

INTEROPERABILITE DIRIGEE PAR LES MODELES :Une Approche Orientée Produit pour l'interopérabilité dessystèmes d'entreprise.

Salah Baïna

▶ **To cite this version:**

Salah Baïna. INTEROPERABILITE DIRIGEE PAR LES MODELES :Une Approche Orientée Produit pour l'interopérabilité dessystèmes d'entreprise.. domain_stic.gest. Université Henri Poincaré - Nancy I, 2006. Français. NNT : . tel-00123271

HAL Id: tel-00123271

https://theses.hal.science/tel-00123271

Submitted on 9 Jan 2007

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.

Nancy-Université
Université Henri Poincaré

FACULTE DES SCIENCES & TECHNIQUES

U.F.R. Sciences et Techniques Mathématiques, Informatique et Automatique
Ecole Doctorale IAEM Lorraine
Département de Formation Doctorale Automatique

Thèse

Présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy I

en Automatique, Traitement du Signal, Génie Informatique

par **Salah BAÏNA**

INTEROPERABILITE DIRIGEE PAR LES MODELES :

Une Approche Orientée Produit pour l'interopérabilité des systèmes d'entreprise.

Soutenue publiquement le 07/12/2006 devant le jury composé de :

Rapporteurs :	M. Bruno VALLESPIR	Professeur, Université Bordeaux 1
	M. Flavio OQUENDO	Professeur, Université de Bretagne Sud
Examinateurs :	M. Jean-Pierre BOUREY	Professeur, Ecole Centrale de Lille
	M. Gérard MOREL	Professeur, UHP Nancy I
	M. Hervé PANETTO	Maître de Conférences, HDR, UHP Nancy I
	M. Khalid BENALI	Maître de Conférences, HDR, Université Nancy 2

Centre de Recherche en Automatique de Nancy
UMR 7039, Nancy-Université, CNRS
Faculté des sciences et Techniques – 54500 Vandoeuvre lès Nancy

Dédicace

Jamais cette thèse n'aurait vu le jour, si dès ma plus tendre enfance mes parents ne m'avaient inculqué l'envie de savoir, la curiosité, la répartie et l'argumentation. Aussi, je ne cesserai jamais de rappeler toute la gratitude que je leur dois et tout l'amour et l'admiration que je leur porte.

Je tiens aussi à exprimer ma gratitude à l'ensemble de ma famille, petits et grands, eux dont les mots ont su me redonner l'envie de sourire à la vie quand les maux de la thèse me donnaient envie de pleurer.

A celle dont les SMS m'ont réchauffé le cœur pendant les longues nuits d'hiver, d'automne, de printemps et d'été.

"*Etudie, et quand tu seras grand tu pourras acheter tout ce dont tu as envie.*"

M^e^ Abdelkhaleq Mohammed BAÏNA
Juge d'instruction et Avocat à la cour,
Eternellement mon père.

Je vous aime tous. Que dieu vous préserve.

Remerciements

Je tiens à remercier tout d'abord mes directeurs de thèse Monsieur le Professeur Gérard MOREL et Monsieur Hervé PANETTO pour avoir dirigé ce travail. Leur expérience et leurs grandes compétences ont permis l'accomplissement de ce travail.

Un merci particulier pour la qualité de leur collaboration, leurs nombreux conseils, et leur aide constante et pour la façon efficace avec laquelle ils ont suivi ce travail. J'ai beaucoup appris à leur contact, pendant les nombreuses heures passées ensemble.

Pour ses précieux conseils de tout ordre, sa disponibilité et sa confiance, je remercie tout particulièrement Monsieur Khalid BENALI, qui a su mettre en œuvre toute son expérience pour apporter une vision particulière et non moins importante au travail présenté dans ce mémoire. Qu'il trouve ici les marques de ma reconnaissance et de mon respect.

Cette thèse a été réalisée au Centre de Recherche en Automatique de Nancy, tout naturellement je tiens donc à remercier son directeur, Monsieur le Professeur Alain RICHARD ainsi que Monsieur le Professeur Thierry DIVOUX responsable du groupe thématique SYMPA pour toute l'attention qu'ils m'ont portée et pour les moyens mis à ma disposition durant ces trois années.

Je remercie également Messieurs le Professeur Bruno VALLESPIR responsable du groupe GRAI au sein du laboratoire LAPS à l'Université de Bordeaux 1, ainsi que le Professeur Flavio OQUENDO responsable du groupe ARCHLOG au sein du laboratoire VALORIA à l'Université de Bretagne Sud d'avoir accepté de participer à ce jury en tant que rapporteurs, je leur exprime toute ma reconnaissance pour l'intérêt porté à ce travail.

Je suis, également, très sensible à l'honneur que m'a fait Monsieur le Professeur Jean-Pierre BOUREY, en acceptant de participer à ce jury.

Un grand merci à l'ensemble des membres du CRAN et de l'AIP-PRIMECA qui ont contribué de prés ou de loin à l'aboutissement de ces travaux.

Enfin, je remercie l'ensemble de mes amis doctorants du CRAN ou d'ailleurs et je leur souhaite bonne chance pour la suite de leurs travaux.

Cette liste ne saurait être exhaustive, je remercie, donc, tout ceux dont les noms n'apparaissent pas sur cette page, et je m'excuse auprès d'eux.

Spéciale dédicace à:

- *La communauté maghrébine du CRAN: Belynda, Hind, Zied, Ahmed et Abdel.*
- *Les plus nancéens des cachanais: Dominique, Pierre et Nicolas.*
- *Mon guitariste préféré: Maxime M.*
- *La connexion syrienne: Salah et Khaled.*
- *Les citoyens du Liban vert: Ramy et Rony.*
- *Sans oublier: Angela, David, Edouard, Jean-Philippe, Rémi et Thomas.*

"Nous devons apprendre à vivre ensemble comme des frères, sinon nous allons mourir tous ensemble comme des idiots."

Martin Luther King

Préambule

Les travaux présentés dans ce mémoire s'inscrivent dans le cadre de ma thèse réalisée au sein du laboratoire CRAN[1] (Centre de Recherche en Automatique de Nancy, UMR 7039), plus précisément, dans l'équipe-projet *Systèmes Contrôlés par le Produit* (SCP) du groupe thématique ***Systèmes de production ambiants*** **(SYMPA).**

Cette équipe-projet est articulée autour de 4 actions portant sur la modélisation, la conception et l'évaluation d'architectures d'automatisation nécessairement nouvelles, pour le contrôle par le produit de divers types d'applications.

Dans ce contexte, l'objectif de cette thèse est d'étudier les différentes facettes de l'interopérabilité des systèmes d'entreprise, en prenant en considération le produit comme élément central dans la modélisation de ces systèmes.

En effet, les productions de biens (secteurs primaire et secondaire) et de services (secteur tertiaire) ont longtemps été clairement séparées. Mais aujourd'hui, le produit tend à devenir un bien intégrant un ou plusieurs services permettant : son usage, sa maintenance, son interopérabilité avec son environnement (usager, système de production, …) et sa traçabilité tout en intégrant celle de ses composants.

Cette thèse s'inscrit aussi dans le cadre de la participation du CRAN dans le réseau d'excellence INTEROP Noe[2] (Interoperability Research for Networked Enterprises Applications and Software) supporté par la commission européenne.

[1] *http://www.cran.uhp-nancy.fr/*
[2] *http://www.interop-noe.org*

Table des matières

INTRODUCTION GENERALE 19

I. PROBLEMATIQUE 21
II. ORGANISATION DU MANUSCRIT 30

CHAPITRE 1 : INTEROPERABILITE DES SYSTEMES D'ENTREPRISE 33

I. INTRODUCTION 35
II. L'INTEROPERABILITE DANS L'ENTREPRISE 36
II.1. DEFINITIONS 37
II.2. SOLUTIONS PRAGMATIQUES POUR L'INTEROPERABILITE EN ENTREPRISES 38
II.2.1. Interopérabilité ERP/MES 39
II.2.2. Interopérabilité SCM/CRM 40
II.2.3. Interopérabilité des systèmes de gestion de workflow 41
II.3. CLASSIFICATION ET TYPES D'INTEROPERABILITE 42
II.3.1. La classification IEC 62390 42
II.3.2. Le modèle LISI 44
II.3.3. Le modèle LCIM 46
II.3.4. Illustration 47
II.4. SYNTHESE 48
III. LA MODELISATION DANS L'ENTREPRISE 49
III.1. MODÈLES ET META-MODÈLES 49
III.2. INTRODUCTION A LA MODELISATION DANS L'ENTREPRISE 51
III.3. LES DIFFERENTS TYPE DE MODELES 53
III.4. SYNTHESE 58
IV. MODELISATION ET INTEROPERABILITE 59
IV.1. L'INTEROPERABILITE PAR INTEGRATION DE MODELES 59
IV.2. L'INTEROPERABILITE PAR TRANSFORMATION DE MODELES 60
IV.3. ONTOLOGIES D'ENTREPRISE, UN OUTIL POUR L'INTEROPERABILITE 63
IV.3.1. Définitions 63
IV.3.2. Ontologies et Interopérabilité dans l'entreprise 64
IV.4. SYNTHESE 66
V. CONCLUSION 68

CHAPITRE 2 : UN MODELE DE REFERENCE POUR LA REPRESENTATION DU PRODUIT 71

I. INTRODUCTION 73
II. REPRESENTATION DES INFORMATIONS DU PRODUIT 74
II.1. LE « PRODUIT INTELLIGENT » 74
II.2. EXEMPLES DE REPRESENTATION DU PRODUIT 75
II.3. SYNTHESE 77
III. UN MODELE DE REFERENCE POUR LES INFORMATIONS DU PRODUIT. 79
III.1. L'HOLON, UN MODELE POUR LE PRODUIT INTELLIGENT 80
III.2. BWW : UNE BASE POUR LA REPRESENTATION DES CHOSES ET DES OBJETS 82
III.2.1. Chose et Construct 83
III.2.2. Classe et catégorie 84
III.2.3. Propriétés et Attributs et états 84
III.2.4. Propriétés émergentes et propriétés héritées 85
III.3. CONCEPTS ET ASPECTS LIES A L'HOLON PRODUIT 86
III.3.1. La composition structurelle 88
III.3.2. La description des caractéristiques 90
III.3.3. L'état d'un holon 91
III.3.4. Les flux d'holon 91
III.4. UN META-MODELE POUR L'HOLON-PRODUIT 93
III.5. L'HOLON DANS SON ENVIRONNEMENT 95
IV. CONCLUSION 100

CHAPITRE 3 : L'APPROCHE HOLONIQUE POUR LA MODELISATION DU PRODUIT 101

I. INTRODUCTION 103
II. IMPLEMENTATION ET PROTOTYPAGE 104
II.1. INTRODUCTION A MEGA 104
II.2. MODELISATION HOLONIQUE DANS MEGA 105
II.2.1. Modification du meta-modèle MEGA 105
II.2.2. Une interface graphique pour les nouveaux concepts 109
III. L'APPROCHE HOLONIQUE SELON LE CADRE DE MODELISATION ZACHMAN 110
III.1. INTRODUCTION AU CADRE ZACHMAN 110
III.2. LA MODELISATION HOLONIQUE ET ZACHMAN 112

IV. MISE EN APPLICATION DE LA MODELISATION HOLONIQUE 114
IV.1. OBJECTIF ET CONTEXTE 114
IV.2. MODELISATION HOLONIQUE DE L'INSTALLATION 116
IV.2.1. Problématique de la traçabilité dans l'entreprise 116
IV.2.2. Méthode de travail 117
IV.2.3. L'approche appliquée 118
IV.3. VERS UN SYSTEME GESTION DE TRAÇABILITE 121
V. CONCLUSION 130

CHAPITRE 4 : L'APPROCHE DIRIGEE PAR LES MODELES POUR UNE INTEROPERABILITE ORIENTEE PRODUIT 131

I. INTRODUCTION 133
II. INTEROPERABILITE ET TRANSFORMATION DE MODELES 134
II.1. MAPPINGS DE META-MODÈLES 135
II.2. LES REGLES DE TRANSFORMATIONS 137
II.3. VERS UNE CLASSIFICATION INTEROPERABILITE SEMANTIQUE 139
III. APPLICATION 141
III.1. UEML 142
III.2. IEC 62264 145
III.3. SYNTHESE 147
III.4. IMPLEMENTATION DES MAPPINGS DANS MEGA 148
IV. L'APPROCHE HOLONIQUE ET LES MAPPINGS DANS LE CADRE ZACHMAN 152
V. MISE EN OEUVRE 154
VI. CONCLUSION 172

CONCLUSIONS ET PERSPECTIVES 175

I. CONCLUSION GENERALE 177
II. BILAN DE L'APPROCHE 178
III. PERSPECTIVES 182

BIBLIOGRAPHIE 183

ANNEXES 195

Table des figures

Figure 1. Environnement de l'entreprise étendue (Alban 1996). 22
Figure 2. Intégration chaotique dans la constellation des systèmes d'entreprises. 23
Figure 3. Structure classique des entreprises manufacturières 24
Figure 4. Séparation des pôles Manufacturing et Business (Baïna and Morel 2006) 25
Figure 5. Flux de matière et d'information 27
Figure 6. Vue synthétique sur les chapitres et leurs interconnexions. 32
Figure 7. Les différentes niveaux de communication selon (Euzenat 2001). 36
Figure 8. Exemple de données échangées entre le niveau ERP et le niveau MES. 39
Figure 9. Niveaux de Compatibilité selon IEC 62390 (IEC 62390 2005) 42
Figure 10. Les cinq niveaux d'interopérabilité du modèle LISI (LISI 1998). 45
Figure 11. Modèle de référence de LISI (LISI 1998) 46
Figure 12. Classification de l'interopérabilité orientée données (Tolk and Muguira 2003) ... 47
Figure 13. Évolution de la modélisation dans le temps (Janis 2005). 52
Figure 14. Les différents types de modèles d'entreprise selon Gustas (Gustas 1995) 55
Figure 15. L'approche MDA et les quatres niveaux ontologiques 57
Figure 16. Construction des systèmes dans l'ingénierie MDE (Bézivin 2005). 61
Figure 17. Transformation de meta-modèle (Lemesle 1998). 62
Figure 18. Interopérabilité avec modèle de référence. 68
Figure 19. Réseau de communication avec sans modèle de référence. 79
Figure 20. Modèle initial de l'holon en tant qu'agrégation physique/informationnelle. 80
Figure 21. L'holon, interface entre le produit et les systèmes ambiants. 81
Figure 22. Formalisation en NIAM du meta-modèle de BWW. 86
Figure 23. Schéma simplifié du procédé de fabrication à l'AIPL. 87
Figure 24. Produits et pièces de l'AIPL. 88
Figure 25. Exemple de nomenclature lié au produit AIPL de type 60-88-11-10. 89
Figure 26. Modèle de composition des holons. 89
Figure 27. Exemple des différents types d'information autour du produit. 90
Figure 28. Formalisation des différents types de flux et leur contenu.(Baïna et al. 2005) 93
Figure 29. Meta-modèle holonique pour la représentation du produit. 94
Figure 30. Exemple de processus holonique. 95

Figure 31. Définition des interfaces nécessaires à l'exécution d'un processus..... 98
Figure 32. Meta-classes des concepts relatifs à un diagramme de processus MEGA..... 106
Figure 33. Meta-modèle holonique tel que implémenté dans l'environnement MEGA. 108
Figure 34. Etapes pour l'utilisation de la modélisation holonique selon le cadre Zachman. 113
Figure 35. Objectif et principe de la traçabilité. 115
Figure 36. Evolution du processus de fabrication de farine dans le moulin étudié. 117
Figure 37. Description de la circulation des flux sur le terrain. 118
Figure 38. Exemple de schéma de traitement de données produit sous OSSAD. 121
Figure 39. Extrait de la modélisation holonique réalisée dans l'environnement MEGA..... 123
Figure 40. Exemple d'utilisation des holons pour la représentation du type d'un produit. .. 124
Figure 41. Diagramme de classes issu de la modélisation holonique. 125
Figure 42. Schéma relationnel de la base de traçabilité. 126
Figure 43. Transformation de meta-modèle (Lemesle, 1998) 134
Figure 44. Exemple de signature partiellement ordonnée d'une ontologie 136
Figure 45. Illustrations des correspondances entre les sémantiques des composants 138
Figure 46. Mapping bijectif entre deux vocabulaires distincts A et B 139
Figure 47. Mappings distincts non bijectifs entre deux vocabulaires distincts A et B..... 139
Figure 48. Structure des entreprises de production 141
Figure 49. Meta-modèle UEML (UEML 2002) 144
Figure 50. Aspects de l'intégration "Business to Manufacturing" dans l'IEC 62264 145
Figure 51. Le modèle matériel l'IEC 62264..... 146
Figure 52. Mécanisme de traduction basé sur les mappings 149
Figure 53. Extrait du schéma DTD des fichiers XML générés par MEGA 150
Figure 54. Interface graphique pour l'application des règles de transformations XSLT. 150
Figure 55. Exemple de règles XSLT réalisées à partir des mappings vers l'IEC 62264. 151
Figure 56. Positionnement de l'approche d'interopérabilité sur le cadre de modélisation Zachman 153
Figure 57. Intégration en pyramide des applications de l'AIPL 154
Figure 58. La séparation des pôles « manufacturing » et « business » appliquée au contexte AIPL. 155
Figure 59. Point de départ pour la modélisation de l'AIPL 156

Figure 60. Vue générale du processus de fabrication de l'AIPL..... 157
Figure 61. Exemple de flux d'holons échangés sur la cellule d'assemblage des produits.... 159
Figure 62. Exemple de l'identification des informations (propriété/attribut) relatives aux produits échangés dans l'environnement AIPL. 160
Figure 63. Exemple de représentation holonique du produit « Prod 01,09 »..... 161
Figure 64. Extrait du modèle de données FLEXNET pour la représentation du produit..... 163
Figure 65. Extrait du modèle de données ADONIX pour la représentation du produit..... 164
Figure 66. La modélisation holonique pour l'ingénierie B2M du produit 165
Figure 67. Illustration des différentes étapes suivies lors de l'application de l'approche holonique dans le cadre de l'environnement AIPL. 167
Figure 68. Illustration du déroulement de l'ensemble de la démarche à travers les différentes sections de l'entreprise. 168
Figure 59. Instanciation des éléments *MaterialClass* et *MaterialClassProperty* correspondant à la classe du produit « Prod01,09 »..... 169
Figure 70. Instanciation d'un holon en utilisant les valeurs effectives correspondant à une instance du produit « Prod 01,09 »..... 170
Figure 71. Instanciation du *MaterialLot* à partir des données de l'holon « Prod 01,09 »..... 171
Figure 72. Instanciation d'un élément *MaterialSubLot* à partir des données de l'holon « Prod01,09 ». 171

Glossaire

APS	Advanced Planning System
CAO	Conception Assistée par Ordinateur
CFAO	Conception et Fabrication Assistées par Ordinateur
CRM	Customer Relationship Management
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
GMAO	Gestion de la Maintenance Assistée par Ordinateur
MDA	Model Driven Architecture
MES	Manufacturing Execution System
SCE	Supply Chain Execution system
SCM	Supply Chain Management system
SCP	Systèmes Contrôlés par le Produit
UML	Unified Modelling Language
XAO	Désigne tout processus Assisté par Ordinateur (AO)
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Introduction générale

I. Problèmatique

Pendant les années quatre-vingt-dix, l'intégration des systèmes d'entreprise a été l'un des challenges des entreprises, et a ainsi suscité de nombreux travaux au sein d'organisations scientifiques internationales de recherche (IFAC, IFIP) et de standardisation (IEC, ISO, CEN). L'intégration des systèmes consiste à assembler les différentes parties d'un système et à assurer leur compatibilité ainsi que le bon fonctionnement du système complet. Il s'agit donc de faire tomber les barrières fonctionnelles et organisationnelles eu sein des entreprises afin que l'ensemble soit vu comme un tout cohérent. Or, les travaux issus de ce domaine ont montré que l'adaptation organisationnelle des entreprises doit, tout d'abord, passer par une adaptation technologique des outils et applications mis en œuvre. Depuis la fin des années quatre-vingt-dix, le paradigme de l'interopérabilité dans l'entreprise a ainsi pris le pas sur l'intégration car plus focalisé sur un aspect technologique en forte évolution. L'interopérabilité des systèmes n'est qu'un moyen, parmi d'autres, pour faciliter l'intégration (Panetto 2006). L'interopérabilité peut être définie comme la capacité de communiquer avec des systèmes distants pour accéder et faire appel à leurs fonctionnalités (Vernadat 1996), l'intégration représente, quant à elle, un concept beaucoup plus large. Selon (Chen 2005), cette mutation intégration/interopérabilité n'est pas seulement un changement de techniques ; elle reflète, surtout, l'évolution des besoins économiques, organisationnels et sociaux de la société face à la globalisation du commerce et de la production manufacturière.

En effet, dès le début des années quatre-vingt-dix, le monde de l'entreprise a vu se déployer des réseaux d'entreprises ; notamment les réseaux de fournisseurs et de sous-traitants autour de leurs clients/donneurs d'ordres. Ces réseaux ont émergé suite à l'augmentation des besoins en sous-traitance pour répondre aux fluctuations d'activités. Les grandes firmes industrielles souhaitent rompre avec les logiques strictes d'intégration et de verticalisation des relations d'échange qui semblent moins adaptées aujourd'hui aux brusques variations des conditions concurrentielles (Alcouffe 2001). Ces changements s'appuient sur la constitution, autour de l'entreprise, d'un réseau stable de partenaires aux activités complémentaires et flexibles dans

lequel des relations durables sont contractualisées. Ce réseau forme ce que l'on peut appeler l'entreprise étendue (*cf*. Figure 1).

Aujourd'hui, une pratique courante tend à assimiler l'entreprise étendue aux technologies qui la caractérisent. Les capacités d'interopérabilité et d'interaction offertes par les Technologies de l'Information (TI) actuelles sont alors retenues comme principe moteur des processus de construction des entreprises étendues (Benchimol 1993) sinon comme éléments stratégiques les conduisant (Bouessel du Bourg 1992). Dans ce cadre, il est donc clair que la solidité de l'entreprise étendue et sa survie sont fortement liées à la qualité, et la viabilité des techniques d'interopérabilité établies entre les différents systèmes impliqués dans le réseau d'entreprises.

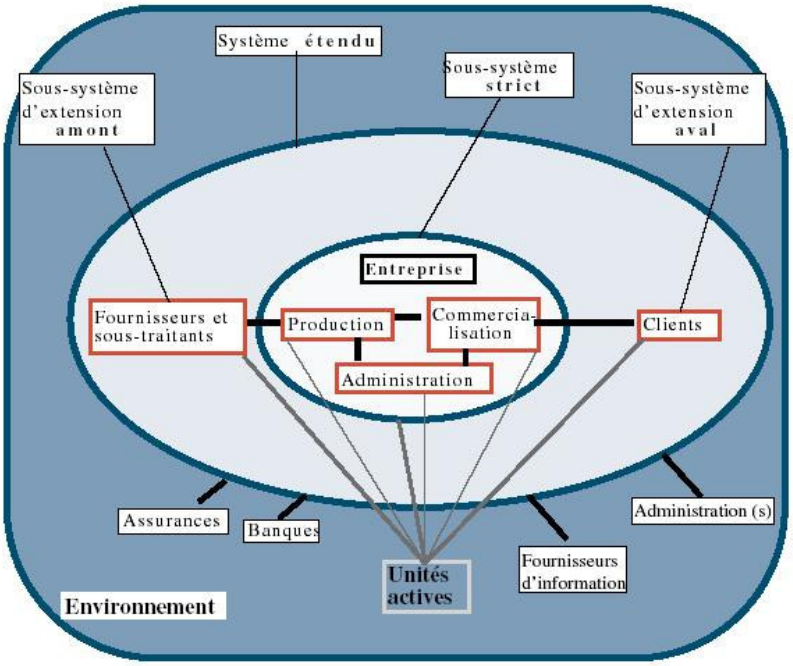


Figure 1. Environnement de l'entreprise étendue (Alban 1996).

Outre le besoin pour une interopérabilité inter entreprises, l'évolution incessante en nombre et en genre des applications impliquées dans la réalisation d'un objectif économique et le besoin de coordination et de coopération entre les différents systèmes font de l'interopérabilité entre systèmes évoluant au sein d'une seule et même entreprise (intra entreprise) une autre facette à pas négliger.

