts majeurs sont identifiables : le développement de l'agilité opérationnelle et la maîtrise des produits. Dans ces travaux de thèse, nous nous focalisons sur le projet PLM (Product Lifecycle Management) et plus particulièrement sur la mise en œuvre du système d'information nécessaire à la gestion du cycle de vie des produits : le système PLM. Ce système d'information coordonne tout ou partie des informations liées à la définition, à la réalisation, à l'usage et au retrait des produits. Après un état de l'art permettant de définir de façon précise les concepts liés à la gestion du cycle de vie des produits, une méthode est proposée pour définir le cahier des charges du système PLM. La définition de cette méthode montre la nécessité d'assurer la cohérence entre les différents modèles du système PLM d'une part et entre les différentes représentations du produit utilisées tout au long de son cycle de vie d'autre part. Un cadre de modélisation basé sur le paradigme systémique, sur le paradigme d'ambivalence et sur des concepts de métamodélisation est alors proposé. Ce cadre de modélisation apporte des éléments de réponse aux besoins de cohérence identifiés. Il supporte également l'adoption du raisonnement synergique indispensable au développement de l'agilité opérationnelle recherchée par l'entreprise. Une expérimentation est réalisée pour illustrer les concepts introduits dans notre cadre de modélisation.

Mots Clés

Gestion du cycle de vie produit, paradigme d'ambivalence, cadre de modélisation, ingénierie dirigée par les modèles

'Product Lifecycle Management' (PLM) in the subcontracting industry as a key for operational agility and product management

Abstract

Faced with a constantly evolving environment, companies have to change and sometimes have to change in depth. These evolutions are usually intentional and monitored and they are done through business projects. Two major goals can be identified among the goals to be reached through these projects: the operational agility development and the product management expertise. This thesis work focuses on PLM (Product Lifecycle Management) project and more precisely on the implementation of the information system needed to manage products' lifecycle : the PLM system. This information system manages all or part of information related to the definition, the manufacturing, the use and the treatment as a waste of products. First of all, a state of the art of the concepts related to product lifecycle management is done. Basing on these concepts, a method for defining the PLM system requirements is proposed. Defining this method highlights two consistency needs. The first one is related to consistency among the several PLM system models. The second one is related to consistency among the several product representations used throughout the product lifecycle. A modeling framework based on the systemic paradigm, on the ambivalence paradigm and on metamodeling concepts is then proposed. On the one hand, this modeling framework provides elements to respond to identified consistency needs. On the other hand, it supports the adoption of the synergistic reasoning that is essential for developing the operational agility sought by the enterprise. An experimentation is carried out to illustrate the concepts introduced by the modeling framework.

Keywords

Product Lifecycle Management, ambivalence paradigm, modeling framework, Model Driven Engineering