En général, l'interopérabilité qu'elle soit inter ou intra entreprise est mise en œuvre grâce à des solutions d'interconnexion dites "point-à-point" sous forme de liens rigides préétablis

entre différentes applications. Ces solutions se basent sur la modification interne de chacune des applications impliquées pour qu'elle soit capable d'échanger (envoyer et recevoir) des messages, données, informations, ou modèles avec le reste des applications de l'entreprise (Linthicum 1999). Cependant, la mise à l'échelle de ces techniques reste très limitée, en effet si ces solutions résolvent efficacement l'interconnexion dans le cas de deux applications, la prise en compte de chaque application additionnelle demande de nouvelles modifications dans celles déjà en place. En effet, la résolution de l'interopérabilité dans l'entreprise en raisonnant directement sur les applications et leurs implémentations donne lieu à un processus d'interconnexion de systèmes qui peut très rapidement évoluer pour devenir complètement chaotique et inextricable (*cf.* Figure 2), il donne lieu généralement à des architectures difficilement gérables, vu le manque d'organisation et de stratégie globale dont il souffre, d'où le chaos informatisé qui en découle.

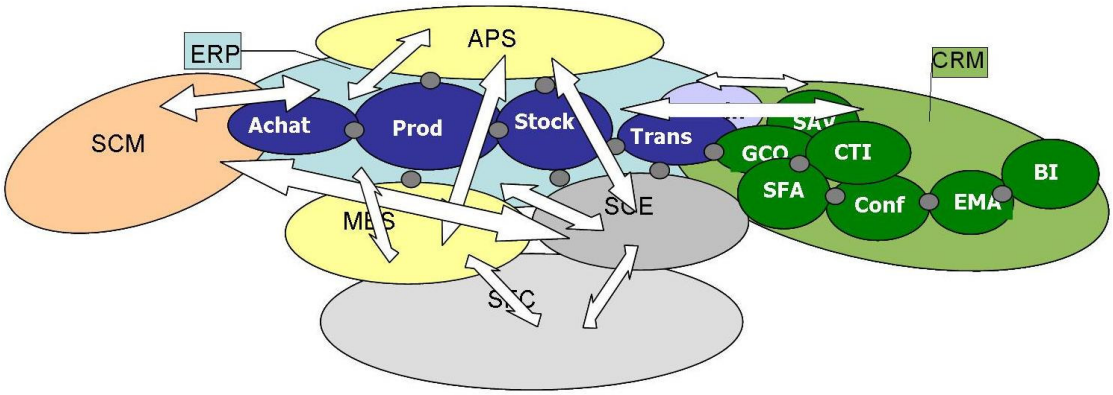


Figure 2. Intégration chaotique dans la constellation des systèmes d'entreprises.

Récemment, plusieurs projets européens ont vu le jour autour du domaine de l'interopérabilité (IDEAS 2002; UEML 2002; ATHENA 2003; INTEROP 2003) ; les travaux de cette thèse se situent dans le cadre du projet européen INTEROP NoE (INTEROP 2003).

INTEROP NoE est un réseau d'excellence européen financé par la commission européenne pour une période de trois ans, dont l'objectif est de mettre en place une activité de recherche innovatrice et compétitive dans le domaine de l'interopérabilité entre applications et systèmes d'entreprise. Pour cela, INTEROP tente de combiner trois différents domaines de recherche nourris par les compétences des différents partenaires (quarante sept en tout). Il s'agit des domaines relatifs :

- aux ontologies (ONT) ; pour l'identification et la formalisation de la sémantique autour de l'interopérabilité des entreprises
- à la modélisation d'entreprise (EM) ; pour définir et formaliser les besoins de l'interopérabilité en terme de modèles,
- aux architectures et aux technologies (A&T) permettant l'implémentation de plates-formes pour l'interopérabilité.

Dans le cadre de ce projet, notre contribution et notre démarche scientifique tirent partie des spécificités du domaine d'application concernant les entreprises de production de biens et de services afin de proposer une approche pour l'interopérabilité des systèmes et applications d'entreprise manipulant des données relatives au produit au sein de son environnement de production.

Traditionnellement, l'intégration et le partage de l'information entre applications d'entreprises s'expriment sous la forme d'une pyramide à trois niveaux (*cf.* Figure 3). Au niveau le plus bas (L1), les processus d'exécution assurent les transformations spatiales, temporelles et morphologiques des produits physiques requises par le procédé de fabrication, Au niveau L2, les processus de pilotage de la production assurent à travers le temps la fluidification des biens et des services. Au niveau le plus haut (L3), les processus de gestion traitent les différents aspects informationnels de l'entreprise.

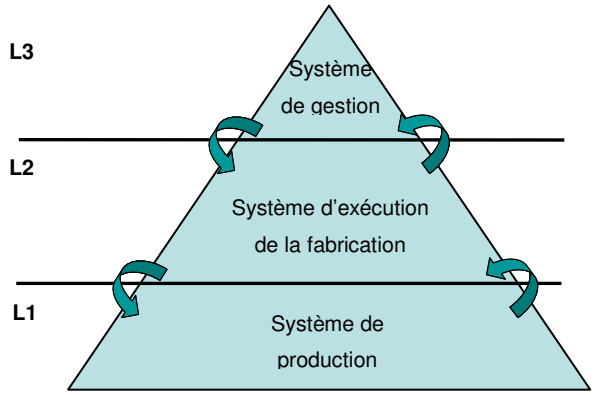


Figure 3. Structure classique des entreprises manufacturières

Cependant, les systèmes et applications appartenant à cette structure se focalisent sur des finalités différentes par rapport au produit. Même si pendant longtemps, seule la production de produits tangibles a été envisagée, la tendance actuelle est de prendre en compte de la

même manière la production de produits tangibles, la production de services et la production mixte (produits tangibles + services) (Vallespir 2003). En effet, dans le contexte des systèmes contrôlés par le produit, où le produit représente le « pivot » et la plaque tournante se trouvant au centre des objectifs de l'entreprise contemporaine, nous distinguons deux types de systèmes ; les systèmes dits « Business », dont la finalité concerne les services relatifs aux produits, et les systèmes dits « Manufacturing » dont la finalité concerne le produit en tant que bien physique tangible. Nous définissons, ainsi, deux pôles ayant deux finalités complémentaires (*cf.* Figure 4) :

- Le pôle « Business » regroupant les systèmes « Business », dont la finalité est le Produit/Service ;
- Le pôle « Manufacturing » regroupant les systèmes manufacturiers, dont la finalité est le Produit/Bien.

Chacun de ces pôles met en œuvre des applications ou progiciels complémentaires du point de vue de leurs fonctionnalités, répondant chacun à des besoins spécifiques dans l'entreprise, mais partageant, au moins partiellement, les mêmes données ou modèles d'entreprise.

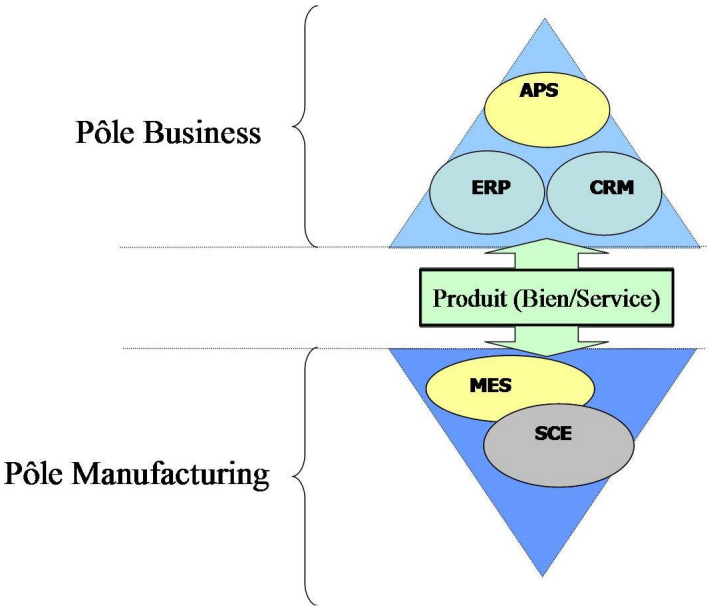


Figure 4. Séparation des pôles Manufacturing et Business (Baïna and Morel 2006)

A titre d'exemple non exhaustif, le pôle « business » peut impliquer un APS (Advanced Planning and Scheduling system), un ERP (Enterprise Resource Planning system) et/ou un

CRM (Customer Relationship Management system). Le pole « Manufacturing » fait intervenir des applications d'entreprise telles qu'un MES (Manufacturing Execution System) et/ou un SCE (Supply Chain Execution system), d'autres applications peuvent aussi faire partie de cette architecture logicielle. L'ensemble de ces applications représente le système d'information de l'entreprise et toutes cherchent à apporter une plus-value aux produits/services de l'entreprise concernée.

Alors que la plupart des travaux existant autour du thème de l'interopérabilité se concentrent sur la seule dimension informationnelle des systèmes d'entreprise au niveau « Business ». (Bouessel du Bourg 1992; Casati et al. 1996; Wegner 1996; Vallecillo et al. 2000; Kalfoglou and Schorlemmer 2004; Molina et al. 2004; Morris et al. 2004), nos travaux visent à prendre en considération la dimension matérielle, reliée à la représentation des caractéristiques physiques du produit et contrainte par la réalité physique des objets circulant dans l'entreprise au niveau « Manufacturing ». Notre objectif est ainsi de proposer une approche méthodologique pour étudier l'interopérabilité entre les systèmes du niveau « Business » et les systèmes du niveau « Manufacturing ». Nous la qualifierons d'interopérabilité B2M (Business to Manufacturing).

Au sein des environnements de production, outre les échanges informationnels (ou d'informations) entre les systèmes d'entreprise, nous considérons aussi les échanges physiques (ou de matière). La figure 5 représente les flux physiques et informationnels échangés entre les différentes étapes du cycle de vie d'un produit depuis l'état de matière première jusqu'à son usage puis son recyclage. Nos travaux sont ainsi motivés par le besoin d'une approche orientée produit pour l'échange de données prenant en compte l'image informationnelle et l'image physique des objets de l'entreprise, afin de permettre leur constante adéquation.

Les technologies d'identification du produit de manière unique (AutoID (McFarlane 2002), UPnP[3], RFID), ainsi que l' « électronisation » du produit (apport d'information électronique sur le produit) impliquent - volontairement ou non – l'unification et la cohérence des différents flux informationnels et physiques relatifs au produit (McFarlane et al. 2002).

[3] *http://www.upnp.org/*

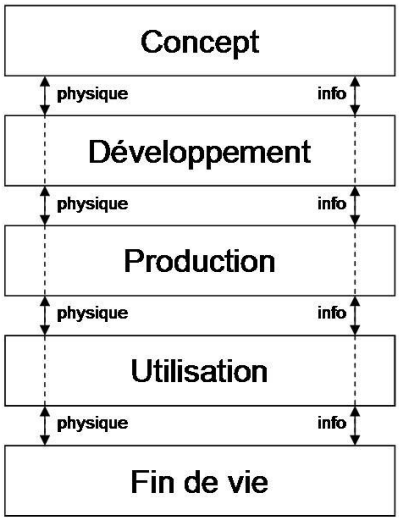


Figure 5. Flux de matière et d'information

La figure 5 illustre les liens forts existant entre traitements informationnels et traitements physiques ayant lieu au sein d'une entreprise dans l'objectif de réaliser la finalité liée a la production des produits. Notons que dans l'enchaînement des étapes relatives à la production d'un produit (objet tangible+service) représenté dans la figure 5, si les deux étapes les plus hautes sont plutôt relatives aux services de conception et de développement des produits références, les étapes suivantes sont plutôt relatives au produit physique en tant qu'entité tangible.

Cependant, chacun des systèmes de l'entreprise possède une représentation différente du produit selon l'utilisation qu'il en fait ; l'unification de l'information rattachée au produit ne pourra donc être réalisée que si l'on dispose d'un modèle générique susceptible d'être utilisé à tout moment du cycle de vie du produit. De plus, les différentes images informationnelles d'un produit peuvent devenir incohérentes, d'une part entre elles-mêmes, et d'autre part avec la réalité physique du produit. En effet, l'interopérabilité autour des produits physiques et leurs représentations virtuelles dans l'entreprise manufacturière entre les différents systèmes du processus de production nécessite une démarche généralisée pour l'établissement de solutions ordonnées et structurées.

L'interopérabilité des échanges relatifs au produit entre systèmes d'entreprise peut s'inscrire, non seulement, dans une politique d'interopérabilité intra-organisationnelle visant à spécifier et formaliser l'ensemble des échanges ayant lieu au sein d'une même entreprise, mais aussi

dans le cadre d'une politique inter-organisationnelle dans laquelle les échanges se font entre applications appartenant à différentes entreprises permettant, ainsi, l'ouverture et la transparence dans la traçabilité des produits de l'entreprise agile contemporaine.

Dans ce contexte, la problématique principale de notre thèse est de proposer une approche pour l'interopérabilité des systèmes d'entreprise en relation avec les produits, composante commune respectivement au domaine de la production et au domaine de la gestion d'entreprise. En effet, dans le contexte de la production, l'importance du produit en tant qu'entité identifiable, interagissant avec son environnement, jouant un rôle actif dans la prise de décision, est indéniable. Selon leurs besoins et leurs interactions avec le produit, la plupart des applications de l'entreprise possèdent une vue ou une représentation même très réduite du produit. Le développement de nouveaux produits passe par plusieurs phases et nécessite l'intervention de plusieurs métiers. La perception du produit dans chaque phase de son cycle de vie est différente. Les informations sur les produits sont perçues sous différents angles selon les applications, les systèmes impliqués et les outils informatiques utilisés lors de chaque phase, par exemple : des vues fonctionnelles, des vues comportementales et des vues informationnelles. Chaque vue est perçue à travers un certain nombre de représentations. Chacune de ces représentations est créée en utilisant des outils spécifiques pour des besoins spécifiques (Boitard 1998; El Hadj Mimoune 2004).

Nous nous intéressons spécifiquement aux représentations des informations collectées sur le produit depuis sa conception et tout au long de son cycle de production. Ces représentations informationnelles peuvent diverger de sa réalité physique. En effet, dans un environnement de production, parallèlement au produit physique fabriqué dans les ateliers, l'image informationnelle du produit est aussi modifiée et mise à jour par l'ensemble des systèmes impliqués dans le processus de production. Cette image est constituée d'informations relatives à l'état d'avancement de la fabrication du produit, et d'informations utiles pour le contrôle et la production du produit final pour des besoins de traçabilité (Terzi et al. 2004 ; Terzi 2005) ou de routage (Gouyon 2004). Il est donc nécessaire d'assurer tout au long du cycle de vie du produit la cohérence entre ses différentes représentations et sa réalité physique, et ce en permettant la coordination entre les différents systèmes manipulant ces représentations.

La problématique peut être décomposée en deux sous problèmes que nous résumons dans les deux questions suivantes :

- Comment faire en sorte que l'ensemble des représentations soient cohérentes avec la réalité physique du produit ?
- Comment faire en sorte que les différentes représentations détenues pas les différents systèmes soient cohérentes entre elles?

En résumé, l'objectif de la thèse est de proposer une approche pour une interopérabilité entre les différentes applications en relation avec le monde du produit en prenant en compte la dimension physique du produit, c'est ce que nous appellerons par la suite « l'interopérabilité orientée produit ». Cette interopérabilité vise à maintenir la cohérence entre les différentes représentations du produit, manipulées par les différents composants du système d'information de l'entreprise. Cette approche doit aussi garantir la cohérence de ces représentations avec l'état réel du produit physique tout au long de son cycle de vie (conception, fabrication, commercialisation, utilisation…). Dans notre proposition, nous nous intéresserons plus précisément aux approches d'interopérabilité relative aux modèles des applications et non la résolution de l'interopérabilité par l'interconnexion des implémentations des applications elles mêmes. En effet, nous tentons dans cette thèse de proposer une approche prenant en compte l'interopérabilité orientée produit lors de modélisation des systèmes.

II. Organisation du manuscrit

Le manuscrit de la thèse est structuré en quatre chapitres et une conclusion générale, dont nous donnons une vision synthétique dans la figure 6:

Le premier chapitre est consacré à une étude bibliographique de l'interopérabilité des applications et des systèmes d'entreprise dans le cadre général. Nous montrons la diversité des définitions établies pour l'interopérabilité selon les différents domaines et les différents systèmes impliqués. Par la suite, nous rapportons quelques une des catégorisations et classifications des différents types et classes d'interopérabilité entre systèmes, et nous présentons aussi l'approche dirigée par les modèles (Mellor et al. 2004) mise en œuvre pour l'établissement de l'interopérabilité entre applications.

Par analogie au problème d'interopérabilité dans le cadre général, nous orientons nos travaux, vers l'étude de l'interopérabilité orientée produit dans un contexte de production, à savoir, assurer la cohérence des différentes vues des produits de l'entreprise et la correspondance entre les vues informationnelles des ces mêmes produits et leur réalité physique. Dans le chapitre 2, nous présentons le « meta-modèle holonique », notre proposition pour la description des produits dans l'entreprise. Ce meta-modèle formalise une représentation unique et générique du produit, pouvant être exploitée par l'ensemble des applications manipulant des informations relatives au produit. Le modèle est construit sur les bases de l'ontologie de Bunge-Wand-Weber (Wand and Weber 1993; Wand and Weber 1995), décrivant les objets du monde réel d'une manière générique et abstraite.

Le chapitre 3 illustre la mise en application de l'approche de modélisation holonique basée sur le meta-modèle présenté dans le chapitre 2. Nous présentons, ainsi, une implémentation du meta-modèle holonique au sein d'un environnement professionnel pour la modélisation d'entreprise. Par la suite nous validons notre approche de modélisation orientée produit par un cas d'application réalisé dans le cadre d'une collaboration dans un projet industriel dont le but est de mettre en place un système de gestion de traçabilité dans des moulins de farine. L'objectif de cette application industrielle est de montrer l'applicabilité de l'approche de

modélisation proposée dans un cadre industriel réel. Cette application s'appuie sur notre approche de modélisation pour établir une représentation des produits de l'entreprise concernée qui corresponde à la réalité des traitements physiques et informationnels auxquels ils sont sujets.

Dans le chapitre 4 nous formalisons les mécanismes de mappings de meta-modèles et de modèles, issus des propositions pour l'interopérabilité dirigée par les modèles Cette formalisation se base sur la théorie des ensembles pour spécifier les mappings nécessaires pour une interopérabilité sémantique dans le contexte d'une architecture dirigée par les modèles. Nous montrons, en suite, l'application de ces mappings dans notre cas particulier pour mettre en place une approche pour l'interopérabilité orientée produit sur la base de notre meta-modèle de référence. Nous illustrons nos propos en réalisant deux exemples de mappings exprimant la correspondance sémantique entre le meta-modèle holonique que nous proposons et le meta-modèle de UEML, relatif au niveau organisationnel de l'entreprise, dans le premier exemple puis le meta-modèle du standard IEC 62264, plus proche du monde opérationnel de l'entreprise, dans le second exemple. Nous montrons aussi comment les mappings identifiés peuvent être intégrés dans l'environnement MEGA. Par la suite, nous montrons comment l'approche de modélisation et les mappings peuvent être utilisés dans le contexte d'une application au sein de l'environnement pédagogique AIPL.

Tout au long du manuscrit, nous nous basons sur le cadre de modélisation *ZACHMAN* pour construire un guide permettant au modélisateur d'avancer étape par étape pour couvrir l'ensemble des aspects définis par notre approche.

Finalement, le dernier chapitre conclut ce mémoire par un bilan général des travaux de cette thèse ainsi que quelques pistes intéressantes en guise de perspectives de recherche pour une suite des travaux.

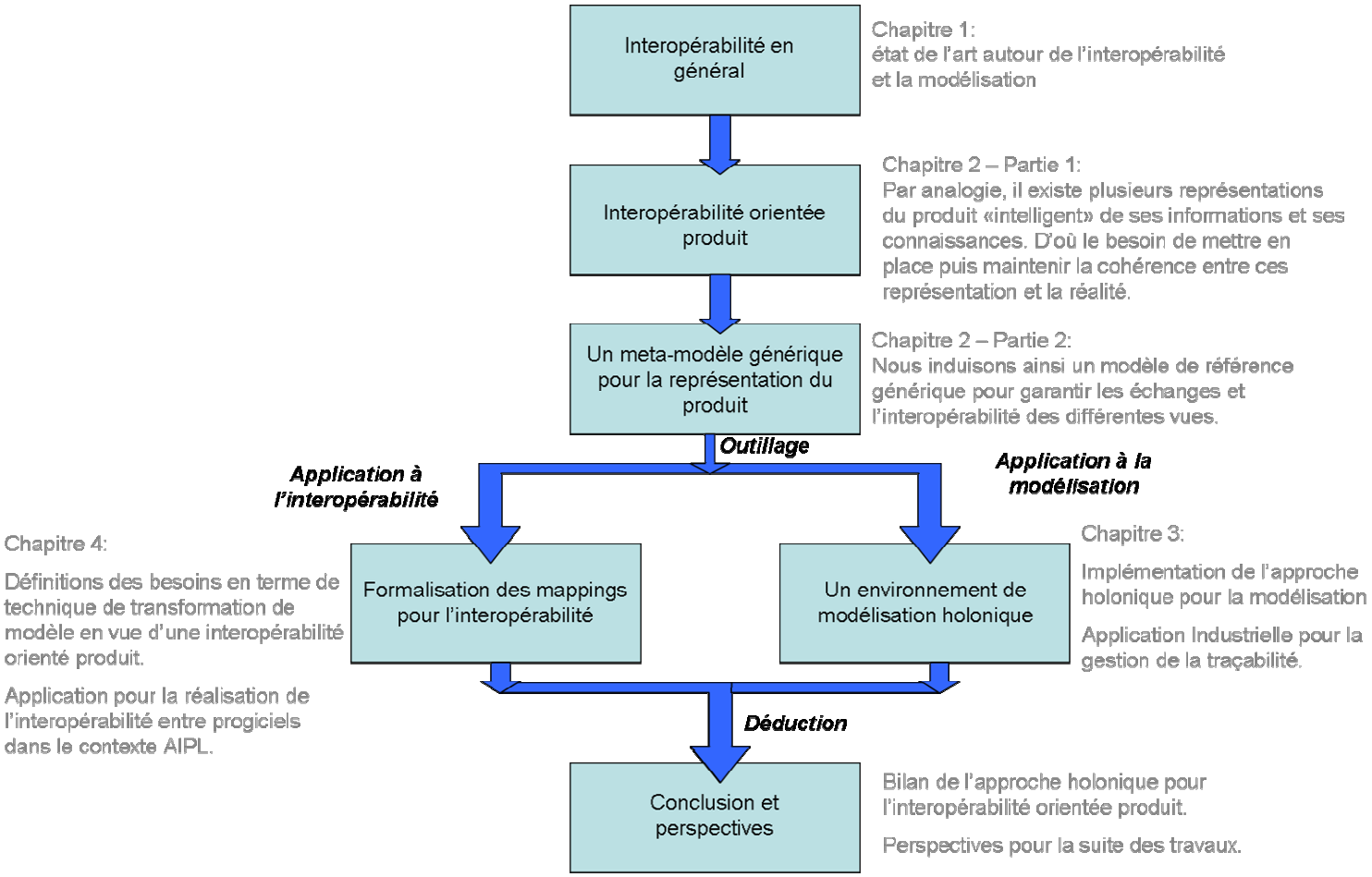


Figure 6. Vue synthétique sur les chapitres et leurs interconnexions.

Chapitre 1 :
Interopérabilité des systèmes d'entreprise

I. Introduction

Nous avons vu dans l'introduction que la compétitivité dans le contexte des réseaux d'entreprises et des entreprises étendues est étroitement liée à la capacité à structurer, partager et échanger les connaissances et le savoir-faire avec l'ensemble des entreprises participant au réseau. Ce partage des connaissances se traduit souvent par l'interconnexion des systèmes d'informations et applications des différentes entreprises en vue de les rendre interopérables, la structuration du savoir-faire quant à elle se nourrit des avancées scientifiques dans le domaine de la modélisation de l'entreprise. Dans ce chapitre, nous présentons un état de l'art couvrant le domaine de l'interopérabilité des systèmes d'entreprises, tout en montrant le lien avec la modélisation dans l'entreprise.

Dans la partie concernant l'interopérabilité, nous introduisons quelques définitions du paradigme « Interopérabilité » ainsi que les différentes catégories de l'interopérabilité. Nous illustrons cette partie par quelques exemples concrets concernant la réalisation de solutions d'interopérabilité dans l'entreprise.

Dans la partie relative à la modélisation dans l'entreprise, nous définissons tout d'abord quelques un des concepts clés dans le domaine de la modélisation tels que modèles et meta-modèles, puis nous introduisons un bref historique de la modélisation dans l'entreprise depuis ses débuts jusqu'à l'apparition des premiers cadres de modélisation et standards pour la modélisation d'entreprise.

Ce chapitre se terminera par une présentation de quelques approches utilisant la modélisation comme moyen fondamental pour l'établissement de l'interopérabilité entre systèmes d'entreprise. Dans cette section nous introduisons également quelques travaux autour de l'utilisation des ontologies comme moyen fédérateur pour l'interopérabilité des entreprises. Nous présentons ainsi quelques unes des ontologies d'entreprise les plus répandues.

II. L'interopérabilité dans l'entreprise

Dans un contexte où deux acteurs (humains ou machines) communiquent en s'échangeant des expressions, phrases, données, ou autres, on peut identifier plusieurs types de communication et ce, selon la compréhension mutuelle des éléments échangés (*cf.* Figure 7) (Euzenat 2001).

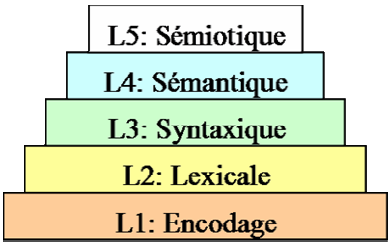


Figure 7. Les différents niveaux de communication selon (Euzenat 2001).

- **Niveau Encodage** : l'aptitude à identifier la représentation des caractères;
- **Niveau Lexical** : la capacité de reconnaître les mots ou les symboles;
- **Niveau Syntaxique** : la capacité de structurer les échanges en phrases, formules ou assertions.
- **Niveau Sémantique** : la capacité de reconstituer le sens propositionnel des représentations échangées.
- **Niveau Sémiotique** : la faculté de restituer le sens pragmatique et le contexte des phrases échangées.

Cette représentation en couches exprime implicitement le fait que chacun de ces niveaux nécessite la satisfaction des niveaux plus bas avant d'être satisfait à son tour. Les trois premiers niveaux (encodage, lexical et syntaxique) sont généralement atteints par le biais de l'utilisation d'un langage commun entre les différents systèmes communicants. La communication sémantique permet, quant à elle, une compréhension et une interprétation unique des données, informations et services échangés entre systèmes communicants (émetteur et récepteur). Ainsi, on appelle « interopérabilité sémantique » la faculté des systèmes communicants d'interpréter les annotations échangées lors de la communication, par

exemple, assigner à chacun des éléments échangés le modèle correspondant ou une interprétation adéquate. Ce paradigme est appelé plus communément « Interopérabilité ». L'interopérabilité sémantique permet ainsi à des systèmes de combiner l'information reçue de la part d'autres sources d'information et de la traiter tout en en conservant le sens.

II.1. Définitions

Il existe de nombreuses définitions de la notion d'interopérabilité. Une recherche pertinente sur le web peut produire jusqu'à 22 définitions différentes, variant selon le domaine et les systèmes impliqués. Voici, de manière non exhaustive, quelques définitions tirées de la littérature :

- la capacité de communiquer, exécuter des programmes ou transférer des données entre différentes unités fonctionnelles de manière transparente par rapport à l'utilisateur ; en ne nécessitant pas ou très peu d'effort de la part de l'utilisateur. (The ISO/IEC 2382 Information Technology Vocabulary)
- la capacité de deux ou plusieurs systèmes ou composants d'échanger puis réutiliser de l'information (IEEE 1990).
- la capacité de communiquer avec des systèmes distants pour accéder et faire appel a leurs fonctionnalités (Vernadat 1996).
- l'aptitude de deux systèmes ou plus, à communiquer, coopérer et échanger des données et services ; et ce malgré les différences dans les langages, les implémentations et les environnements d'exécution ou les modèles d'abstraction. (Wegner 1996)
- accomplie seulement si l'interaction entre les systèmes impliqués couvre les aspects *données*, *ressources* et *business process* en utilisant les sémantiques définies dans le contexte business (Chen and Doumeingts 2003)
- au sens informatique, l'aptitude à échanger et utiliser de l'information. (WordNet)

Toutes ces définitions illustrent bien l'intérêt et l'engouement qui existent autour du paradigme de l'interopérabilité. Cependant, l'interopérabilité des applications n'est pas seulement un problème technologique ou conceptuel, mais elle peut aussi être due à un

problème organisationnel (Panetto 2006a). En effet, selon l'EIF[4] (EIF 2004), il existe plusieurs types d'interopérabilité:

- L'interopérabilité technologique relative à la mise en œuvre des technologies de l'information et de la communication concernant les normes pour présenter, stocker, échanger, traiter et communiquer les données au moyen de matériels informatique ;
- L'interopérabilité sémantique, de niveau plus conceptuel, qui doit assurer que les informations échangées sont compréhensibles du point de vue de leur signification et de leur interprétation par les applications qui les utilisent mais ayant été développées pour des objectifs différents. L'interopérabilité sémantique permet à des systèmes de combiner l'information reçue de la part d'autres sources d'information et de la traiter tout en en conservant le sens ;
- Et enfin, l'interopérabilité organisationnelle définissant les responsabilités, autorisations, confiances, aspects légaux, propriétés intellectuelles et structures organisationnelles nécessaires à l'acceptation des échanges d'information entre applications par les différents acteurs. Ce niveau d'interopérabilité est particulièrement mis en avant dans le cadre de l'interopérabilité de l'administration électronique et des gouvernements (Klischewski 2004).

La diversité des définitions, des utilisations et des typologies établies dans le domaine de l'interopérabilité se traduit dans la multitude de solution et de technologies mises en œuvre pour répondre à ce besoin grandissant d'interopérabilité. En effet, les technologies et outils mis en œuvre pour implémenter des solutions pour l'interopérabilité dépendront du type et du degré d'interopérabilité visés (Visser et al. 2000).

II.2. Solutions pragmatiques pour l'interopérabilité en entreprises

Dans cette section nous présentons quelques exemples de solutions pour l'interopérabilité des systèmes d'entreprise. Certaines de ces solutions sont dites ad hoc, d'autres, plus structurées,

[4] *EIF : European Interoperability Framework*

sont réalisées grâce à l'établissement de modèles d'échange, d'interfaces, de langages ou de syntaxes communs pour formuler les échanges entre les applications.

II.2.1. Interopérabilité ERP/MES

L'objectif principal de l'interopérabilité entre les plates-formes ERP (Enterprise Resource Planning) et les MES (Manufacturing Execution System) est d'améliorer la synchronisation de l'aspect administratif de la gestion de l'entreprise et la réalité au niveau de l'atelier (*cf.* Figure 4). L'interfaçage ERP - MES est censé accélérer la circulation des flux entre les plannings réalisés par l'ERP et le coté contrôle du processus réalisé au niveau du MES. La conception de cette interface nécessite l'analyse des informations manipulées par chacun des deux systèmes afin d'identifier l'information adéquate concernant l'ERP et l'information destinée au MES (*cf.* Figure 8).

Le standard IEC 62264 (IEC 62264 2002), travail commun à l'ISO et au CEN, est une proposition pour normaliser ces échanges informationnels mais, à ce jour, seules quelques implémentation très partielles ont été réalisées (Siemens SIMATIC IT, Ordinal GlobalSCREEN Intra). Cependant, il existe plusieurs tentatives propriétaires visant à interconnecter des systèmes ERP et MES, dans le but de réaliser une solution d'interopérabilité entre les deux niveaux.

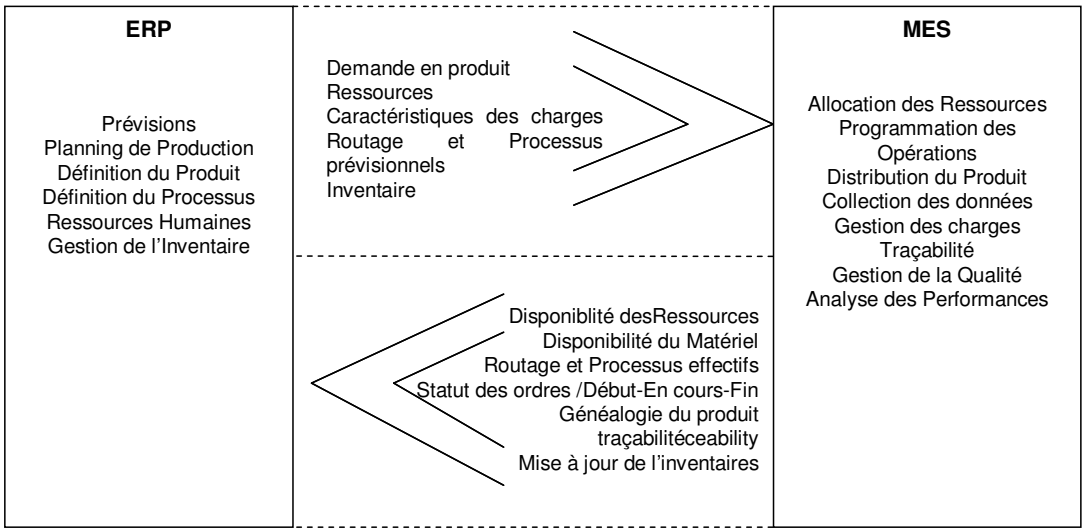


Figure 8. Exemple de données échangées entre le niveau ERP et le niveau MES.

LinkForSap (Yutaka 1998) est un exemple pour l'unification des données entre l'ERP "SAP[5] R/3" et le système de contrôle de la production "Centum" (Centum). Même si les solutions similaires à "LinkForSap" résolvent le problème de l'interconnexion dans un contexte CIM (Computer Integrated Manufacturing), ces solutions sont spécifiques à un ERP donné, et à un MES donné. En effet, ce type de solution est un exemple de connexion dite « point à point » dont la maintenabilité peut être très coûteuse ; il arrive même parfois que la maintenance des solutions d'interconnexion devienne plus coûteuse que les applications interconnectées elles-mêmes.

II.2.2. Interopérabilité SCM/CRM

Outre les solutions d'interopérabilité intra-organisation telles que l'interconnexion ERP-MES, le développement des systèmes d'information d'entreprise, ainsi que le besoin de collaboration entre différentes entreprises ont ouvert le chemin vers un nouveau type d'interopérabilité ; l'interopérabilité inter- organisationnelle ; autrement dit l'interopérabilité entre applications mises en œuvre dans différentes entreprises. Généralement, l'interopérabilité « inter-entreprises » a pour objectif de structurer, améliorer ou consolider les échanges d'informations entre les fournisseurs et leurs clients et ce pour assurer la bonne circulation des flux d'information entre entreprise, nécessaires pour le contrôle de la qualité de service relative aux flux physiques échangés. A titre d'exemple, nous citons ici les travaux présentés dans (Selk et al. 2005). Cette architecture propose une solution pour l'interconnexion de systèmes de gestion des relations avec les clients (CRM : Customer Relationship Management system) et de systèmes de gestion de la chaîne logistique (SCM : Supply Chain Management system), en vue de la construction un système d'information étendu pour couvrir l'ensemble de l'entreprise « étendue ». Cette proposition est basée sur une interconnexion fonctionnelle d'une application CRM et une application SCM. Cependant dans cette architecture, une supposition forte veut que clients et fournisseurs possèdent la même architecture interne pour leurs systèmes d'informations, ce qui constitue une hypothèse très peu réaliste. En effet, cette hypothèse est tout à fait contraire aux propriétés de flexibilité,

[5] *SAP, http://www.sap.com/france*

ouverture et réactivité constituant la base même des réseaux d'entreprises et entreprises étendues (Alban 1996).

II.2.3. Interopérabilité des systèmes de gestion de workflow

Dans les différents niveaux de la hiérarchie des applications d'entreprise, outre l'interopérabilité des applications ERP et MES, il existe d'autres solutions pour l'interopérabilité impliquant d'autres types d'application et de systèmes d'entreprise. Les systèmes de gestion de workflow (WfMS : Workflow Management Systems) font partie de la constellation de systèmes que l'on peut rencontrer. Dans ce paragraphe, nous présentons quelques unes des tentatives visant à rendre interopérables les systèmes de gestions de workflows.

La WfMC (Wokflow Management Coalition) a définit une interface pour la communication et l'interopération entre moteurs de gestion de workflow. L'objectif principal de cette tentative est de définir des standards pour établir la communication entre WfMS en dépit des différentes implémentations propres à chaque fournisseur (WFMC 1995a). L'interface définie dans (WFMC 1995b) contient la spécification de tous les appels de fonction possibles entre deux WfMS en couvrant l'ensemble des possibilités de coopération (*e.i.* échange d'informations, synchronisation des exécutions, envoi de requêtes et retour de réponses). Wf-XML (WFMC 2000) est une implémentation sous format XML combinant les commandes abstraites définies par l'interface d'interopérabilité WfMC, et la technologie basée sur l'échange de message XML via un protocole http. Wf-XML définit l'ensemble des messages requête/réponse échangés entre une application dite observatrice, qui peut être ou non un moteur de workflow, et un moteur de workflow (WfMS) distant contrôlant l'exécution d'une instance de workflow. Le protocole Wf-XML a été inclus dans quelques uns des WfMS pour en assurer l'interopérabilité (ex : AFRICA (Zur Muehlen and Klein 2000))

La majorité des travaux dans le domaine de l'interopérabilité entre WfMS, se basent en général sur les protocoles d'échange de messages (Wf-XML, BizTalk[6], FIPA ACL (FIPA 2002), etc.). Ces solutions sont particulièrement utiles pour résoudre les problèmes

[6] *www.microsoft.com/**biztalk/***

d'interopérabilité syntaxique (format et vocabulaire des messages échangés, unification des types de données, etc.).

II.3. Classification et types d'interopérabilité

Pour répondre au besoin de mesurer l'interopérabilité potentielle ou celle mise en œuvre entre les système, plusieurs travaux ont été réalisés dans le but d'étudier et classifier les différentes approches et méthodes utilisées pour l'interopérabilité, afin d'en mesurer l'impact sur la qualité de la communication et des échanges mis en place. Une étude plus approfondie peut être trouvée dans (Panetto 2006b). Dans ce qui suit nous présentons quelques exemples de classifications d'interopérabilité issus de la littérature.

II.3.1. La classification IEC 62390

Dans (Kosanke 2005), l'auteur propose une adaptation de la classification IEC 62390 (IEC 62390 2005) aux systèmes d'information d'entreprise. Cette classification est initialement issue d'une étude concernant la compatibilité des composants électroniques mis en place au sein d'un système global. Selon l'IEC 62390 (IEC 62390 2005), l'interopérabilité d'un système n'est autre qu'un niveau de compatibilité qu'il est possible d'obtenir en garantissant certaines caractéristiques régissant la qualité de la communication entres les composants de ce système.

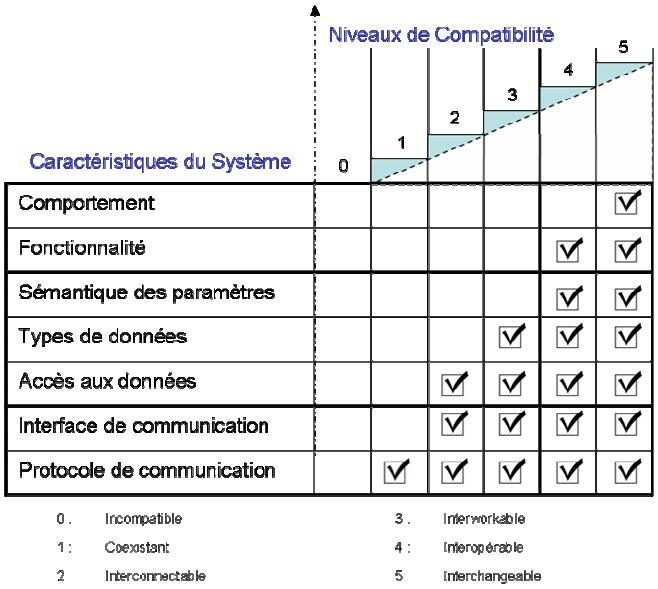


Figure 9. Niveaux de Compatibilité selon IEC 62390 (IEC 62390 2005)

L'IEC 62390 propose donc une classification des niveaux de compatibilité entre composants basée sur les caractéristiques du système de communication (*cf.* Figure 9), ce degré de compatibilité dépend du degré de coopération établi entre les composants. l'IEC 62390 définit ainsi cinq niveaux de compatibilité:

0- Incompatibilité : incapacité de deux systèmes (ou composants) ou plus à cohabiter dans le même environnement sans interférer. Cette incompatibilité peut être due à des différences au niveau des fonctionnalités, de la sémantique ou des types des données, des interfaces de communication, ou même des protocoles de communications. Des systèmes incompatibles peuvent interférer et dégrader le fonctionnement des autres systèmes faisant partie de leur environnement.

1- Coexistence : aptitude de deux systèmes ou plus provenant de différents fournisseurs à fonctionner indépendamment ou dans le cadre d'un protocole de communication dans le même environnement de communication sans interférer dans le fonctionnement des autres systèmes.

2- Interconnectivité : aptitude de deux systèmes ou plus provenant de différents fournisseurs à communiquer en utilisant les mêmes protocoles de communication et les mêmes interface. Les systèmes interconnectables s'échangent des données sans a priori sur le type des données échangées, une conversion de types peut être nécessaire.

3- Interworkability: aptitude de deux systèmes ou plus à permettre l'échange structuré de paramètres d'entrées/sorties, cet échange est basé sur une même interface de communication ainsi que les mêmes types de données pour les deux parties de la communication.

4- Interopérabilité : aptitude de deux systèmes ou plus à fonctionner simultanément au sein d'une même application. La sémantique des données utilisées, les fonctionnalités supportées sont suffisamment documentées, de telle façon à ce que le remplacement d'un système par un autre système similaire n'altère en rien le bon déroulement des fonctionnalités du système global. Cependant, le comportement dynamique des systèmes peut être différent.

5- Interchangeabilité : aptitude de deux systèmes ou plus à se remplacer mutuellement pour une certaine tâche dans une ou plusieurs applications. Après

remplacement, le nouveau système conserve les mêmes fonctionnalités que le système remplacé, le comportement dynamique des deux systèmes reste le même.

Pour la classification de l'interopérabilité dans les domaines des applications et des systèmes d'informations, il existe deux approches majeures ; la première est une approche basée sur l'étude de l'interconnexion technologique des applications et systèmes interopérant (LISI 1998; NATO C3 2003), la deuxième se focalise sur l'aspect conceptuel de l'interopérabilité à travers la définition des données et des interfaces d'échange entre les systèmes communiquant (Tolk and Muguira 2003). Nous illustrons chacune de ces deux approches par un exemple de la littérature. L'interopérabilité technique est illustrée par le modèle LISI[7] développé pas le département de la défense aux Etats Unis d'Amérique (DoD) à la fin des années 90 (LISI 1998). La classification conceptuelle de l'interopérabilité est, quant à elle, représentée par le modèle LCIM[8] (Tolk and Muguira 2003).

II.3.2. Le modèle LISI

Le modèle LISI (LISI 1998) est probablement le modèle d'interopérabilité le plus répandu dans le domaine de l'interopérabilité technologique des systèmes d'information, et même si son développement date de 1998, il reste le modèle de référence dans le domaine. Le modèle LISI aide à développer un profil d'interopérabilité entre systèmes indépendants, la comparaison de ces profils permet d'étudier le degré d'interopérabilité potentielle entre les systèmes. Le modèle LISI définit un modèle de référence composé de cinq niveaux de maturité pour l'interopérabilité (*cf.* Figure 10), ainsi qu'un ensemble de caractéristiques relatives à ces cinq niveaux.

0- Systèmes isolés : aucune connexion n'existe entre les applications. Les échanges de données sont effectués manuellement ;

1- Systèmes connectés : les applications sont connectées électroniquement (sur un même réseau), mais leurs fonctions sont complètement séparés, chacune possède son propre environnement et ses propres données. Les échanges de données semi-structurées homogènes sont possibles ;

---

[7] *Levels of Information Systems Interopèrability.*
[8] *Levels of Conceptual Interoperability Model*

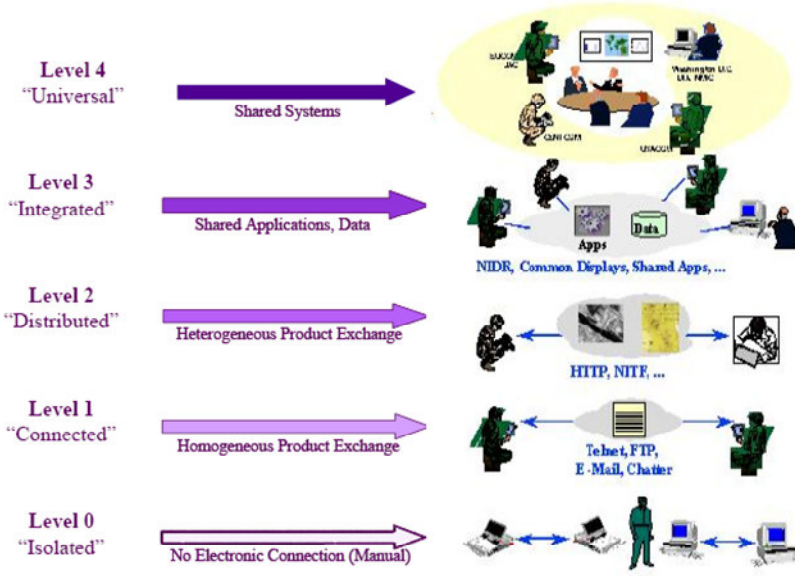


Figure 10. Les cinq niveaux d'interopérabilité du modèle LISI (LISI 1998).

2- **Systèmes distribués :** les applications ont quelques fonctions communes, mais sont totalement indépendantes, leurs données aussi sont séparées. Les échanges de données hétérogènes sont possibles ;
3- **Systèmes intégrés :** les applications sont séparées mais partagent des données communes. Elles peuvent collaborer de manière sophistiquée.
4- **Systèmes universels :** les applications et leurs données sont partagées au sein d'une même entreprise ou entre plusieurs entreprises. Les mécanismes de collaboration entre application sont très développés.

Les cinq niveaux d'interopérabilité du LISI sont basés sur quatre caractéristiques interdépendantes pour la catégorisation de l'interopérabilité. Ces quatre aspects sont connus sous l'anagramme PAID :

- **P** pour procédures, reflète les procédures, approches, règles, standards et méthodes mis en œuvre pour établir l'échange d'informations entre systèmes.
- **A** pour applications, décrit les applications permettant l'échange, le traitement et la manipulation des données partagées entre différents systèmes.
- **I** pour infrastructures, décrit l'environnement (matériel, réseaux etc.) qui entre en jeu pour assurer l'interaction des systèmes.

- **D** pour données, décrit les formats et protocoles qui permettent l'échange de données, ainsi que l'aspect sémantique permettant l'échange d'information.

La matrice résultant du croisement des cinq niveaux d'interopérabilité et les quatre caractéristiques PAID donne lieu au modèle de référence LISI (*cf.* Figure 11). Le modèle LISI offre les bases pour la comparaison et la discussion de l'interopérabilité.

Description	*Environnement d'exécution*	*Niv.*	P	A	I	D
Entreprise	Universel	4	Niveau Entreprise	Applications Interactives	Topologies Multi Dimensions	Modèle d'Entreprise
Domaine	Intégré	3	Niveau Domaine	GroupWare	World Wide Netwrok	Modèle du Domaine
Fonctionnel	Distribué	2	Niveau Application	Desktop Automation	Réseaux Locaux	Modèle de Program
Connecté	Peer to Peer	1	Niveau Local	Drivers Standards	Connections Simples	Local
Isolé	Manuel	0	Contrôle d'accés	N/A	Indépendant	Privé

Figure 11. Modèle de référence de LISI (LISI 1998)

II.3.3. Le modèle LCIM

LCIM propose une alternative plus conceptuelle aux méthodes de classification technique de l'interopérabilité en se basant sur l'étude conceptuelle de la qualité et de la documentation des interfaces relatives aux données échangées entre les systèmes interopérant, le modèle LCIM définit cinq niveaux d'interopérabilité (Tolk and Muguira 2003), dont le niveau 0 et exprime l'absence totale d'interopérabilité entre les systèmes (*cf.* Figure 12).

Niveau 0 – Données spécifiques au système : Pas d'interopérabilité entre les deux systèmes. Les données sont utilisées à l'intérieur du système d'une façon propriétaire et sans visibilité externe. Le système est dit « boite noire ».

Niveau 1 – Données documentées : les données sont documentées en utilisant un protocole commun offrant une interface d'accès. Le système est dit « boite blanche».

Niveau 2 – Données statiques alignées: les données sont documentées en utilisant un model de référence commun base sur une ontologie commune, i.e., le sens des données est décrit d'une manière non ambiguë. Ceci est possible également grâce à l'utilisation de standard de définition de meta-données. Le système est, alors, dit « boite noire avec interface standard».

Niveau 3 – Données dynamique alignées: les données utilisées à l'intérieur même du système (ou composant) sont bien définies, et ce en utilisant des méthodes standards telle que UML. Le problème à ce niveau est que même en utilisant des interfaces identiques, les différents systèmes peuvent avoir des pré requis et postulats différents.

Niveau 4 – Sémantique harmonisée : les relations sémantiques non triviales existant entre les données sont explicitées à l'aide d'un modèle conceptuel documenté. Cette documentation permet de comprendre comment les données sont traitées lors de l'exécution du système, ainsi que les connexions sémantiques non triviales entre les données.

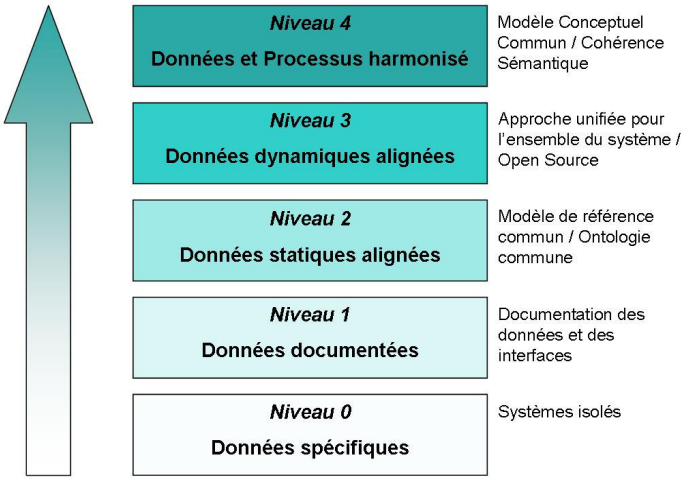


Figure 12. Classification de l'interopérabilité orientée données (Tolk and Muguira 2003)

II.3.4. Illustration

En utilisant les exemples de solutions d'interopérabilité présentés précédemment, nous illustrons l'utilisation des différentes classifications étudiées (IEC 62390, LISI et LCIM). Le tableau suivant montre le niveau d'interopérabilité auquel appartient chacun des exemples, et ce pour chacune des classification.

Cependant, comme nous pouvons le constater, le résultat de la classification peut varier selon que les critères appliqués soient techniques, ou conceptuels, ou autres. Par exemple, LinkforSap est classé comme étant un cas d'interopérabilité dans un système distribué selon la classification LISI, par contre le modèle LCIM le classe comme étant un système isolé.

Tableau 1 : Illustration des classifications sur des exemples

	Applications	IEC 62390	LISI	LCIM
LinkForSap	ERP/MES	interconnectivité (2)	Distribué (2)	isolés (0)
(Alban 1996)	CRM/SCM	interopérabilité (3)	Distribué (2)	isolés (0)
WfMC	WfMS	interopérabilité (3)	Universel (4)	données documentées (1)

II.4. Synthèse

Nous avons vu dans cette section une introduction à l'interopérabilité, à travers quelques unes des définitions les plus répandues et un ensemble d'exemples concrets issus de travaux de recherche et applications pragmatiques dans le domaine de l'interopérabilité de l'entreprise.

Par la suite, nous avons présenté quelques travaux issus de l'état de l'art, dont l'objectif est de proposer un canevas permettant de mesurer et classifier l'interopérabilité mise en œuvre selon les moyens conceptuels ou matériels utilisés pour cette interopérabilité, pour illustrer l'approche de classification basée sur l'infrastructure nous présentons le modèle LISI (LISI 1998), par ailleurs pour illustrer les classifications conceptuelles nous avons choisi le modèle LCIM (Tolk and Muguira 2003). Puis, nous avons illustrés l'utilisation des classifications sur quelques uns des exemples de solutions pour l'interopérabilité présentés auparavant.

Outres les classifications présentés dans cette partie, il existe d'autres classifications spécifiques à d'autres domaines d'application. A titre d'exemple, (SEMATECH 1995) définit trois classes d'interopérabilité (A, B, C) pour spécifier l'interopérabilité des équipements électroniques.

Dans la suite de ce mémoire, nous nous focalisons plus précisément sur l'aspect sémantique de l'interopérabilité. En effet, dans la section suivante, nous présentons un bref état de l'art du domaine de la modélisation d'entreprise. En effet, la modélisation en entreprise a souvent été annoncée comme l'un des outils pouvant aider à résoudre l'interopérabilité des systèmes d'entreprise (Vallespir et al. 2005).

Par la suite, dans la section IV l'apport des approches dirigées pas les modèles dans la conception de solutions homogènes et structurées pour l'interopérabilité sémantique des systèmes d'entreprise.

III. La modélisation dans l'entreprise

Le mouvement de mutation généralisée de notre société, d'une société industrielle vers une société de l'information (ou société du savoir) se traduit au niveau de l'entreprise par l'apparition de nouveaux besoins relatifs à la capitalisation des connaissances et le partage du savoir. Avec l'évolution des technologies de l'information et la floraison des systèmes d'information pour l'entreprise, le besoin de méthodes agréées pour la construction, la validation et la maintenance de ces systèmes n'a cessé d'augmenter. Dans ce contexte, la modélisation tente de répondre aux besoins spécifiques de l'entreprise dans le but de:

- Construire une vision et une culture communes communiquées lors de l'utilisation de modèles,
- Capitaliser la connaissance et le savoir-faire de l'entreprise pour construire une mémoire interactive de l'entreprise,
- et finalement, offrir des éléments pour l'aide à la décision pour le contrôle et l'évolution de l'entreprise.

Dans cette section, nous introduisons quelques un des concepts principaux de la modélisation, à savoir *modèle* et *meta-modèle,* et nous présentons les définitions communément admises de modèle, meta-modèle et langage de modélisation.

III.1. Modèles et meta-modèles

Tout d'abord et avant toute chose, nous commençons cette section par la définition du mot *« modèle » :*

Modèle : nom (pluriel modèles)

- Une copie d'un objet, en général réalisée dans une échelle plus petite que l'original.
- Une version simplifiée d'une chose complexe, par exemple, l'utilisation des modèles financiers pour faire de la prédiction financière. *(Dictionnaire de l'environnement)*

Les deux définitions précédentes correspondent à peu de choses prés aux définitions consensuelles de (Rothenberg 1989) et (Vernadat 1996):

> ***Définition 1.*** *Un modèle est une abstraction de la réalité dans le sens où il permet de représenter certains aspects de cette réalité dans un contexte précis. Ceci permet l'utilisation d'une vision du monde plus simplifiée, en évitant la complexité et l'irréversibilité des éléments du monde réel. La modélisation, au sens large, est donc l'utilisation d'une chose à la place d'une autre pour simplifier, faire abstraction de certaines préoccupations.* (Rothenberg 1989)

> ***Définition 2.*** *Un modèle est une représentation utile d'un sujet ; il représente une abstraction (plus ou moins formelle) de la réalité (ou de l'univers de discours) exprimé dans un formalisme (ou langage) défini par des concepts de modélisation adaptés au besoin de l'utilisateur. Ces concepts de modélisation forment les éléments d'un langage de modélisation défini par une syntaxe et une sémantique particulières.* (Vernadat 1996)

La deuxième définition s'avère la mieux appropriée pour le travail que nous exposons dans cette thèse, nous la considérons donc comme référence.

Par ailleurs, dans le domaine de la modélisation la notion de meta-modèle est elle aussi souvent évoquée. Même s'il n'existe pas de définition unique de meta-modèle (Favre 2004), nous donnons ici quelques unes des définitions les plus pertinentes :

> ***Définition 3.*** *Un meta-modèle est modèle définissant le langage utilisé pour exprimer un modèle.* (Bézivin 2004)

> ***Définition 4.*** *Un meta-modèle est le modèle d'un langage de modélisation.* (Favre 2004)

> ***Définition 5.*** *Un meta-modèle est le modèle d'un modèle.* (Seidewitz 2003)

La définition de Seidewitz (*cf.* Définition 5) paraît être la plus proche de notre compréhension du terme « meta-modèle » ainsi que de l'utilisation que nous en faisons. Dans la section suivante, nous présentons un bref survol de l'historique de la modélisation dans l'entreprise depuis ses débuts dans les années 60. Par la suite, nous présentons les différents types de modèles concernés par la modélisation d'entreprise.

III.2. Introduction à la modélisation dans l'entreprise

Dans cette section, nous ne nous intéressons pas à l'histoire de la modélisation au sens général car ceci peut nous mener à remonter à « quelques dizaines de siècles » (Favre 2004; Favre 2005). Notre objectif est de présenter un panorama succinct et précis de ce qu'est la modélisation dans le milieu de l'entreprise.

Les chercheurs de L'Ecole Scandinave furent des précurseurs de la modélisation d'entreprise au sens général, et ce dès les années 60 (Langefors en 1966). Cependant il faut attendre le début des années 80 pour voir apparaître les premières briques de la modélisation d'entreprise telle qu'on la connaît aujourd'hui (Plandata, SISU : Swedish Institute for System Development). En effet, la prise en considération des objectifs et des intentions des métiers et fonctions dans la modélisation de l'entreprise fait son apparition, et ce, en plus des autres composants de modèles tels que les entités, les associations et les processus.

Outre l'école scandinave, de nombreux travaux ont vu le jour dés le début des années 80, les plus connus proviennent des Etats-Unis ; CAMI-ICAM dans le cadre de l'initiative CIM (Computer Integrated Manufacturing), ou bien d'Europe dans le cadre de projets européens tels que AMICE (CIM-OSA), ou encore résultent de travaux de recherche tels que ceux de GRAI.

- CAMI: Association américaine à but non lucratif composée d'organisations industrielles dont le but est la création de progiciel de niveau atelier de production et de standards de modélisation.
- CIM-OSA: Modèle de référence développé par le projet européen: AMICE (Jorysz and Vernadat 1990a; Jorysz and Vernadat 1990b; Klittich 1990)
- ICAM: Projet lancé par le Materials Lab. of the US Air Force (Davis et al. 1983; Martin and Smith 1983; Martin et al. 1983).
- GRAI: Méthode de modélisation initiée dès le début des années 80 au laboratoire LAP/GRAI à Bordeaux (Doumeingts 1984), elle donnera lieu, par la suite, à la méthodologie GRAI-GIM (Doumeingts et al. 1992).

La figure 13 présente un bref panorama de l'évolution de la modélisation d'entreprise, des années 60 jusqu'à nos jours. Une étude de cette tendance et un historique de la modélisation d'entreprise ont présentés dans (Vallespir 2003).

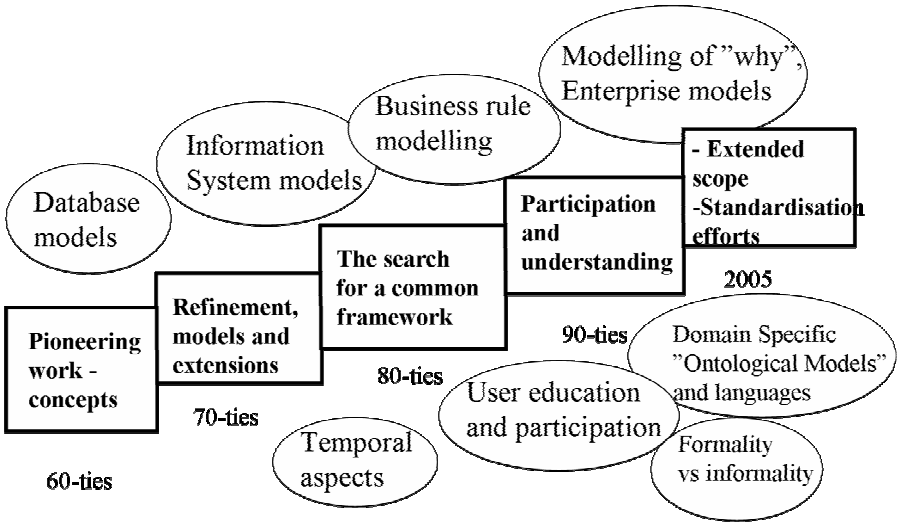


Figure 13. Évolution de la modélisation dans le temps (Janis 2005).

Généralement par le terme « entreprise », nous faisons référence à une « entité organisationnelle », cette entité peut représenter une ligne de production, un département d'une grande société ou un ensemble de processus métier (*business process*) d'une société ou bien la totalité de la compagnie (Vernadat 1996).

Définition 6. *La modélisation d'entreprise est, donc, l'ensemble des activités et des processus utilisés pour développer les différentes parties d'un modèle d'entreprise, et ce, pour répondre à certaines finalités de modélisation.*

Le plus souvent, la modélisation de l'entreprise est vue comme un moyen pour :

- Offrir une meilleure représentation et une meilleure compréhension du fonctionnement d'une entreprise,
- Capitaliser la connaissance et le savoir-faire de l'entreprise pour une utilisation ultérieure,
- Rationaliser et structurer les échanges d'information,
- Concevoir et spécifier une partie de l'entreprise (aspects structurels, organisationnels, informationnels, fonctionnels ou comportementaux),
- Analyser certains aspects d'une partie de l'entreprise (analyse économique, organisationnelle, quantitative, qualitative, etc.)
- Simuler le comportement d'une partie de l'entreprise
- Contrôler, synchroniser ou coordonner certaines parties de l'entreprise (des processus, par exemple.)

Dans le domaine des systèmes de production, les actions s'appuyant sur la modélisation d'entreprise concernent la conception des systèmes, leur évaluation ou leur restructuration (Vallespir 2003). La conception d'un système de production utilise donc la modélisation d'entreprise pour la spécification des composantes techniques, humaines et organisationnelles mises en œuvre au sein de l'entreprise ainsi que la spécification des moyens d'intégration entre ces différentes composantes. L'évaluation des systèmes de production quand à elle vise à connaître les performances de ces derniers, dès lors les techniques de structuration permettent d'améliorer ces performances.

III.3. Les différents type de modèles

Selon (Vernadat 1996), un modèle d'entreprise est le plus souvent composé de (liste non exhaustive) :

- **Modèles économiques :** définissent une vue analytique des coûts de l'entreprise, ces modèles sont utilisés pour étudier la compétitivité et l'efficacité des coûts des différentes sous-parties de l'entreprise.
- **Modèles organisationnels :** utilisés pour documenter de la structure et l'organisation de l'entreprise en termes de ligne de production, départements, cellules et centre de travail tout en représentant les responsabilités et les autorités affectées à chaque niveau de décision.
- **Modèles informationnels :** décrivent les structures et relations des éléments informationnels faisant parties des systèmes d'information de l'entreprise.
- **Modèles d'activités :** indiquent l'ensemble des opérations (ou actions) à enclencher lors de l'exécution du travail dans l'entreprise.
- **Modèles de ressource :** décrivent les caractéristiques règles de gestion et les actions supportées par les différents équipements en vue de l'exécution des activités de l'entreprise.
- **Modèles de produits :** sont utilisés pour représenter les caractéristiques géométriques ou non, les détails de conception des produits ainsi que leur composants fabriqués dans l'entreprise à travers le cycle de vie du produit.

Cependant, selon ces mêmes définitions, il est possible de définir des modèles informationnels destinées aux systèmes d'information de l'entreprise pour représenter un modèle économique, organisationnel, d'activités, de ressources ou bien de produits. Le terme informationnel lui-même est très discutable ; il laisse sous-entendre qu'il existe des modèles non informationnels, or un modèle quel qu'il soit contient forcement de l'information. En effet, dans la littérature anglaise on parle plutôt de « computational models » au lieu de « informational models » (Sowa and Zachman 1992), mais « computational » est un terme difficilement traduisible en français.

En effet, (Gustas 1995) et bien d'autres avant lui ; (Willars 1993) par exemple; considèrent les modèles informationnels (ou « computationnels ») et les modèles d'entreprise comme faisant partie de deux mondes séparés ; selon eux, les modèles informationnels ne sont qu'une représentation de la structure et des fonctions des éléments informationnels d'une entreprise (Sowa and Zachman 1992), tandis que les modèles d'entreprise tentent, quant à eux, d'aider à l'étude, la compréhension et la collection de la connaissance et du savoir de l'entreprise (Pohl 1993). Selon (Gustas 1995), la modélisation d'entreprise s'articule essentiellement autour de quatre axes principaux :

- **L'axe des objectifs :** regroupent l'ensemble des modèles décrivant les objectifs et les motivations régissant le développement d'autres modèles de l'entreprise. Ces modèles répondent au « Pourquoi » de la modélisation d'entreprise.
- **L'axe des acteurs :** cet axe définit les relations et dépendances reliant les différents acteurs entre eux lors de l'accomplissement de leur tâches et objectifs. L'ensemble des modèles de cet axe tente de répondre à la question « Qui ».
- **L'axe des activités :** cette composante traite le « Comment » de l'entreprise, elle définit l'ensemble des activités (actions ou tâches) capable d'appliquer des changements d'état sur les concepts manipulés par l'entreprise (ressource ou produit).
- **L'axe des concepts :** répond à l'interrogation « Quoi »; en décrivant l'ensemble des concepts manipulés, transformés et échangés dans l'entreprise, qu'il s'agissent de ressource ou de produits finis. Cette composante peut aussi traiter les interrogations « Où » et « Quand ».

La figure 14 illustre les différentes dépendances sémantiques et pragmatiques qui existent entre les différentes composantes de la modélisation d'entreprise selon (Gustas 1995).

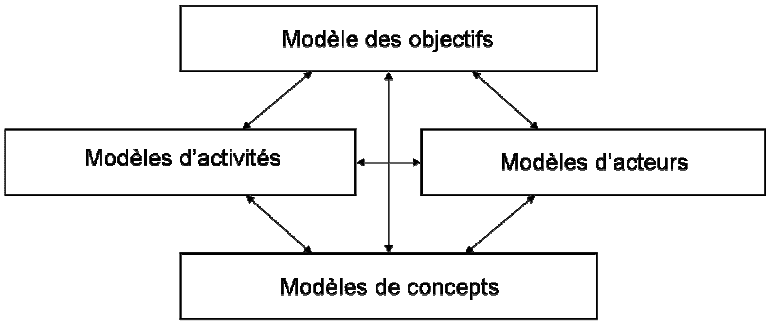


Figure 14. Les différents types de modèles d'entreprise selon Gustas (Gustas 1995)

En effet, on peut utiliser des modèles informationnels afin de représenter les axes relatifs aux objectifs, acteurs, activités ou même les concepts. Dans le cadre de cette thèse nous nous focalisons essentiellement sur l'axe « Concepts », et plus précisément, les modèles de produits. Nous présenterons dans le chapitre suivant, quelques uns des standards les plus reconnus pour la modélisation et la représentation des informations relatives au produit dans les entreprises de production.

Le Tableau 2 montre l'adéquation entre la typologie de modèles proposée par (Vernadat 1996) et celle proposée par (Gustas 1995)

Tableau 2 : Correspondances entre les classifications de (Vernadat 1996) et de (Gustas 1995)

	Objectifs	Acteurs	Activité	Concepts
Modèles économiques	☑			
Modèles Organisationnels		☑		
Modèles d'activités			☑	
Modèles de ressources				☑
Modèles de produits				☑
Modèles informationnels	Ou bien ☑	Ou bien ☑	Ou bien ☑	Ou bien ☑

Si nous considérons que les modèles dits informationnels sont la représentation électronique des informations et données pertinentes dans un cadre bien précis, on peut percevoir les modèles informationnels comme étant une implémentation possible d'un modèles dits non

informationnels. Chaque application ou système d'entreprise est ainsi basé sur un modèle informationnel qui lui-même fait référence à un modèle dit non informationnel.

Cette approche basée sur les modèles est étudiée par la communauté relative à l'ingénierie dirigée par les modèles dite *IDM* (en anglais, MDE : Model Driven Engineering). Cette technique a montré son efficacité dans le domaine du développement logiciel ; surtout dans le domaine de la transformation de modèles (Lemesle 1998; Roque 2005), des applications à grande échelle (Vega 2005) et du développement d'architectures logicielles (Manset et al. 2006). Cette discipline consiste justement à ne plus utiliser les modèles d'entreprise ou les modèles en général comme une simple documentation pour les systèmes développés, mais comme vraies entrées/sorties du processus de développement (Bézivin and Gerbé 2001 ; Bézivin 2005). Dans le contexte de l'IDM, les modèles du domaine étudié contiennent tout le détail nécessaire pour permettre une génération complète de l'implémentation du système. Les travaux et recherches de la communauté *IDM* ont donné lieu à l'architecture dite dirigée par les modèles (MDA : Model Driven Architecture) proposée par l'OMG (Object Management Group). L'approche MDA[9] est dite dirigée par les modèles parce qu'elle fournit les bases pour l'utilisation des modèles pour guider la compréhension, la conception, la construction, le déploiement, la maintenance et la modification des systèmes développés (Mellor et al. 2004). Les trois principaux objectifs du MDA sont la portabilité, l'interopérabilité et la réutilisabilité à travers la séparation des aspects dépendants de la plate-forme ou de l'application étudiée et des aspects plus abstraits indépendants de l'application. L'approche MDA se base sur quatre niveaux ontologiques (Naumenko and Wegmann 2003) (*cf*. Figure 15)**.**

[9] *http://www.omg.org/mda.*

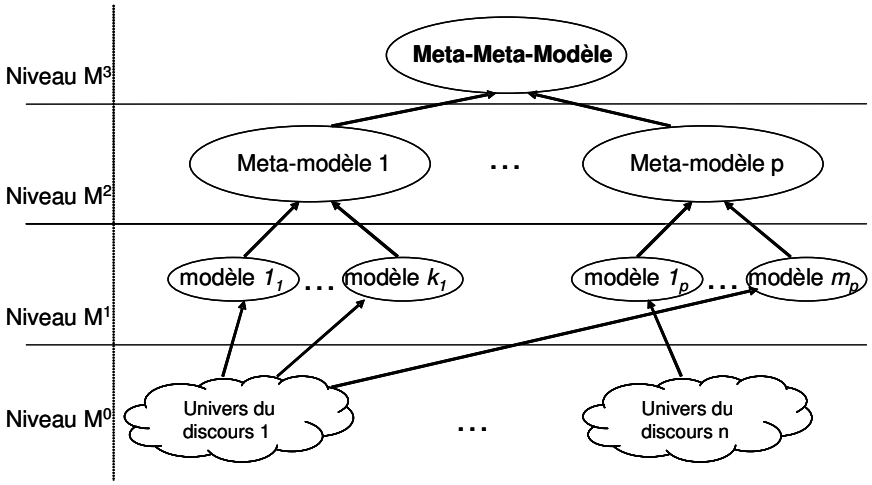


Figure 15. L'approche MDA et les quatres niveaux ontologiques.

Le niveau le plus bas M^0 représente les différents objets de modélisation aussi appelés « univers du discours ». Le niveau M^1 spécifie les différents modèles de chaque univers de discours. Le modèle d'un système est sa description ou sa spécification. Les modèles du niveau M^1 appartiennent à différents domaines d'intérêt relatifs aux univers du discours représentés par les modèles. Il est possible qu'un domaine d'intérêt recouvre plusieurs univers du discours. De même, des modèles provenant de différents univers de discours peuvent relever du même domaine d'intérêt. Le niveau M^2 représente les meta-modèles spécifiques à chaque domaine : un meta-modèle pour chaque domaine d'intérêt pertinent pour les modèles de niveau M^1. Finalement, le niveau M^3 présente le meta-meta-modèle et doit être conçu pour permettre la définition de tous les concepts nécessaires pour la modélisation de meta-modèles et pour leur unification dans un cadre de référence commun. Un meta-meta-modèle est indépendant du domaine, il contient des meta-caractéristiques pour des meta-modèles spécifiques.

Avec ces différents niveaux, le MDA offre une base théorique facilitant la manipulation et la transformation des modèles pour la formalisation des échanges dans le but de réaliser des solutions pour l'interopérabilité. Dans la section suivante, nous abordons la relation entre la modélisation des systèmes d'entreprise dans le cadre MDA (Système, modèle, meta-modèle) et l'interopérabilité de ses systèmes. Nous exposerons les différentes approches et techniques pour l'utilisation des modèles dans la mise en œuvre de l'interopérabilité des systèmes.

III.4. Synthèse

La modélisation en entreprise a souvent été utilisée comme un outil pour résoudre l'interopérabilité des systèmes d'entreprise, que ce soit dans le cadre de méthodes formelles basées des langages de modélisation pour l'alignement et l'unification des concepts du domaine étudié, ou bien dans le cadre d'approches plus pratiques basées plutôt sur la standardisation ou le développement d'activité supplémentaires pour l'ajustement mutuelle des modèles (Vallespir et al. 2005).

Selon les besoins à traiter, plusieurs types de modèles peuvent être réalisés dans l'entreprise ; des travaux comme ceux de (Gustas 1995) et de (Vernadat 1996) ont tenté d'identifier les grandes catégories de modèles d'entreprises, et ce en prenant en considération les concepts qu'il manipule et les objectifs du modèle. La modélisation d'entreprise permet d'établir la représentation d'une partie ou l'ensemble entreprise selon un point de vue bien défini, afin de :

- Capitaliser la connaissance et le savoir-faire de l'entreprise,
- Rationaliser et structurer les échanges d'information,
- Analyser, Concevoir et spécifier une partie de l'entreprise,
- Simuler le comportement d'une partie de l'entreprise,
- Contrôler, synchroniser ou coordonner certaines parties de l'entreprise.

Elle peut aussi être utiliser dans le cadre de l'ingénierie de solutions logicielles (applications, systèmes d'informations, etc.) parfaitement adaptées au contexte au savoir et aux besoins spécifiques de l'entreprise exprimés dans les modèles, ce type d'ingénierie est ainsi dit « dirigé par les modèles » (IDM). Nous étudions, dans la suite de ce chapitre, l'apport des approches dites dirigées par les modèles pour la spécification de solution pour l'interopérabilité.

IV. Modélisation et Interopérabilité

Dans cette section, nous exposons quelques unes des approches basées sur de la manipulation de modèles pour l'établissement de l'interopérabilité entre systèmes. La première approche appelée « *Intégration de modèles* » a fait son apparition bien avant la définition de l'architecture dirigée par les modèles MDA (Mellor et al. 2004) par l'OMG (Petrie 1992). La deuxième approche dite « *transformation de modèles* » a fait son apparition au milieu des années 90, dès les premières briques de l'approche MDA. Enfin, pour sa pertinence nous présentons aussi une approche pour l'interopérabilité dite dirigée par les ontologies.

IV.1. L'interopérabilité par intégration de modèles

Avant même la définition du MDA par l'OMG, des méthodes pour l'intégration de systèmes basées sur l'utilisation de modèles avaient déjà fait leur apparition. En effet, (Petrie 1992) définissait déjà trois approches de bases pour l'intégration utilisant des modèles maîtres, unifiés ou fédérés:

- ***Intégration basée sur un modèle maître.*** Dans cette approche, on définit un modèle de référence unique ; les modèles spécifiques de toutes les applications à intégrer sont instanciées à partir du modèle de référence selon les besoins de leurs utilisateurs, cette approche est dite descendante « top-down ». Certes cette approche contribue à l'intégration de l'entreprise, mais elle donne lieu à des solutions fortement couplées ; les applications restent très dépendantes du modèle de référence. En effet, un changement au niveau du modèle de référence nécessite une propagation en cascade au niveau de tous les modèles spécifiques.

Nous pouvons citer comme exemple le standard IEC 62264 (IEC 62264 2002) pour l'échange de données entre les applications du niveau logistique et business et les applications du niveau contrôle de la production. Ce standard est une tentative de définition de modèles de référence pour l'ensemble des données et structures de données manipulées et échangées avec le niveau contrôle de l'entreprise.

L'IEC 62264 est une norme comprenant plusieurs parties qui définit le contenu de l'interface entre les activités de l'entreprise et les activités de contrôle. La première partie définit des modèles d'objets d'échange d'information entre les systèmes d'entreprise et les systèmes de contrôle. La seconde partie, quant à elle, complète, de manière plus détaillée, les modèles d'objets par la définition d'attributs, de manière à instaurer les interfaces pouvant être mises en œuvre.

- ***Intégration basée sur l'unification.*** Avec cette approche dite ascendante « bottom-up », les modèles locaux peuvent être complètement différents. Cependant, une unification sémantique de tous les concepts partagés par les différents modèles est nécessaire. Il en résulte un meta-modèle représentant l'ensemble des concepts communs à l'ensemble des systèmes.

Comme exemple de ce type d'intégration on peut citer le langage unifié pour la modélisation d'entreprise UEML (UEML 2002 ; Berio et al. 2003). Le langage unifié pour la modélisation de l'entreprise est, typiquement, un exemple d'intégration par l'unification de langage de modélisation d'entreprise. En effet, UEML est construit autour d'un meta-modèle définissant un ensemble de concepts (Berio et al. 2003) identifié comme étant l'unification sémantique de quelques uns des langages de modélisations les plus répandus dans le monde de l'entreprise (GRAI, IEM, EEML, etc.).

- ***Intégration basée sur la fédération.*** Ici, l'intégration des systèmes se fait à la demande. En effet, cette approche est basée sur de l'interconnexion retardée (*late binding*) qui signifie qu'on n'effectue l'unification sémantique entre deux systèmes A et B que lorsque A sollicite B ou l'inverse. Cette solution permet d'économiser l'effort d'unification au moment de la modélisation des systèmes, et le décale dans le temps jusqu'au moment de leur sollicitation.

IV.2. L'interopérabilité par transformation de modèles

Dans le contexte MDA, le problème de résolution de l'interopérabilité entre systèmes se traduit en un problème d'échange d'informations depuis un modèle vers un autre. En effet, étant donné que le MDA considère que chaque système est représenté par un modèle, l'interopérabilité entre deux systèmes dans le cadre d'une communication, se résout alors en identifiant le moyen de faire communiquer les deux modèles.

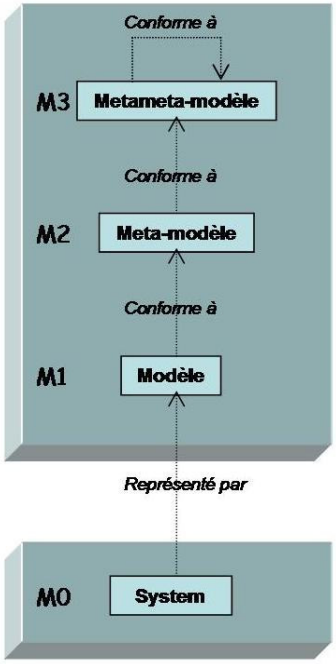


Figure 16. Construction des systèmes dans l'ingénierie MDE (Bézivin 2005).

Il est donc nécessaire de disposer d'un moyen pour transformer des instances (ou données) allant d'un modèle source vers un modèle destination. La transformation des modèles est une discipline sujette à de nombreux travaux que ça soit dans le domaine dit « *schema matching* » (pour les schémas de données) (Rahm and Bernstein 2001), ou dans le domaine de la « *web sémantique* » et le besoin de l'interrogation des données distribuées à travers le web (Doan et al. 2002). Cependant la majorité des solutions proposées dans la littérature, s'intéresse souvent aux concepts plus qu'aux associations les reliant, les relations sont en effet souvent mises en second plan.

Dés 1996, (Blaha and Premerlani 1996) propose une théorie pour la transformation de modèles dans le cadre de modèles orientés objet. La majeure contribution de ce travail concerne l'identification et la documentation d'un ensemble de *transformations élémentaires* permettant la transformation de modèles. Selon les auteurs, il existe trois principaux types de transformation entre deux modèles:

1. *Transformation d'équivalence*. Exprime une relation d'équivalence entre l'élément source appartenant au modèle source et son image dans le modèle destination. Ce genre de transformation permet de conservée l'ensemble des informations propre à

chacun des éléments transformés. Cependant les relations entre les objets dans le modèle source ne sont par forcément conservées.

2. *Transformation avec perte d'informations*. Le modèle source est plus précis et plus contraint que le modèle destination ; toutes les instances du modèle source peuvent être traduit en instance du modèle destination mais l'inverse n'est pas toujours possible.

3. *Transformation avec gain d'informations*. Le modèle source est moins précis et moins contraint que le modèle destination ; certaines instances du modèle source peuvent ne pas avoir de correspondant dans le modèle destination.

Avec l'avènement de l'aire MDA, la transformation de modèles devient de plus en plus pertinente. Ainsi (Lemesle 1998) présente une technique de transformation basée sur la représentation des deux meta-modèles source et destination dans le formalisme sNets ; sous forme de réseaux sémantiques dotés de nœuds typés (Bézivin et al. 1994). Selon ces travaux, les meta-modèles sources et destination sont décrits dans le même formalise ; en l'occurrence sNets ; le passage de l'un à l'autre se fait grâce à des règles textuelles définissant les correspondances entre la source et la destination (*cf*. Figure 17).

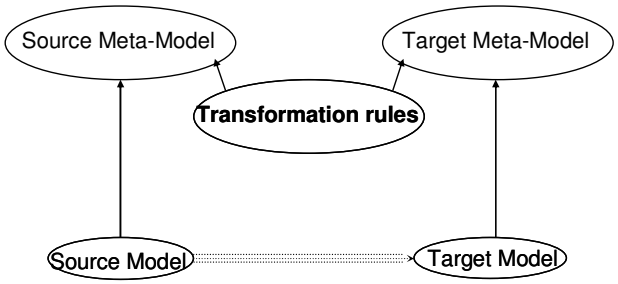


Figure 17. Transformation de meta-modèle (Lemesle 1998).

Enfin, (Denivaldo et al. 2005) propose un meta-modèle pour la spécification des mappings entre meta-modèles pour structurer la définition des règles de transformations et permettre ainsi le développement d'une approche pour la transformation automatique de modèles. Dans son approche il présente une étude des règles de transformation de modèles dans le contexte MDA. Les auteurs identifient ainsi, différents types de règles, et ce selon le nombre d'éléments faisant partie de la source et de la destination pour chacune des transformations :

- Les régles A2B exprimant une transformation d'un élément source vers un élément destination (1→ 1),
- Les règles A2B* exprimant une transformation d'un élément source vers plusieurs éléments destination (1→ n),
- et finalement les règles A*2B pour exprimer le cas où plusieurs éléments source sont assemblés pour correspondre un seul élément destination (n→ 1).

Pour la mise en œuvre de solutions pour l'interopérabilité des systèmes, outre la transformation de modèles et l'approche dirigée par les modèles, les ontologies sont elles aussi de plus en plus utilisées. Dans le paragraphe qui suit, nous introduisons brièvement les ontologies ainsi que quelques exemples d'utilisations des ontologies dans le domaine de l'entreprise pour l'interopérabilité.

IV.3. Ontologies d'entreprise, un outil pour l'interopérabilité

Dans cette section nous tentons de donner quelques définitions pour le terme « Ontologie » puis nous exposons quelques exemples d'utilisation des ontologies pour l'interopérabilité dans l'entreprise.

IV.3.1. Définitions

Une définition qui fait autorité dans le domaine des ontologies, est celle proposée par Gruber en 1993 :

Définition 7. *Une ontologie est une spécification explicite d'une conceptualisation partagée dans un domaine de connaissance*. (Gruber 1993)

Cette définition s'appuie sur deux dimensions :

- la *conceptualisation* d'un domaine, c'est-à-dire un choix quant à la manière de décrire un domaine.
- la *spécification* de cette conceptualisation, c'est-à-dire sa description formelle.

Une ontologie est donc une base de formalisation des connaissances, elle se situe à un certain niveau d'abstraction et dans un contexte particulier. Elle est aussi une représentation d'une conceptualisation partagée et consensuelle, dans un domaine particulier et vers un objectif commun. Une ontologie classifie en catégories les relations entre les concepts.

L'étymologie renvoie à la « *théorie de l'existence* », c'est à dire la théorie qui tente d'expliquer les concepts qui existent dans le monde et comment ces concepts s'imbriquent et s'organisent entre eux. Contrairement à l'homme, la connaissance pour un système informatique se limite à la connaissance qu'il peut représenter. Chez l'homme, les connaissances représentables (c'est-à-dire, *l'univers du discours* ou *domaine de discours*) sont complétées par des connaissances non exprimables (sensations, perceptions, sentiments non verbalisables, connaissances inconscientes, etc.). Ces éléments non représentables participent pourtant aux processus de raisonnement et de décision. Les performances cognitives d'un système vont donc en partie reposer sur le champ des représentations auquel il aura accès, c'est-à-dire concrètement au champ des représentations qui aura été formalisé. Les ontologies informatiques sont des outils qui permettent précisément de représenter un corpus de connaissances sous une forme utilisable par une machine.

IV.3.2. Ontologies et Interopérabilité dans l'entreprise

La modélisation dans l'entreprise a permit la structuration du savoir et des connaissances de l'entreprise en vue d'un éventuel échange d'informations entre différents systèmes communicants. Cependant, il est nécessaire de distinguer le modèle lui-même et l'information qu'il contient. En effet, un modèle est toujours exprimé dans un formalisme qui lui est propre et structuré selon un cadre spécifique. Le modèle d'entreprise contient donc, une connaissance générique propre au domaine, et des informations spécifiques concernant l'entreprise. Les ontologies offrent le moyen de décrire les connaissances du domaine tout en restant indépendante du formalisme utilisé pour l'incarner. Plusieurs travaux de recherches tente d'exploiter le formalisme des ontologies pour mettre en œuvre des solutions pour l'interopérabilité sémantique et organisationnelle que ce soit dans le domaine de l'entreprise, ou d'autres domaines tels que la web sémantique, (Stumme and Maedche 2001 ; Kalfoglou and Schorlemmer 2003; Silva and Rocha 2003; Elbyed 2006).

L'objectif d'une ontologie d'entreprise est de fournir une définition consensuelle pour chacun des éléments du vocabulaire d'un *domaine de discours*, et ce afin de permettre l'unification sémantique des différents concepts représentés dans différents modèles sous différents formalismes, unification nécessaire dans le cadre de l'interopérabilité et l'intégration de modèles.

Pour identifier le vocabulaire du domaine auquel appartient une entreprise, et pouvoir ainsi échanger les connaissances des entreprises de façon non ambiguë, plusieurs travaux de recherche ont tenté de formaliser des ontologies spécifiques aux entreprises et leurs domaines d'activités. Nous exposons quelques uns des résultats les plus importants (Boudjlida et al. 2006).

Edinburgh Enterprise Ontology ou, plus communément, *Enterprise Ontology* (***EO***) est une collection de définitions des termes et concepts en relations avec le monde des entreprises (Uschold et al. 1997). Ces concepts sont largement répandus pour la description des entreprises en général et peuvent ainsi être utilisés pour la spécification des applications et systèmes d'entreprises. Cette ontologie est le résultat des travaux de l'institut des applications d'intelligence artificielle à l'université d'Edinburgh (Artificial Intelligence Applications Institute) et de nombreux partenaires dans le cadre d'un effort collaboratif pour l'établissement d'un cadre de modélisation pour l'intégration des méthodes et outils de l'entreprise. Les objectifs principaux de *EO* sont:

a. Garantir une communication fluide entre les participants, en facilitant le partage d'une compréhension unifiée autour des modèles d'entreprise.

b. Offrir une infrastructure stable tout en étant adaptable pour le partage et la spécification des besoins des modèles d'entreprise.

c. Accomplir l'interopérabilité entre différents outils dans un environnement de modélisation d'entreprise en utilisant un format d'échange commun.

Dans le domaine de la modélisation des processus, *PSL*[10] *(ISO 18629)* est un langage formel défini par un lexique (ensemble de symboles) et une grammaire (spécification de la façon dont les symboles peuvent être combinés). La grammaire utilisée pour PSL est basée sur KIF

[10] *http://www.mel.nist.gov/psl/*

(Knowledge Interchange Format), un langage formel, utilisant une logique du premier ordre, développé pour l'échange de connaissance entre machines ayant différentes représentations. Actuellement, PSL contient plus de 330 concepts et 46 extensions. À la base de PSL se trouve une ontologie qui permet une définition rigoureuse et non ambiguë pour chacun des concepts nécessaires pour l'échange d'information relative aux procédés dans un contexte de production tels que : Activité, objet ou occurrence d'activité, ces concepts constituent le noyau de l'ontologie PSL. Cependant, le noyau de PSL contient les concepts minimaux pour l'échange de spécifications très simplifiées. PSL propose donc un mécanisme pour permettre l'extension des concepts de l'ontologie par de nouvelles définitions.

L'objectif principal du projet *TOVE*[11], quant à lui, fut de proposer une ontologie définissant le sens de chaque concept dans une entreprise, et d'offrir une implémentation pour cette ontologie. Cet objectif se décompose en quatre activités distinctes :

a. Créer une représentation partagée de l'entreprise ; chaque agent de l'entreprise distribué devrait pouvoir la comprendre et l'utiliser facilement (aka ontology),

b. Définir le sens et la sémantique de chaque élément de cette ontologie (aka semantics),

c. Implémenter ces sémantiques en utilisant un ensemble d'axiomes permettant à l'outil TOVE de déduire la réponse à certaines questions automatiquement,

d. Définir les symboles à utiliser pour la représentation graphique de chaque concept.

IV.4. Synthèse

Dans cette section, nous avons présenté quelques unes des approches orientées par les modèles pour la mise en place de solution pour la mise en place de l'interopérabilité entre applications utilisant chacune un modèle spécifique ; les premiers travaux dans ce sens sont liés à l'intégration de modèles (Petrie 1992) ; exemple : les approches pour l'intégration basée sur l'unification, l'intégration basée sur un modèle maître ou l'intégration basée sur un modèle maître. Par la suite, d'autres travaux issus de l'ingénierie dirigée par les modèles

[11] *http://www.eil.utoronto.ca/enterprise-modelling/tove/*

(Mellor et al. 2004) ont tenté a apporté eux aussi des réponses à ce problème, nous citons à titre d'exemple les travaux de (Lemesle 2000; Denivaldo et al. 2005; Roque 2005)

Nous avons ainsi montré l'apport de la méthode dirigée par les modèles pour l'interopérabilité des applications. En effet, la manipulation des modèles permet de spécifier les correspondances sémantiques existant entre les différents modèles d'applications. Plusieurs travaux en été réalisés pour identifier ces correspondances sémantiques entre modèles :

Par la suite, nous avons présenté l'approche basée sur l'utilisation des ontologies d'entreprise pour l'amélioration de l'interopérabilité, puis nous l'avons illustrée par quelques exemples (*EO, PSL* et *TOVE*). En effet, les ontologies d'entreprise permettent l'unification des termes, concepts et vocabulaires utilisés pour la coopération et les échanges d'information entre différentes entreprise.

V. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la problématique générale de la thèse, liée à l'interopérabilité entre applications dans le monde de l'entreprise. Nous avons aussi étudié le lien existant entre la spécification de l'interopérabilité entre applications et la modélisation de ces applications. Nous avons ainsi montré quelques unes des approches dirigées par les modèles visant à répondre au besoin d'interopérabilité.

Dans la suite de ce mémoire, nous nous focalisons sur une problématique plus restreinte concernant l'interopérabilité entre systèmes et applications manipulant des données ou informations relatives aux produits d'une entreprise. En effet, la modélisation du produit dans l'entreprise fait intervenir de nombreux modèles, langages, et représentations. Chaque modèle correspond à une vue spécifique des objets manipulés.

Notre contribution vise à mettre en œuvre une approche pour l'interopérabilité orientée produit entre les différents modèles et représentations des produits de l'entreprise en établissant un modèle de référence pour spécifier et formaliser les correspondances sémantiques entre ces différents modèles (*cf.* Figure 18). L'objectif étant de maintenir la cohérence entre les différentes représentations du produit dans les différentes applications.

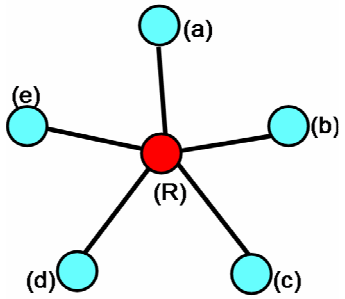


Figure 18. Interopérabilité avec modèle de référence.

Nous nous intéressons ainsi à la formalisation d'un modèle de référence permettant la représentation générique du produit en tant que partie informationnelle et partie physique. Pour maintenir la cohérence modèle/réalité des mécanismes devront être spécifiés au niveau

des processus eu même afin de mettre à jour correctement les vues informationnelles en cas de modification physique.

L'approche que nous proposons est une solution hybride entre l'approche basée sur un modèle maître, et l'approche basée sur l'unification de modèles rencontrées dans l'état de l'art (*cf.* section IV.1). En effet, contrairement à l'approche par modèle maître, les différents modèles spécifiques à chacun des systèmes et applications ne seront pas instanciés directement du modèle de référence, par ailleurs le modèle référence proposé n'est pas le résultat de l'unification des différents modèles comme le stipule l'approche basée sur l'unification.

Par ailleurs, afin d'assurer l'interopérabilité entre les différentes applications manipulant les données du produit, et permettre ainsi le maintien de la cohérence entre les différentes vues du produit tout en prenant en compte le modèle de référence préalablement établi, nous formaliserons les mécanismes de mappings et transformations de modèles nécessaires.

Le but de l'approche globale étant de proposer une approche facilitant la prise en compte, dès la phase de modélisation, de la problématique d'interopérabilité entre systèmes se référant aux différentes vues du produit, qu'il s'agisse de la vue physique qui peut subir des transformations physiques, des transports, etc., ou de la vue informationnelle mettant en œuvre des traitements pour la création de nouvelles informations ou la mise à jour d'informations préexistantes.

Dans le chapitre qui suit, nous introduisons les besoins de représentations des données relatives au produit dans le cadre de la considération du produit dit « intelligent », par la suite, nous présentons quelques exemples de modèles et langages pour la représentation d'informations relatives au produit pour différentes utilisations. Dans le même chapitre nous introduisons le modèle Bunge-Wand-Weber (BWW) que nous utiliserons par la suite comme base théorique pour notre modèle de référence.

Chapitre 2 :
Un modèle de référence pour la représentation du produit

I. Introduction

L'interopérabilité relative aux données décrivant les produits entre applications d'entreprise s'appuie, souvent, sur des échanges de données à travers l'utilisation de standards d'échange, de formats de données et de modèles spécifiques. En effet, dans l'entreprise de production moderne, la représentation des produits ainsi que la gestion et l'échange de données et d'informations propres au produit prend de plus en plus d'importance au sein des systèmes d'informations, des systèmes de gestion de production ou tout autre type d'applications. La représentation et les échanges de données du produit sont les éléments essentiels pour construire des systèmes intégrés autour du produit, où même le produit est amené à interopérer avec les systèmes évoluant dans son environnement. En quelques décennies seulement le produit est passé du statut d'un simple objet inerte, au statut d'un objet dit *« intelligent »* participant activement à sa propre destinée. Dans le cadre de la modélisation du produit, deux aspects sont à noter :

- *l'aspect descriptif* :

  Comment représenter les caractéristiques et les connaissances du produit, puisqu'on parle d'un produit dit « intelligent » ?

- *l'aspect comportemental* :

  Comment ce produit intelligent va pouvoir interagir et échanger des informations avec son environnement ?

Nous nous intéressons ainsi à la modélisation des informations relatives au produit et non à la technologie d'implémentation. En effet, plusieurs technologies permettent d'implémenter le paradigme de « *produit intelligent* », mais peu de moyens pour la modélisation de ces produits sont mis à disposition de l'utilisateur pour structurer et modéliser le concept de produit intelligent dans son environnement.

II. Représentation des informations du produit

Dans cette section, nous nous focalisons sur le besoin de représentation des informations relatives au produit dans l'entreprise, nous présentons également quelques uns des standards, normes ou formats permettant la représentation de données relatives au produit et leur échange entre systèmes et applications d'entreprise.

II.1. Le « produit intelligent »

Le Produit est devenu un acteur indépendant, ayant ses propres modèles, informations, et connaissances au sein de l'entreprise. Nous pouvons, ainsi, considérer qu'il acquiert un rôle lui permettant de « contribuer » directement dans la prise de décisions affectant son propre cycle de vie. Depuis quelques années, on parle de plus en plus d'objets intelligents, produits intelligents ou machines intelligentes ; plusieurs travaux ont eu pour objectif de définir le profil de l'objet intelligent capable de contrôler par lui-même l'ensemble des paramètres et des interactions qui entrent en jeu pour sa propre transformation depuis l'état de matière première jusqu'à l'état de produit fini prêt à l'utilisation entre les mains de l'utilisateur final, voir même jusqu'à son recyclage en fin de vie (Mc Farlane et al. 2002; Wong et al. 2002).

Ces dernières années, le terme *« Intelligent »* a été largement appliqué dans le domaine des systèmes de production. L'initiative IMS (Intelligent Manufacturing Systems) a ainsi contribué à la diffusion de résultats de recherche dans ce domaine. Le terme « *Produit Intelligent* » est utilisé, généralement, pour désigner un produit doté d'une certaine technologie lui permettant de posséder un ensemble de capacités. En effet, selon (Wong et al. 2002) un produit intelligent présente une partie ou la totalité des caractéristiques suivante :

1. Une identification unique
2. aptitude à communiquer avec l'environnement
3. aptitude à retenir ou enregistrer de l'information le concernant
4. aptitude à utiliser un certain langage pour afficher ses caractéristiques et ses besoins de production.

5. aptitude à prendre ou participer à la prise de décision concernant sa propre destinée.

En se basant sur ces caractéristiques, (Wong et al. 2002) définissent deux niveaux d'intelligence chez le produit :

Intelligence de niveau 1 : Ce niveau d'intelligence permet au produit de connaître et communiquer son statut (forme, composition, emplacement, etc.). Cette intelligence est plutôt *orientée information.* Ce niveau couvre principalement les caractéristiques 1 à 3 présentées précédemment.

Intelligence de niveau 2 : En plus des fonctionnalités de niveau 1, l'intelligence de niveau 2, permet au produit d'influencer et contrôler le déroulement des opérations le concernant (production, stockage, distribution, etc.). A ce niveau, l'intelligence est dite *orientée décision.* Ce niveau couvre la totalité de caractéristiques d'intelligence énumérées précédemment.

Dans le cadre de ce chapitre ainsi que dans la suite de la thèse, nous nous positionnons dans le contexte de l'approche orientée information, participant à une intelligence dite de niveau 1. Il s'agit, donc, d'étudier les différents types d'informations associées aux produits ainsi que la formalisation d'un modèle générique pour représenter l'ensemble de ces informations tout au long de leur existence.

II.2. Exemples de représentation du produit

Nous exposons brièvement, quelques exemples pour la représentation des données du produit. Ces exemples sont issus de l'utilisation des données techniques du produit dans les différentes tâches assistées par ordinateur. En effet, (Pierra 2000) identifie deux grandes catégories dans ce que l'on appelle les données techniques du produit:

- les informations qui décrivent les produits de l'entreprise aux différentes phases de leur cycle de vie et selon les points de vue propres aux différents métiers, et
- les informations qui se réfèrent aux composants préexistants, qui sont utilisées pour concevoir, fabriquer ou maintenir les produits de l'entreprise.

Si ces deux types de données ont de nombreux points communs (par exemple le fait qu'elles décrivent, dans les deux cas, des objets techniques), plusieurs aspects les rendent différents :

- L'origine de l'information : L'information sur les produits de l'entreprise est créée dans l'entreprise elle-même. La description et la connaissance sur les composants proviennent de leurs fournisseurs. Permettre un échange purement numérique d'information entre fournisseurs et utilisateurs de composants augmenterait la qualité et réduirait considérablement les coûts de tous les partenaires impliqués dans l'entreprise étendue.
- La structure de l'information : Chaque produit réalisé par l'entreprise est individuellement, explicitement et exhaustivement représenté pour permettre sa fabrication. Cependant, pour faciliter leur recherche et leur sélection, et pour éviter l'explosion des données, les composants sont regroupés en familles d'objets similaires dans lesquelles chaque instance est seulement implicitement décrite à l'aide de tables, d'expressions ou de formules.

Pour répondre au besoin de partage de données techniques nécessaire entre les différents systèmes impliqués tout au long du cycle de vie du produit, plusieurs standards et normes ont vu le jour pour proposer des représentations du produit communes à l'ensemble de l'entreprise. Les données techniques des produits constituent la nouvelle représentation du savoir-faire de l'entreprise étendue donc le patrimoine informationnel de la nouvelle entreprise (El Hadj Mimoune 2004). La mise en œuvre de l'échange et du partage des données a amené les industriels à créer de nombreux standards. En effet, la recherche d'un format commun pour les données issues de la XAO (ensemble des procédés d'entreprise assistés par ordinateur) a été un souci récurrent des industriels dans le monde entier depuis les années 70. Voici quelques uns des exemples les plus connus :

STEP, le projet STEP (ISO 10303 1994) a eu pour objectif définir une représentation non ambiguë des données techniques du produit, interprétable par tout système informatique, et couvrant tout le cycle de vie des produits. STEP comporte la définition d'un langage formel de spécification des données ; le langage EXPRESS (ISO 10303-11 1994), ainsi que la définition d'un format neutre d'échange et de stockage des données décrites dans le langage EXPRESS (ISO 10303-21 1994). STEP propose ainsi un ensemble de modèles de références (protocoles d'application), explicités dans le langage EXPRESS, spécifiant une vue particulière des modèles de produits selon un point de vue métier donné.

SET, le Standard d'Echange et de Transfert est un standard français lancé en 1983 par Aérospatiale (SET 1989). Il représente une solution aux exigences relatives à l'échange de données entre différents systèmes de CFAO, et au besoin d'archiver ces données.

IGES, proposé par le National Institute of Standards and Technology (NIST), est un standard pour l'échange des dessins techniques. Il est utilisé principalement dans le domaine de la mécanique et de l'aéronautique (IGES 1980).

Il existe aussi des standards dits « de fait », nous citons à titre d'exemple DXF produit par AUTODESK, qui édite le logiciel AUTOCAD, pour la sauvegarde des modèles de CAO (Conception Assistée par Ordinateur) d'AUTOCAD. Le succès qu'a connu le logiciel AUTOCAD dans l'industrie a élevé ce format au rang de standard largement utilisable pour échanger des modélisations de DAO. DXF permet d'échanger sous forme numérique des plans de dessin industriel ; des entités géométriques bi ou tridimensionnelles, des annotations de dessin technique.

II.3. Synthèse

Le besoin incessant lié à l'informatisation et l'électronisation du produit de l'entreprise contemporaine se traduit par les innombrables vues et représentations du produit appartenant aux différents systèmes et applications de l'entreprise. La majorité des standards existant, incluant ceux exposés ci-dessus, se focalisent essentiellement sur la description statique des données techniques du produit telle qu'elles sont définies à la conception du produit. En effet, les données représentées par ces normes et standards sont souvent relatives à la phase de conception des produits à l'aide d'applications de CAO. Cependant, tout au long du cycle de vie du produit, suite à des transformations physiques ou suite à des traitement informationnels certaines données relatives au produit peuvent être mises à jour et changer de valeur. Pour des besoins tels que le contrôle de qualité des produits finis, ou la gestion de la traçabilité et de la généalogie des produits, prise en compte de ces données « *dynamiques* » peut s'avérer indispensable. Ainsi, que la cohérence entre les différentes représentations du produit d'un côté et la réalité physique du produit lui même.

Selon le principe de la matérialité énoncé par (Frachet 1987), *« un objet du monde réel est porteur de l'ensemble de ses vues et en assure la cohérence ».* Cependant, la considération de

l'objet intégrant ses propres vues dans un environnement de production, ne suffit pas à elle seule à maintenir la cohérence des représentations au sein des systèmes et applications. En effet, chaque système ou application s'intéresse aux données et informations relatives à une ou plusieurs vues du produit. Il est donc nécessaire d'avoir des mécanismes permettant de maintenir la cohérence entre les différentes vues et représentations du produit manipulées par chacun des systèmes ambiants, tout en assurant la synchronisation de l'ensemble avec la réalité physique du produit.

Pour répondre à ce besoin, nous proposons dans cette thèse une approche basée sur un modèle générique pour la représentation structurelle du produit, et permettant la prise en compte des interactions entre ces produits et les processus impliqués dans leur cycle de fabrication. L'objectif n'est pas de définir un nouveau standard de modélisation, mais à un niveau plus conceptuel, de capitaliser les connaissances dynamiques et statiques du produit, afin de constituer la représentation du produit la plus proche de son état réel. Nous montrerons aussi comment cette représentation unifiée, peut être maintenue cohérente avec la réalité du produit. Par la suite, nous définirons les mécanismes nécessaires pour permettre à cette représentation d'être l'élément unificateur des représentations du produit, permettant ainsi le maintien de la cohérence entre ses différentes vues et la réalité du produit lui-même.

III. Un modèle de référence pour les informations du produit.

Dans cette section, nous nous intéressons à la formalisation d'un modèle de référence permettant la représentation générique du produit en tant que partie informationnelle et partie physique. Pour maintenir la cohérence représentation/réalité des mécanismes devront être spécifiés au niveau des processus eu même afin de mettre à jour correctement les vues informationnelles en cas de modification physique.

La construction d'une représentation de référence facilite la mise en cohérence des différentes représentations du produit appartenant à chacune des systèmes ambiants. En effet, grâce au modèle de référence la synchronisation des différentes représentations est optimale.

Soit a, b, c d et e cinq applications pouvant s'échanger des données ou informations dans un environnement de collaboration. La figure 19***(a)*** illustre un graphe complet exprimant l'ensemble des échanges nécessaires pour synchroniser les représentations du produits dans les 5 applications; chaque arc représente un échange d'information bidirectionnel, le nombre d'échanges est ainsi donc égal à n*(n-1), la complexité du graphe est d'ordre $O(n^2)$. Dans le cas d'échange avec de communication avec modèle de référence, le nombre d'échange est de 2n, la complexité du graphe est d'ordre $O(n)$.

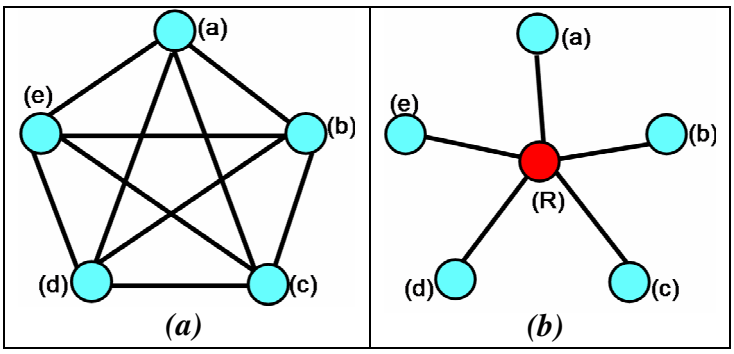


Figure 19. Réseau de communication avec sans modèle de référence.

Par la suite, nous établirons les mécanismes permettant l'échange d'informations entre la représentation issu du modèle de référence et l'ensemble des représentations du produit dans

les systèmes ambiants. Le but de l'approche globale étant de proposer une approche facilitant la prise en compte, dès la phase de modélisation, de la problématique d'interopérabilité orientée produit entre systèmes se référant aux différentes vues du produit, qu'il s'agisse de la vue physique qui peut subir des transformations physiques, des transports, etc., ou de la vue informationnelle mettant en œuvre des traitements pour la création de nouvelles informations ou la mise à jour d'informations préexistantes.

III.1. L'holon, un modèle pour le produit intelligent

Pour modéliser le produit, ses échanges avec son environnement, les informations qu'il détient et ses caractéristiques, nous considérons l'entité de modélisation représentant le couple (objet physique; modèle). Pour que cette entité puisse représenter un produit dit intelligent ; une intelligence de niveau 1, selon le sens (Wong et al. 2002) ; il est nécessaire qu'elle puisse prendre en compte les fonctionnalités nécessaires et suffisantes pour ce niveau d'intelligence, à savoir :

1. Une identification unique
2. l'aptitude à enregistrer et stocker un certain nombre d'information la concernant
3. l'aptitude à communiquer avec l'environnement ambiant à fin d'échanger des informations et mettre à jour ses connaissances.

Il faut donc que le modèle du couple (objet physique, modèle) soit le moyen de représenter les données à enregistrer. Les travaux de (Morel et al. 2003) proposent d'agréger cette dualité partie physique/partie informationnelle au sein du concept unificateur d'holon. La figure 20 illustre une représentation abstraite en UML de l'holon en tant qu'association d'une partie physique et une partie informationnelle (Gouyon et al. 2004).

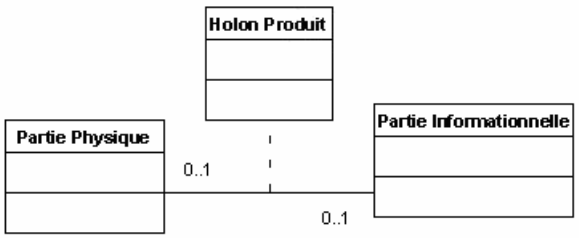


Figure 20. Modèle initial de l'holon en tant qu'agrégation physique/informationnelle.

Dans notre cas, le terme holon représente le concept de modélisation nous permettant d'agréger les représentations conceptuelles et de la partie physique correspondant à un seul et unique produit, il représente ainsi l'entité recomposée du couple (objet physique, modèle). Le but étant de simplifier la modélisation des caractéristiques physiques et conceptuelles du produit tout au long de son cycle de vie. L'holon est donc la fusion du produit en tant qu'entité réelle et de ses images virtuelles telle qu'elles sont perçues par les systèmes ambiants. Il joue dès lors le rôle d'interface informationnelle entre les différents systèmes ambiants dans un environnement de production (machines, applications, systèmes d'information, opérateurs humains, etc.), et le produit ; unité évoluant au sein même de cet environnement de production (*cf.* Figure 21).

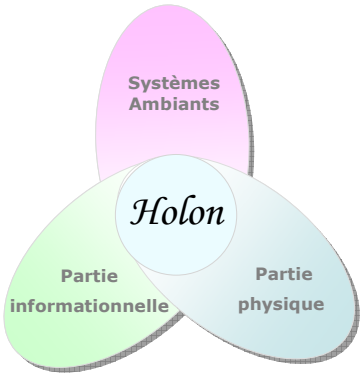


Figure 21. L'holon, interface entre le produit et les systèmes ambiants.

Le concept d'holon a été introduit par le philosophe Koestler dans le but d'expliquer l'évolution des systèmes sociaux et biologiques (Koestler 1967). Durant leur évolution, ces systèmes ont développé des formes intermédiaires stables, indépendantes et autosuffisantes. D'autre part, dans ce genre de systèmes, il est souvent difficile de dissocier le système dans sa totalité de l'ensemble des parties qui le constituent. En effet, pratiquement tout élément peut être vu comme un tout, et en même temps comme un ensemble de parties regroupées pour former ce tout. Ces observations ont poussé Koestler à proposer le terme *holon* pour parler du tout et des parties composant ce tout. *Holon* est en effet l'agrégation du mot grec « *Holos* » signifiant « tout » et le suffixe grec « *on* » signifiant particule ou partie (neutron, proton, positron, etc.). Koestler a observé que dans les organisations sociales, il n'existe pas d'entité n'interagissant pas avec les autres entités du même environnement. Il a ainsi défini l'*holarchie* comme étant une hiérarchie d'holons autorégulateurs, ayant certaines caractéristiques : (a) autonomes et contrôlant leur parties internes, (b) dépendants et contrôlés

par les entités de plus haut niveau, (c) interagissant et coordonnant avec l'environnement extérieur.

Dans les environnements de production, les systèmes holoniques ont fait leur apparition dès le début des années (HMS 1994; Seidel and Mey 1994 ; Bajic and Chaxel 1997).Le paradigme d'holon a été largement appliqué dans le domaine de la production pour la mise en place d'un système manufacturier intelligent (Valckenaers 2001). Plusieurs modèles ont ainsi été proposés ; à titre d'exemple, on peut citer PROSA (Van Brussel et al. 1998; Wyns 1999) ou MetaMorph (Maturana 1997; Maturana et al. 1999). Le concept d'holon a aussi été introduit dans des domaines autres que les systèmes de production, par exemple le projet Robot Soccer (Candea et al. 2000), ou encore dans les systèmes multi agents (Gerber et al. 1999).

Pour la construction de notre modèle de référence basé sur le concept d'holon, nous nous appuyons sur les concepts définis dans l'ontologie de haut niveau BWW (Wand and Weber 1993; Wand and Weber 1995), qui elle-même est construite à partir des travaux de Bunge, portant le nom de son auteur, le philosophe autrichien Mario Bunge (Bunge 1977; Bunge 1979). L'ontologie de Bunge a été à la base de approche de modélisation de l'univers en décrivant les objets par leurs propriétés. Cette approche est appelée communément l'approche de modélisation orientée propriétés. Nous choisissons BWW pour deux raisons, la première consiste en la simplicité des concepts qu'elle introduit, la seconde est sa généricité, en effet elle permet aussi bien de décrire des choses concrètes que des choses plus conceptuelles.

III.2. BWW : une base pour la représentation des choses et des objets

La modélisation par propriétés est une des caractéristiques de l'approche objet dont l'origine remonte aux années 60 avec le langage SIMULA (Dahl and Nygaard 1966 ; Birtwistle et al. 1973). Cette approche a été utilisée sous le nom de modélisation dirigée par les propriétés (Property Driven Model) (Barthes et al. 1979). Elle a été également influencée par le modèle de "frames" de (Minsky 1975) (ce sont des granules de connaissances plus importantes que les nœuds d'un réseau sémantique), et le modèle sémantique d'Abrial (Abrial 1974). Cette approche considère que le monde est constitué d'entités (les objets) ayant des propriétés qui sont également des objets (Shen and Barthes 1995).

L'objectif initial de la modélisation par propriétés était de pouvoir accommoder un grand nombre d'objets changeant dynamiquement et de les stocker de façon permanente dans une base de données. Cette modélisation comporte une structure récursive à deux niveaux (objet, attribut), dans lequel les attributs eux-mêmes sont des objets. Les autres facettes possibles de l'attribut deviennent alors des attributs de l'objet attribut.

Ainsi le monde réel est considéré comme étant constitué d'objets ou d'entités, ces entités elles-mêmes possédant un certain nombre d'attributs ou propriétés. On distingue deux types d'attributs objets :

- propriétés de structure : elles pointent sur d'autres objets ;
- propriétés terminales : elles pointent directement sur une valeur.

Les propriétés et les modèles (les classes) sont à leur tour représentés comme des objets. De plus, certaines propriétés peuvent être partagées entre les objets.

Les concepts à l'origine de la modélisation par les propriétés ont été introduits par l'ouvrage de Bunge intitulé "*Treatise on Basic Philosophy"*. Dans son manuscrit, Bunge traite en deux volumes l'ontologie dans laquelle l'auteur établit un ensemble de concepts abstraits par le biais desquels il construit une représentation abstraite des phénomènes du monde réel (Bunge 1977; Bunge 1979). Par la suite, Wand et Weber ont étendu la philosophie de Bunge pour l'adapter au domaine des systèmes d'informations, et ce en utilisant l'ontologie de Bunge pour la spécification des éléments appartenant au monde réel tels qu'ils sont perçus et représentés dans les systèmes d'informations (Wand and Weber 1993; Wand and Weber 1995). Leurs travaux seront connus par la suite sous le nom de l'*ontologie BWW (Bunge-Wand-Weber). BWW* reprend l'essentiel des concepts définis par Bunge ; à savoir, *chose*, *propriété*, *attribut*, *classe*, *catégorie* et *état*. Nous présentons, ci-dessous, un bref panorama des concepts principaux dans *BWW*; *chose (thing) construct*, *propriétés* et *attributs*, *état (state), catégorie (kind)* et *catégorie naturelle (natural kind)* et *loi (law).*

III.2.1. Chose et Construct

Dans l'ontologie de Bunge, il existe deux types d'objets: des *choses concrètes* ou plus simplement *choses*, et des *objets conceptuels* ou *constructs*. Bunge décrit le monde réel comme étant composé de *choses*; les *constructs* étant uniquement des conceptions propres au

cerveau humain. Tous les objets (concrets ou conceptuels) ont des propriétés ; si ces objets sont conceptuels alors les propriétés sont appelées des *attributs*, si par contre les objets sont des *choses* alors leurs propriétés sont dites *propriétés substantielles* (ou simplement *propriétés*). Selon Bunge, une *propriété substantielle* est une caractéristique que les *choses* possèdent même quand on l'ignore. Un *attribut* quant à lui est une caractéristique que l'on attache à une entité conceptuelle. Toujours selon Bunge, un attribut peut être une représentation d'une propriété substantielle. L'ensemble des *propriétés* et *attributs* d'un objet permettent de l'identifier de manière distincte et unique.

III.2.2. Classe et catégorie

Une *classe* est une collection de *choses* qui partagent en comment une propriété donnée. Une *catégorie* (*Kind*) est définie comme étant une collection de *choses* ayant en commun un certain ensemble de propriétés. Toutes les choses possédant cet ensemble de propriétés font partie de la même catégorie.

III.2.3. Propriétés et Attributs et états

Toujours selon Bunge, toute propriété est associée à un objet, certaines propriétés telles que le poids, la taille ou l'âge d'une personne, par exemple, sont dites propriétés inhérentes ou *intrinsèques*. D'autres propriétés sont relatives à des couples ou plus généralement des n-uplets de *choses*, elles sont dites *mutuelles*.

Les conceptualisations conçues par le cerveau humain pour représenter les *choses* sont des *constructs* (objets conceptuels), les *attributs* sont donc des caractéristiques associées aux modèles des choses selon les perceptions et les besoins humains. Différents modèles peuvent être utilisées pour représenter une seule et même chose, il est donc possible d'associer différents ensembles d'attributs à une même chose selon le modèle étudié. En effet la conception humaine des propriétés des choses se traduit en termes d'attributs de leurs modèles conceptuels, les propriétés sont donc perçues par le cerveau humain uniquement comme attributs. Les attributs d'un schéma fonctionnel d'une chose sont appelés *variables d'état,* elles modélisent les valeurs permettant d'identifier et caractériser les différents états associés à cette chose, au long de son cycle de vie. Dans l'ontologie de Bunge, toute chose – à un instant donné – est dans un état précis. "*Tout modèle théorique ou conceptuel d'une chose*

tente de représenter les états possibles (ou valides) ou les changements d'états possibles (ou valides) pour cette chose"(Bunge 1977).

III.2.4. Propriétés émergentes et propriétés héritées

D'après Bunge, une chose peut être composée d'autres choses; une hypothèse fondamentale dans l'ontologie de Bunge dit que l'ensemble des propriétés d'une chose, n'est pas égal à l'ensemble des propriétés de tous ses composants réunis. En effet, une chose composite doit avoir au moins une propriété émergente qui lui est propre et non provenant d'autres propriétés associées aux sous-parties. Ainsi, les propriétés des choses composites sont de deux types: *propriétés héritées* appartenant aussi à des sous-parties de la chose, et des *propriétés émergente*s caractérisant la chose composite comme une entité à part entière.

Nous synthétisons, ci-dessous, l'ensemble des concepts ainsi que les axiomes spécifiés par BWW.

Chose est un concept.
    Une ***Chose*** peut-être composée par plus d'une ***Chose.***
    Et une ***Chose*** peut composer plus d'une ***Chose.***
Propriété est un concept.
    Chaque ***Chose*** possède une ou plusieurs ***Propriétés***.
    Chaque ***Propriété*** décrit une ou plusieurs ***Choses***.
Attribut est un concept.
    Chaque ***Attribut*** modélise une ou plusieurs ***Propriétés***.
    Il est possible qu'une ***Propriété*** soit représentée par plus d'un ***Attribut***.
Classe est un concept.
Catégorie est un concept.
    Chaque ***Catégorie*** est dérivée d'une ou plusieurs ***Classes***.
    Chaque ***Classe*** peut être utilisée dans une ou plusieurs ***Catégories***.

L'ontologie BWW a souvent été critiquée pour son manque de formalisme et le manque de clarté des définitions. Ainsi, beaucoup de travaux ont tenté d'apporter le formalisme manquant à BWW. Rosemann et Green ont formalisé un meta-modèle définissant les concepts contenus dans l'ontologie BWW ainsi que les relations entre ces concepts. La figure 22 représente une formalisation en NIAM du meta-modèle de BWW inspiré des travaux de Rosemann et Green (Rosemann and Green 2002).

D'autres travaux récents continuent d'utiliser BWW comme ontologie descriptive pour la classification et la définition des éléments de leurs domaines (Opdahl and Henderson-Sellers 2002 ; Fettke and Loos 2003). Ainsi, les travaux, toujours en cours, du réseau d'excellence européen INTEROP utilisent BWW en guise de meta-meta-modèle pour la construction d'un langage de modélisation Unifié pour la modélisation de l'entreprise (Berio et al. 2005)

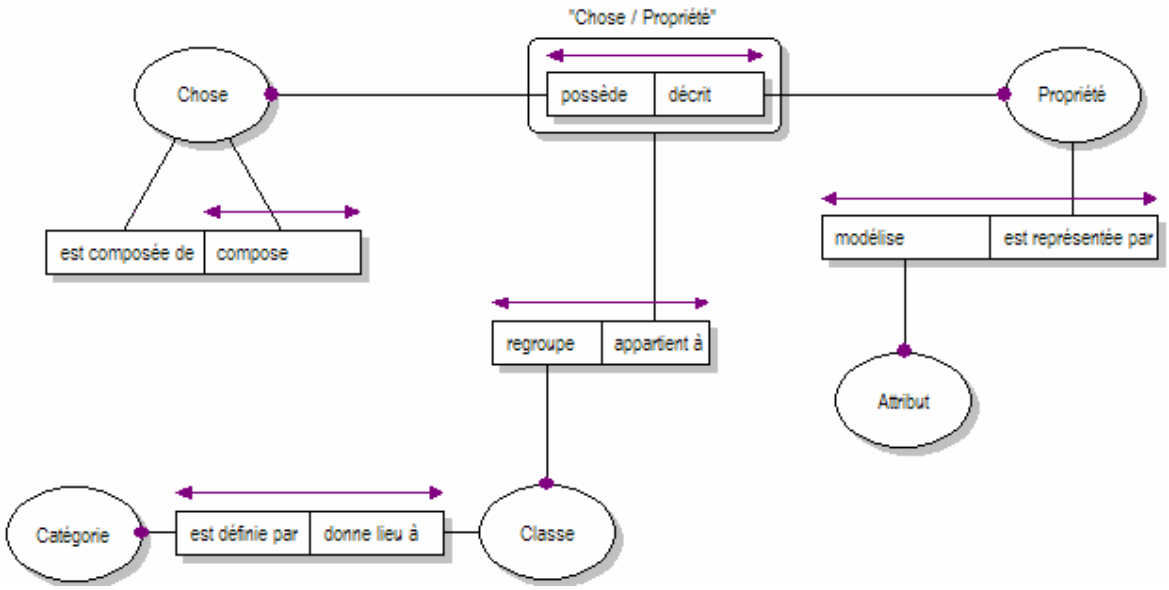


Figure 22. Formalisation en NIAM du meta-modèle de BWW.

Nous combinons ainsi donc la vision holonique du produit et les concepts induits par l'approche BWW pour formaliser chacun des concepts et aspects relatifs à la modélisation du produit holonique.

III.3. Concepts et aspects liés à l'holon produit

Pour construire le meta-modèle décrivant l'holon produit, intéressons nous tout d'abord aux aspects que doit décrire un holon et aux concepts relatifs à cette définition de l'holon.

Pour illustrer les propos que nous aborderons dans cette section, nous nous basons sur des exemples concrets issus de l'Atelier Inter-établissements de Productique Lorrain (AIPL), le centre de ressource régional du réseau national AIP-PRIMECA[12] qui regroupe des pôles de compétences pluridisciplinaires autour de technologies innovantes. Le pôle AIP Lorrain se justifie comme support expérimental de formation approfondie dans le domaine de la

productique et de la conception intégrée en mécanique. Dans nos exemples, nous utiliserons les produits « pilotes » du site de l'AIPL situé sur le campus scientifique de la Faculté des Sciences et Techniques de l'Université Henri Poincaré Nancy I. Pour des raisons pédagogiques, les produits fabriqués sont relativement simples.

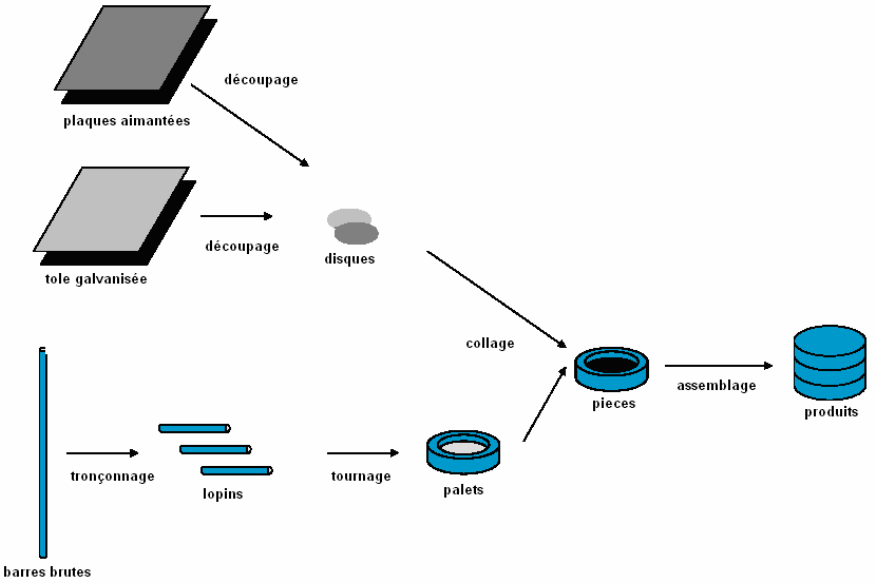


Figure 23. Schéma simplifié du procédé de fabrication à l'AIPL.

Comme on peut le voir sur la figure 23, le procédé de fabrication des produits est composé de plusieurs étapes :

- Le tronçonnage des barres brutes en lopins,
- Le tournage des palets dans les lopins,
- Le découpage des disques dans les plaques de tôle aimanté ou acier galvanisé,
- Le collage des disques dans sur les palets pour obtenir des pièces,
- Et finalement l'assemblage des pièces, pour obtenir des produits finis.

Lors du procédé de fabrication différents produits intermédiaires sont réalisés:

- des lopins (barres d'aluminium d'environ 1 m de long),
- des disques de tôle aimantée ou d'acier galvanisé,
- des palets d'aluminium,

[12] *http://www.aip-primeca.net/*

- des pièces de différents types, chaque type étant identifié par un numéro (*cf.* Figure 24).

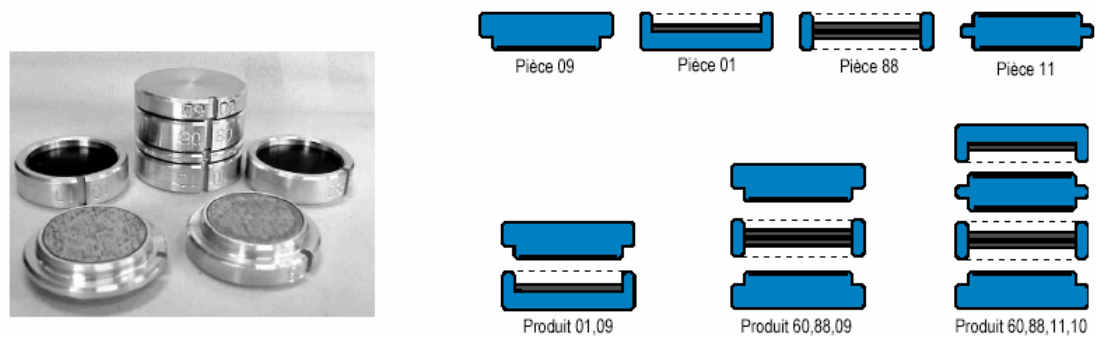


Figure 24. Produits et pièces de l'AIPL.

III.3.1. La composition structurelle

Les produits des entreprises de production sont souvent fabriqués à partir de matières premières ou par composition d'autres produits intermédiaires. L'arborescence dont la racine est un produit fini, les nœuds des produits intermédiaires ou des éléments de matière première, et dont les arcs représentent la relation de composition entre les nœuds est appelée la nomenclature ou structure industrielle du produit situé à la racine (*cf.* Figure 25). La nomenclature d'un produit est utilisée pour identifier les étapes de productions et les gammes associées, établir un plan directeur pour la production ou lancer et suivre les ordres de fabrication. Pour exprimer cette relation de composition, un holon doit pouvoir être lui-même composé d'autres holons (*cf.* Figure 25). Cette relation est appelée un « lien de composition », ou également « lien composé-composant » (Maurino 1995).Nous distinguons, ainsi, deux types d'holons : *(i)* les holons simples et *(ii)* les holons complexes :

- Un holon élémentaire est l'agrégation d'une partie physique unique et de la partie informationnelle lui correspondant.
- Un holon complexe quant à lui est le résultat d'une composition ou une décomposition d'un ou plusieurs holons (élémentaires ou complexes) ; un holon complexe est obtenu suite à un processus qui transforme un ou plusieurs holons en un nouvel holon. L'holon est ainsi dit complexe parce qu'il fait référence non seulement à une partie physique et une partie informationnelle qui lui sont propres mais aussi à l'ensemble des holons qui entrent directement dans sa fabrication sous forme de liens de composition.

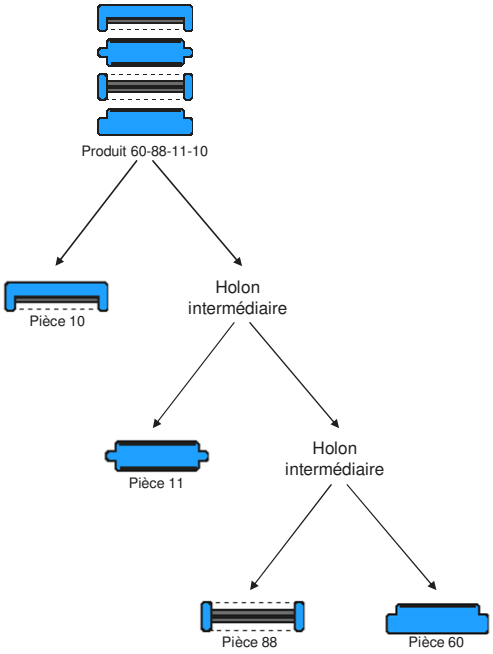


Figure 25. Exemple de nomenclature lié au produit AIPL de type 60-88-11-10.

Dans certains cas très particuliers, un holon complexe peut ne pas avoir de partie physique, par exemple lors de la composition de plusieurs holons pour en obtenir un nouveau, le nouvel holon peut posséder de l'information qui lui est propre mais sa partie physique n'étant que la composition des parties physiques des autres holons, il fait référence à l'ensemble des holons préexistants qui rentrent dans sa fabrication, plus une partie informationnelle décrivant les informations propres à cet holon.

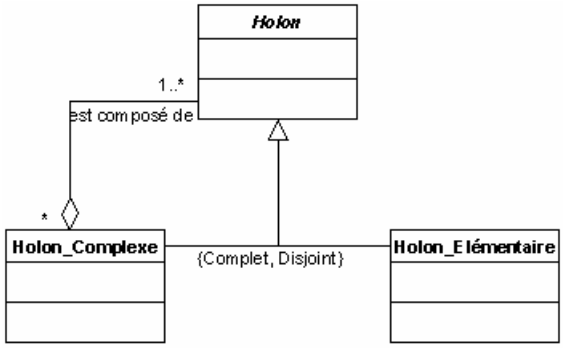


Figure 26. Modèle de composition des holons.

La figure 26 illustre la composition de l'holon telle que nous l'avons présentée pour la première fois dans (Baïna et al. 2005). Certaines contraintes que nous nous les détaillerons par la suite doivent cependant être exprimées pour assurer la cohérence et la sémantique du modèle.

III.3.2. La description des caractéristiques

La partie informationnelle de l'holon décrit l'ensemble des caractéristiques du produit, ces dernières peuvent être classées en deux catégories (*cf.* Figure 27):

- Des caractéristiques décrivant des propriétés intrinsèques à l'objet physique lui-même. Ces caractéristiques sont substantielles à la partie physique même de l'holon (ex : poids, couleur, forme, matériau…)
- Des caractéristiques représentant des informations attachées au produit durant les différentes étapes de modélisation ; chacune de ces informations étant relative à un certain domaine de modélisation ou destinée à une certaine utilisation spécifique (ex : identifiant, durée de vie moyenne, date de fabrication, etc.)

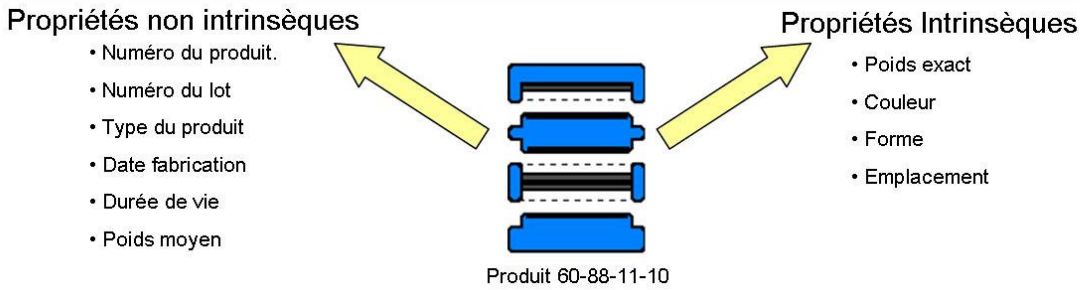


Figure 27. Exemple des différents types d'information autour du produit.

Pour distinguer entre ces deux catégories nous appellerons les informations exprimant des caractéristiques de la première catégorie des *Attributs (*propriétés intrinsèques selon l'ontologie *BWW*), et nous appellerons les informations décrivant des caractéristiques de la deuxième catégorie des *Propriétés,* cette définition couvre plusieurs types de propriétés définis par *BWW ; (i)* des propriétés mutuelles, *(ii)* émergentes ou *(iii)* héréditaires.

(i) Les propriétés mutuelles sont des propriétés communes à un ou plusieurs objets.

Selon BWW, l'ensemble de propriétés d'une chose inclut l'union des ensembles de propriétés des composants de cette chose. Les propriétés des choses composées sont classées en deux types :

(ii) *des propriétés émergentes.* Une chose composée doit avoir au moins une propriété, dite émergente, qui la caractérise en tant qu'entité à part entière.

(iii) *des propriétés héréditaires.* Une chose composée peut avoir des propriétés provenant de ses composants.

III.3.3. L'état d'un holon

Lors de sa fabrication, un produit passe par plusieurs *états* principaux décrivant la totalité de son historique. L'enregistrement de l'ensemble de ces états permet d'assurer la traçabilité des produits lors de leur cycle de vie (Terzi 2005). L'*état* d'un holon est définit par un ensemble de couples (attribut, valeur) et (propriété, valeur). Chaque passage d'un holon au travers d'un processus implique un changement d'état.

L'exécution d'une transformation sur un holon, que ce soit une transformation physique ou un traitement de la partie informationnelle implique des changements au niveau des *attributs* ou des *propriétés* ; en cas de transformation physique suite à un processus physique, le changement concerne un ou plusieurs *attributs (*physique*)* et en cas de transformation informationnelle suite à un traitement informationnel des données du produit, c'est une *propriété* qui s'en trouve altérée. Un changement d'un attribut ou d'une propriété peut être de différents types : attribution d'une valeur ou mise à jour d'une valeur. En cas de modification d'un attribut, cela doit correspondre à une transformation physique effective au niveau de l'objet et ce pour maintenir la cohérence entre les informations faisant partie de l'holon et l'objet physique correspondant.

Supposons que la dernière étape lors de la fabrication d'un produit dans l'entreprise, nécessite une opération de peinture. L'holon est décrit par un ensemble de propriétés et attributs dont un attribut relatif à la couleur du produit.

Un produit est donc dit « *fini* » s'il a été peint, c'est donc une observation qui devra tenir compte de la valeur de l'attribut « *couleur* », une fois que cet attribut aura acquis une valeur, l'objet sera identifié comme étant *fini*. « *fini* » est donc un état qui va dépendre, dans cet exemple, d'une valeur de l'attribut « *couleur* ».

III.3.4. Les flux d'holon

Dans notre contexte, un flux représente un échange d'information, de matière ou des deux en même temps entre différents processus, tâches ou activités de l'environnement de production.

Les holons, les parties physiques et les parties informationnelles, peuvent ainsi être véhiculés à travers différents types de flux. Le type d'un flux étant contraint par la nature de ce qu'il contient, nous identifions trois types de flux:

- des *flux physiques* contenant uniquement de la matière (parties physiques)
- des *flux informationnels* transportant des informations (parties informationnelles).
- des *flux d'holons* contenant à la fois information et matière sous formes d'holons.

Outre le typage des flux par leur contenu, les flux échangés peuvent aussi être classifiés en tenant compte de leur fonction et leur interaction avec le processus ou le système récepteur (stimuli, données, énergies ou autres…); ainsi, un flux reçu peut être perçu comme un flux de Devoir-Faire (DF), un flux de Pouvoir-Faire (PF), un flux de Savoir-Faire (SF) ou un flux de Vouloir-Faire (VF). Ces différents types de flux représentent ce que l'on appelle communément des modalités (Mayer 1995). Ainsi, par leurs interactions avec les processus de fabrication, les produits intelligents représentés par des holons devraient définir le type de flux auquel ils appartiennent (DF, PF, SF ou VF). Cependant, dans le cadre de cette thèse, nous nous intéressons surtout à la représentation du produit et non son comportement et ses interactions avec son environnement. Nous nous positionnons ainsi dans le cadre de flux informationnels ou flux physiques ou encore de flux holoniques c'est-à-dire des flux contenant à la fois des entités physiques et des entités informationnelles.

La figure 28 représente le modèle décrivant les différents types de flux identifiés, cette classification n'est pas exhaustive mais elle représente cependant, l'ensemble des flux auxquels nous nous intéressons dans le cadre de la considération du concept d'holon. A noter qu'un élément peut être véhiculé via plusieurs flux différents, lui permettant ainsi de passer de processus en processus tout au long de sa fabrication.

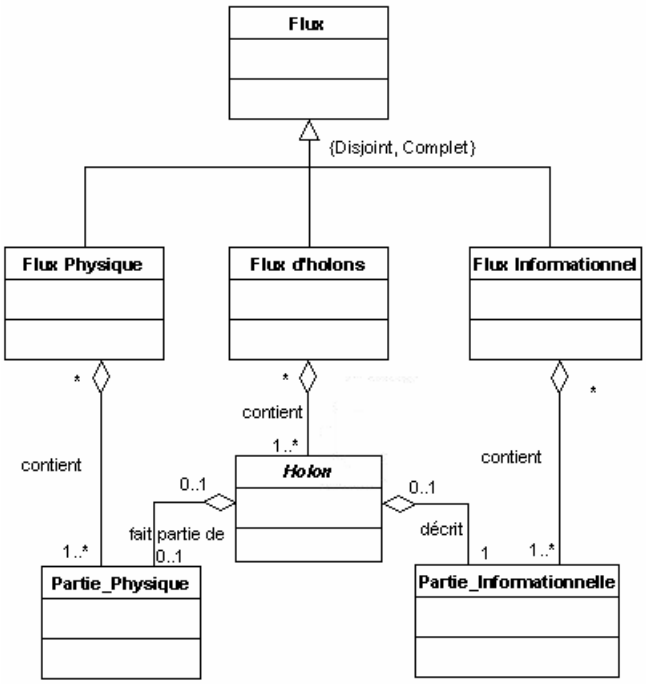


Figure 28. Formalisation des différents types de flux et leur contenu.(Baïna et al. 2005)

Dans ce qui suit, nous rassemblons les différents aspects de l'holon décrit dans cette section pour construire le meta-modèle holonique que nous proposons pour la représentation des produits et leur information dans un environnement de production.

III.4. Un meta-modèle pour l'holon-produit

Le meta-modèle proposé dans cette section est obtenu suite à la factorisation et la synthèse de l'ensemble des éléments explicités dans la section précédente, le résultat est un meta-modèle générique permettant de décrire le produit holonique tel que nous l'avons définis. L'instanciation de ce modèle donnera lieu à des modèles spécifiques décrivant les holons étudiés. Nous appellerons ainsi notre modèle « le meta-modèle holonique ».

Nous avons regroupé l'ensemble des concepts relatifs à l'utilisation de l'holon comme unité identifiable pour la désignation du produit dans son aspect physique et son aspect informationnel. La figure 29 formalise l'ensemble des classes décrivant les notions liées à l'holon ainsi que l'ensemble des relations entre elles.

Ce qui suit est une brève description des classes de ce meta-modèle (Baïna et al. 2006c). La Classe *Holon* définit la structure de base d'un holon, qu'il soit élémentaire ou complexe. Un *Holon Complexe* est, contrairement à un *holon élémentaire*, un holon ayant subit au moins un

procédé de composition ou décomposition lors de sa fabrication. Une *Partie physique* est une référence à la partie réelle du produit associée à un *holon*. Chaque *partie physique* est décrite par des *attributs* qui peuvent avoir des valeurs (*valeur d'attribut*). Une *partie informationnelle* regroupe l'ensemble des propriétés relatives au produit décrit par l'holon en question. Chaque *propriété* peut avoir des valeurs (*valeur de propriété*). Elle peut aussi, en cas d'holon complexe, agréger des attributs relatifs aux holons qui composent cet holon. La classe é*tat* décrit l'ensemble des états par lesquels passe un *holon*. L'état d'un holon est définit par un ensemble de couples (attribut, valeur) et/ou (propriété, valeur).

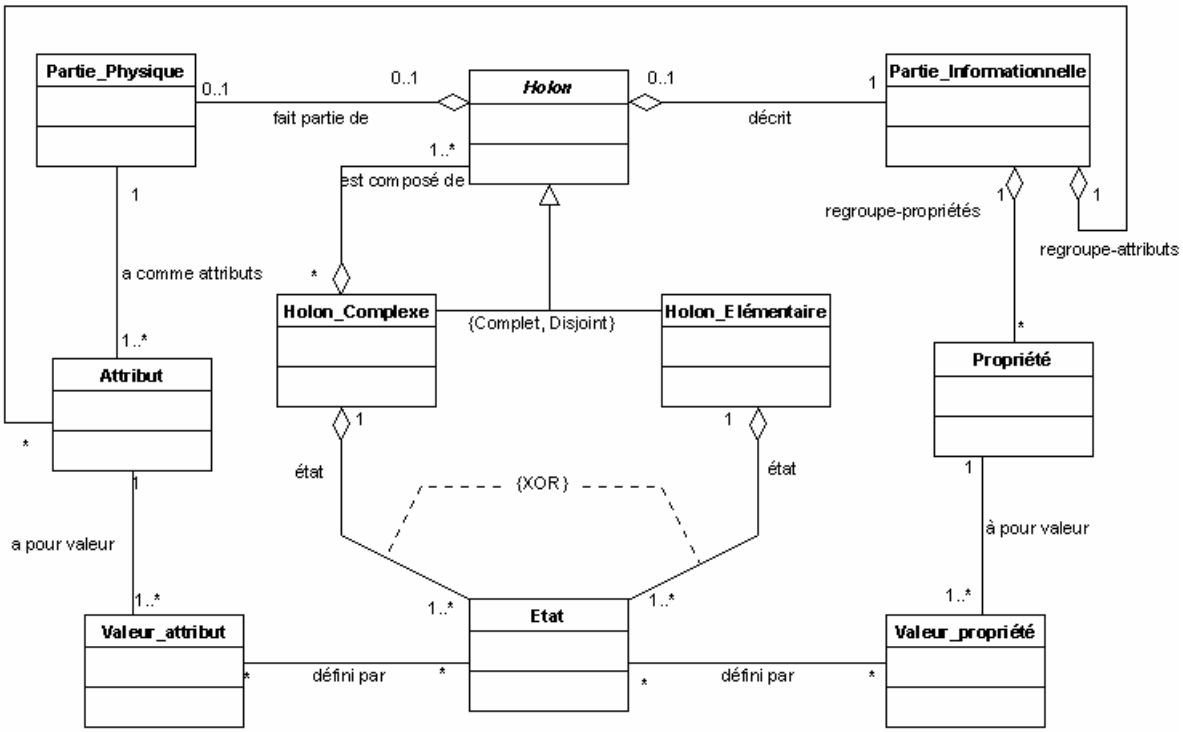


Figure 29. Meta-modèle holonique pour la représentation du produit.

Afin d'instaurer des règles sémantiques lors de l'instanciation de modèles valides selon le meta-modèle présenté ci-dessus, nous définissons dans l'annexe A un ensemble de contraintes que nous formalisons en OCL (Warmer and Kleppe 1999). Cependant, ces contraintes ne suffisent pas à elles seules pour assurer la cohérence entre les données exprimées dans les modèles holoniques et la réalité des produits représentés. En effet, si ces contraintes assurent la cohérence entre les différentes données des modèles en régissant les relations entre ses données, il reste néanmoins évident que sans des règles à respecter lors de la mise à jour des informations et de la transformation de matière, la cohérence des modèles holoniques et de la

réalité physique des objets ne sera pas maintenue. Dans la section suivante, nous montrons les mécanismes utilisés pour maintenir la cohérence entre les informations exprimées dans les modèles instanciés et la réalité des objets physiques.

III.5. L'holon dans son environnement

Si le concept d'holon permet une représentation des produits relativement proche de leur réalité physique, il nous faut néanmoins définir des mécanismes de mise à jour et de transformation physique adaptés pour maintenir la cohérence entre la réalité physique des objets et la représentation de cette réalité sous forme de modèles holoniques. En effet, le concept de produit holonique dans l'environnement de production nécessite la définition d'un nouveau type de processus, capables de maintenir cette cohérence grâce à certaines règles. Ainsi, le fait qu'un holon représente la fois la vue physique et la vue information d'un produit implique que les nouveaux processus dit « *holoniques* » eux aussi doivent pouvoir traiter simultanément, autant que possible, et la partie informationnelle, et la partie physique afin de conserver la cohérence des deux visions.

Toujours dans le cadre de la production AIPL, supposons que la fabrication d'un produit nécessite une opération de peinture. Etant donné que la peinture nécessite une modification de la partie physique du produit, le processus physique effectuant la peinture est donc naturellement impliqué. D'autre part, il faut aussi que l'information relative au produit soit mise à jour pour notifier le fait que le produit été peint.

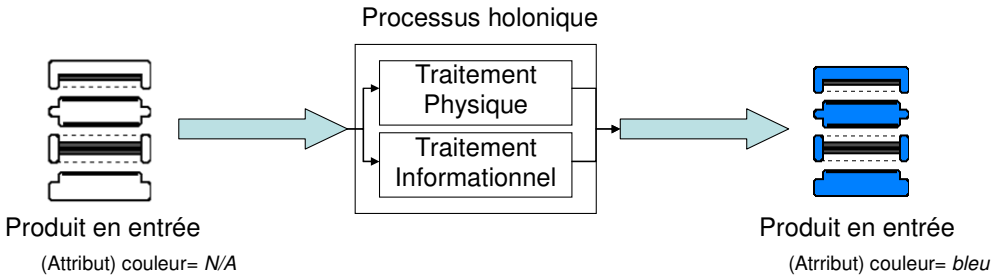


Figure 30. Exemple de processus holonique.

La partie « Traitement Physique » du processus réalise la mise en peinture proprement dite du produit physique. La partie « Traitement Informationnel » s'occupe quand à elle de mettre à jour et renseigner correctement l'information relative à la réalisation du processus de peinture.

Pour que la cohérence entre physique et information soit conservée, nous devons assurer certaines propriétés lors de l'exécution d'un processus holonique (traitement informationnel/traitement physique). Ces propriétés sont inspirées du modèle classique des transactions, qui a fait ses preuves dans le domaine des bases de données.

Nous considérons le processus holonique comme étant une transaction déclenchée par l'introduction de l'holon-produit en entrée, et se terminant par la récupération du même holon-produit en sortie. Le même raisonnement est appliqué lors de l'exécution de processus de transformation d'un élément en plusieurs éléments (1→n) ou l'inverse (n→1). L'exécution des transactions classiques respecte un ensemble de quatre propriétés, nommées les propriétés ACID : Atomicité, Cohérence, Isolation, Durabilité (*cf.* Annexe B). Aussi, nous adaptons les définitions des propriétés ACID des transactions à notre contexte relatif au cadre d'exécution des processus holonique pour la production.

Pour garantir l'accès concurrent aux données et la cohérence des données, le modèle classique utilise des techniques de verrouillage des données. Ainsi, une transaction verrouille les données qu'elle veut modifier. Les autres transactions peuvent les consulter, mais ne pourront les modifier qu'après que la transaction qui les détient ne les déverrouille. Dans le contexte des processus holonique dans le domaine de la production, ce verrouillage des données en modification est plus simple à appliquer. En effet, étant donné que le produit à une existence physique, le produit joue le rôle d'un jeton circulant entre les différents processus. Ainsi, seul le processus détenant le jeton (produit) peut modifier les données relatives à ce produit.

Essayons maintenant d'adapter les propriétés ACID pour leur application au niveau des processus de production que nous appellerons processus holoniques :

- *Atomicité* **–** Il n'est pas toujours possible de revenir en arrière une fois qu'un processus physique a été effectué (par exemple : un perçage). En effet, certaines opérations d'ordre physique ne peuvent être défaites. Nous définirons donc une notion d'atomicité pour laquelle il n'y a pas de retour en arrière possible. En cas d'échec, l'holon (et le produit physique et sa partie informationnelle) est mis au rebut. Nous devons tout de même garder trace de cette mise au rebut qui correspond ainsi à un nouvel état.

- *Cohérence* – Un processus holonique s'exécutant doit préserver la cohérence de la partie informationnelle du produit avec sa réalité physique ; après la terminaison réussie d'un processus, la partie informationnelle doit tenir compte des transformations physiques que vient de subir le produit physique.
- *Isolation* **-** Chaque processus doit pouvoir s'exécuter en parallèle avec d'autres processus dans le même environnement: L'exécution en parallèle d'un certains nombre de processus, doit avoir le même effet que leur exécution en série. Ceci nécessite deux axiomes principaux :
  - Durant l'exécution d'un processus, aucun état intermédiaire des données ne doit être accessible ou exposé aux autres processus.
  - Deux processus concourants ne peuvent modifier simultanément les mêmes informations ; ceci est assuré par le fait qu'un processus ne peut modifier que les informations relatives au produit qu'il détient, l'unicité du produit assure l'unicité du jeton permettant aux processus de modifier les informations.
- *Durabilité* - Les effets d'un processus sur les holons sont persistants à travers le temps. Ils ne s'estompent pas spontanément.

Les propriétés exprimées ci-dessus, permettent de maintenir la cohérence de la représentation informationnelle avec la réalité physique des produits à travers leurs traitements dans les processus « holoniques ». La mise en œuvre de ces propriétés au niveau des processus d'un système de production répond à notre première problématique relative à l'interopérabilité orientée produit :

- Comment faire en sorte que l'ensemble des représentations soient cohérentes avec la réalité physique du produit ?

En effet, dans un tel système il n'est pas nécessaire d'observer les modifications sur les produits sauf en cas de suspicion d'une défaillance au niveau d'un processus physique puisque chaque traitement physique est indissocié du traitement informationnel lui correspondant ; la partie physique et la partie informationnelle d'un holon sont ainsi maintenues cohérentes tout au long du cycle de production.

Dans le chapitre 4, nous verrons l'apport des mappings et des transformations de modèles pour la mise en cohérence de l'ensemble des représentations du produit par le biais du meta-modèle holonique.

L'application des propriétés ACID des processus holoniques suppose une connaissance préalable, pour chaque processus, des informations impliquées dans son exécution. Les informations utilisées lors de l'exécution d'un processus peuvent l'être de deux façons différentes. En effet, elles peuvent être soient modifiées, soient consultées. Cette indication permet une spécification supplémentaire pour le développement des processus de production. En effet, outre la compatibilité physique nous pouvons dés lors exiger une compatibilité informationnelle autour des processus de production, ce point sera détaillé et discuté dans le chapitre suivant.

Ainsi, les processus mis en œuvre pour la modélisation et l'implémentation d'un système assurant la cohérence physique/information doivent offrir une description détaillée de l'ensemble de l'information auquel ils ont accès et dont ils ont besoin (Baïna et al. 2006b). Cette description est appelée l'interface du processus, elle définit pour chaque objet reçu en entrée les propriétés ou attributs auxquels le processus a accès ; cet accès peut être un accès en lecture, en écriture ou lecture/écriture. Aussi, les données (propriétés ou attributs) accédées en lecture, sont les éléments d'entrées, les données accédées en écriture, sont les éléments de sorties, enfin, les données en lecture/écriture font partie des entrées mais également des sorties (*cf* . Figure 31).

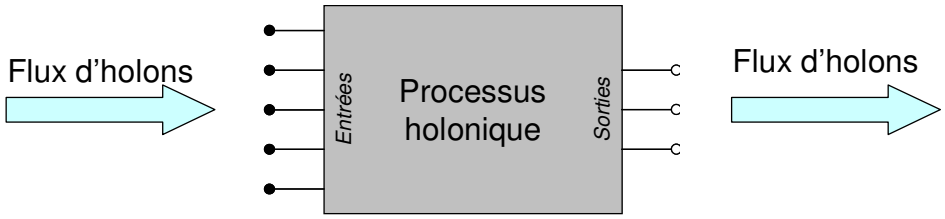


Figure 31. Définition des interfaces nécessaires à l'exécution d'un processus.

Cette interface permet d'établir les interconnexions des processus entre eux. En effet, si le produit joue le rôle de passerelle transportant les informations utilisées par les processus holoniques, il n'en est pas moins vrai que les informations concernant un produit et utilisées par un processus sont nécessairement produites, avant l'exécution de ce dernier, par un autre

processus. Ainsi, un processus ayant besoin d'un ensemble d'information pour son exécution dépend donc d'un processus le précédant (au sens temporel du terme) et qui a produit ces informations. Cette relation de dépendance entre processus peut se traduire en une forme d'interopérabilité entre processus basée sur les interfaces entrées/sorties de ces derniers. Nous définissons ainsi l'interopérabilité entre les processus holoniques de la manière suivante :

Définition 8. *Un processus holonique P est dit interopérable avec l'ensemble des processus d'un système S, si et seulement si tout élément défini par l'interface de P comme entrées, a été préalablement défini dans S comme étant une sortie d'un prédécesseur de P.*

Dans la Définition 8, nous avons utilisé la notion de précédence entre les processus, que nous définissons comme suit:

Définition 9. *La relation de précédence est un ordre partiel dans l'ensemble des processus d'un système. Un processus P1 précède un processus P2 ($P1 <_{Pred} P2$) s'il existe un chemin composé de flux conduisant les objets (holon, information ou autre) depuis P1 vers P2. En cas de systèmes cyclique, la relation de précédence s'applique alors sur les occurrences d'exécution des processus; exemple $P1_i <_{Pred} P2_j$ signifie que la $i^{ème}$ exécution de P1 est effectuée avant la $j^{ème}$ exécution de P2.*

L'interopérabilité définie dans la Définition 8 se situe au niveau syntaxique, puisqu'elle ne concerne que la structure des échanges entre les processus et non la signification de ces échanges. Elle peut être classée au niveau 1 du modèle LCIM présenté dans l'état de l'art (*cf.* Section II.3.3). En effet, cette définition considère que chaque processus du système étudié possède une interface définissant les données en entrée et en sortie de ce processus, permettant ainsi de documenter les besoins du processus en terme de données consommées et produites, ceci correspond aux exigences définies par le niveau 1 du modèle LCIM.

L'utilisation de la modélisation holonique au niveau des processus de production peut ainsi permettre la prise en compte de consignes relatives à l'interopérabilité des processus et de leur interfaçage, dès la phase de modélisation du système.

IV. Conclusion

Dans ce chapitre, nous avons proposé un meta-modèle pour la représentation générique des produits, tout en tenant compte de l'ensemble des informations provenant du cycle de fabrication et leurs interactions avec l'ensemble des systèmes ambiants mis en place dans l'environnement de production. L'instanciation du meta-modèle de référence permet l'identification des différents types d'informations relatives à produit donné. En effet, L'introduction du concept d'holon ou « produit holonique » (information et physique) lors de la phase de modélisation permet d'identifier et classifier l'ensemble des informations concernant le produit. Le résultat de cette modélisation peut, par la suite, être utilisé en tant que base pour la gestion des données collectées par le produit tout au long de son cycle de vie.

Nous avons ainsi adapté le concept d'holon à notre contexte, en le définissant comme étant la combinaison d'une partie physique représentant le produit en tant qu'objet physique, et d'une partie informationnelle représentant l'ensemble des informations relatives au produit. Ces informations sont classifiées en deux catégories : *(i)* des *attributs* décrivant des caractéristiques intrinsèques à l'objet physique qu'est le produit, et *(ii)* des *propriétés* décrivant toute autre type de caractéristique. L'ensemble des valeurs des ses attributs et propriétés définissent les états par lesquels passe un holon. Nous avons aussi montré les mécanismes mis en œuvre pour maintenir la cohérence des représentations du produit avec la réalité physique du produit dans le cadre holonique.

A partir du meta-modèle proposé, dans le chapitre suivant, nous mettons en application l'approche de modélisation basée sur le concept d'holon pour la représentation du produit. Cette mise en application s'appuie sur une implémentation du meta-modèle holonique dans un environnement professionnel pour la modélisation d'entreprise.

En relation avec ce meta-modèle, dans le chapitre 4, nous formalisons les mécanismes de mappings et de transformations de modèles permettant l'utilisation des représentations holoniques du produit comme base pour la mise en cohérence des différentes représentations du produit dans les différents systèmes de l'entreprise.

Chapitre 3 :
L'approche holonique pour la modélisation du produit

I. Introduction

Dans le chapitre précédent, nous avons défini le meta-modèle holonique comme un meta-modèle de référence générique permettant de décrire l'ensemble des caractéristiques du produit (propriétés ou attributs). La représentation holonique du produit peut ainsi être utilisée pour capitaliser l'ensemble des informations relatives à un produit donné. Dans ce chapitre, nous mettons en application le meta-modèle holonique proposé dans le chapitre précédent, en proposant un outil pour la modélisation holonique implémenté dans un environnement professionnel de modélisation d'entreprise. Pour mettre en œuvre la modélisation holonique dans un environnement de modélisation, nous avons choisi l'environnement ***MEGA Suite***[13]. Nous montrons ainsi, comment le meta-modèle holonique à été pris en compte dans l'environnement ***MEGA*** afin de proposer un outil pour l'approche de modélisation holonique.

Par la suite, nous utilisons le cadre de modélisation *ZACHMAN*[14] pour positionner l'approche de modélisation holonique en identifiant les différentes étapes nécessaires. Dans le cadre Zachman chaque étape est définie par l'acteur (ou les acteurs) qui la réalise et selon la question à la quelle elle répond (Quoi ?, Comment ?, Où ?, Qui ?, Quand ?, et Pourquoi ?).

Nous proposons par la suite un cas d'application industriel où la modélisation holonique a été appliquée pour la réalisation d'un système de gestion de traçabilité dans le cadre d'un projet d'informatisation de la traçabilité des produits au sein d'un groupe industriel dans le domaine de la production de farines. L'illustration de cette partie se fait sur la base de quelques exemples extraits des résultats de ce projet.

[13] http://www.mega.com
[14] http://www.zifa.com

II. Implémentation et prototypage

Pour mettre en œuvre notre approche nous avons opté pour l'implémentation de l'ensemble des concepts de modélisation définis par le meta-modèle holonique dans l'environnement MEGA. L'objectif de cette implémentation n'est nullement de prôner l'environnement MEGA comme l'outil parfait, mais surtout de montrer comment notre approche peut s'intégrer dans un outil de modélisation professionnel.

II.1. Introduction à MEGA

MEGA édité par *MEGA International* s'articule autour d'un référentiel qui regroupe trois produits d'analyse et de conception : **MEGA Process**, **MEGA Architecture** et **MEGA Designer**.

- **MEGA Process** pour Cartographier les chaînes de valeur de l'organisation
- **MEGA Architecture** pour Cartographier et urbaniser les systèmes d'information
- **MEGA Designer** pour réaliser la conception détaillée des systèmes et applications informatiques.

Les différentes composantes de la suite MEGA sont organisées autour d'un référentiel unique, fournissant pour l'ensemble des produits MEGA (Process, Designer et Architecture) des services de stockage, de documentation, d'administration et de sécurité.

Pour l'implémentation de l'approche holonique nous utilisons la composante ***MEGA Process***, dont la fonction principale est:

- d'assister les organisateurs dans l'amélioration ou la conception des processus de l'entreprise.
- de permettre aux qualiticiens de décrire les procédures et processus de leur organisation.

Il permet notamment :

- de décrire et évaluer les processus, ainsi que les principaux acteurs de l'entreprise.
- de décrire la séquence détaillée des opérations exécutées au sein des procédures.
- de détailler les besoins en informatisation dans les procédures et ainsi d'établir une cartographie de l'organisation et du système informatique de l'entreprise.

MEGA Process est un atelier muni d'outils pour la modélisation de plusieurs types de diagrammes, dont :

- Des diagrammes UML : diagrammes de cas d'utilisation, d'activités, d'états, de classes, de collaboration, déploiement, de paquetage, séquence.
- Des diagrammes de workflow : selon la notation BPMN.
- Des diagrammes de processus : pour représenter les métiers, acteurs, processus et activités de l'entreprise.

MEGA Process intègre un meta-modèle personnalisable pour s'adapter aux besoins spécifiques des utilisateurs. Ce meta-modèle est compatible avec les principaux standards du marché. De plus, ***MEGA*** intègre plusieurs outils d'import/export ; nécessaires à l'échange d'information avec des applications externes.

II.2. Modélisation holonique dans MEGA

II.2.1. Modification du meta-modèle MEGA

Le meta-modèle de *MEGA* décrit l'ensemble des meta-classes utilisées dans la construction des diagrammes *MEGA*. En effet, chaque meta-classe MEGA représente un type d'objet qu'il est possible de créer dans les différents diagrammes proposés par *MEGA*. Nous décidons d'intégrer les concepts de la modélisation holonique dans le diagramme de processus étant donné qu'il est le plus apte pour la cartographie des processus et les échanges entre les processus d'une entreprise. La figure 32 montre un extrait du meta-modèle du langage de processus mis en œuvre dans l'environnement *MEGA*.

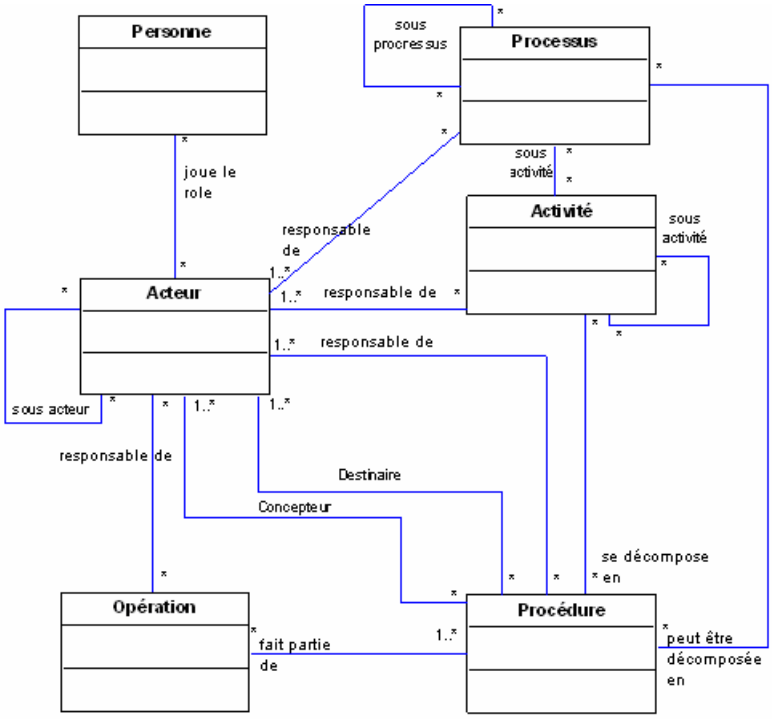


Figure 32. Meta-classes des concepts relatifs à un diagramme de processus MEGA.

Dans ce qui suit une brève description de chacune des meta-classes de la figure 33 :

Un *acteur* représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise :

- Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf. type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial.
- Un acteur externe représente un organisme qui échange des flux avec l'entreprise ; ex : client, fournisseur, administration.

Un *processus* est une chaîne de valeur fournissant un bien ou un service à un client interne ou externe à l'entreprise. Cette chaîne de valeur est décrite par une séquence d'activités de transformation. Elle est mise en œuvre par des procédures.

Une *activité* est une étape d'un processus. Cette étape exprime la contribution d'un métier à la chaîne de valeur du processus.

Une *procédure* décrit la marche à suivre pour mettre en œuvre tout ou partie du processus d'élaboration d'un produit ou un flux. Une procédure est représentée par une succession d'opérations déclenchées par la réception d'un message.

Une *opération* est une étape d'une procédure correspondant à l'intervention d'un acteur de l'organisation dans le cadre d'une des activités de l'entreprise.

L'ensemble de ces concepts peuvent pour communiquer s'échanger des messages. Un *message* représente, donc, un flux circulant à l'intérieur de l'entreprise ou échangé entre l'entreprise et son environnement. C'est généralement un flux d'information comme une commande ou une facture. Par commodité, un flux financier comme le règlement du client, ou un flux de matière comme la livraison d'un produit est également représenté par un message.

La modélisation des processus métiers et des activités de l'entreprise est la première étape de toute étude d'amélioration sérieuse. La modélisation des processus de *MEGA Process* nous permet de représenter les processus ou activités d'une entreprise sous forme d'étapes placées sous la responsabilité d'unités de travail; les étapes s'enchaînent chronologiquement. La démarche de modélisation des processus consiste à créer un ou plusieurs processus initiaux. Ensuite, ce ou ces processus sont décomposés sous formes de sous processus; cette décomposition est représentée sous forme de diagrammes composés d'activités ou de procédures; décrivant aussi les messages échangés, les acteurs impliqués, etc.

L'intégration des concepts de base de notre approche de modélisation holonique dans le référentiel *MEGA* a commencé tout d'abord par la prise en compte de l'échange de flux entre les processus et activités représenté par *MEGA*. Ainsi, pour définir la meta-classe flux, nous nous sommes inspirés du concept de message défini par MEGA. Selon le vocabulaire MEGA, un message peut être informationnel, financier ou flux de matière, le concept de *flux* tel que nous le considérons qu'il soit *flux holonique, informationnel ou de matière* est un sous-type particulier du concept de message introduit par MEGA, dans lequel, le contenu est décrit en terme d'objets clairement identifiés. Par la suite, les autres concepts de la modélisation holonique ont aussi été intégrés autour dans le meta-modèle *MEGA*.

Nous avons donc intégré une partie du meta-modèle holonique dans le référentiel interne de *MEGA*. La partie non prise en compte concerne les états des holons, en effet, nous estimons que les états, étant définis par rapport aux valeurs effectives des attributs et propriétés et non à leur définition propre, doivent être définis à partir de données effectives récoltées, et non à partir de la modélisation.

La figure 33 montre les classes implémentées du meta-modèle holonique telles qu'elles ont été intégrées dans *MEGA*. La relation de toutes les nouvelles meta-classes et de la meta-classe « Diagramme » *MEGA* permet d'exprimer le fait qu'un élément de la nouvelle classe peut être créé dans un diagramme standard de MEGA. Dans cette figure, les cardinalités ne sont pas exprimées car *MEGA* ne vérifie pas les cardinalités du meta-modèle lors de l'instanciation des modèles. Cependant, les cardinalités peuvent être vérifiées grâce à un mécanisme de règles de modélisation que nous expliquerons par là suite.

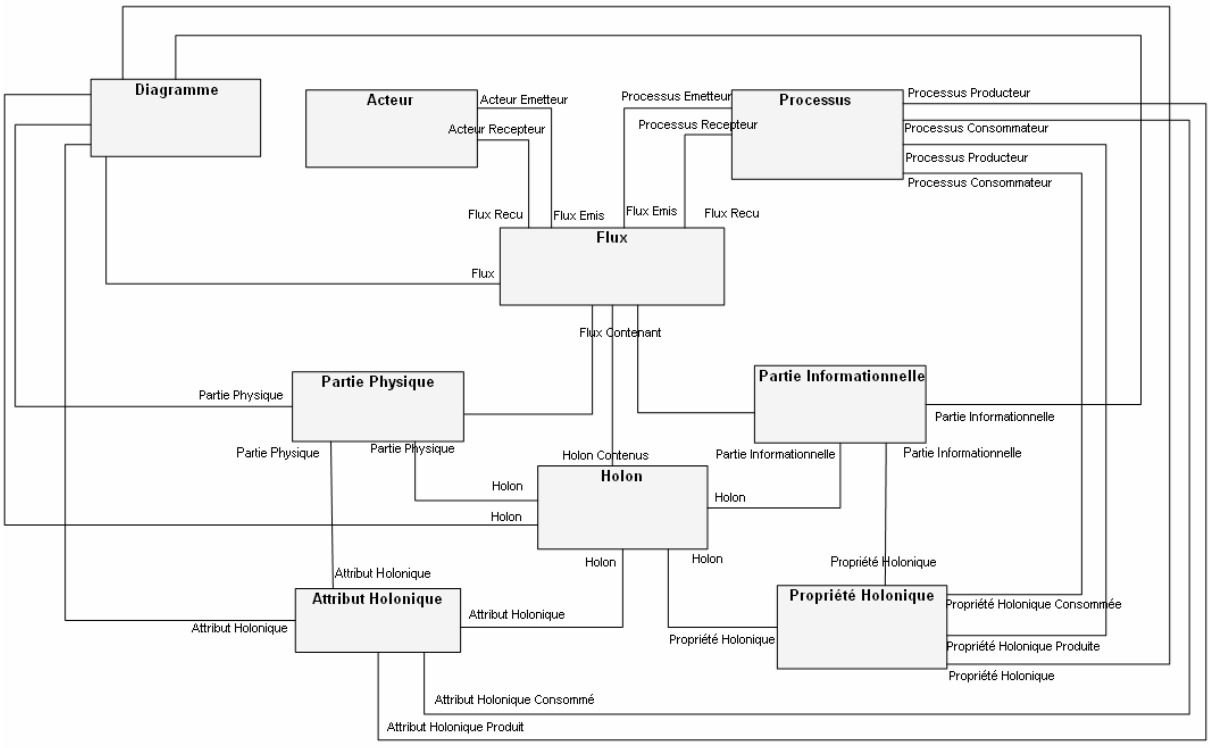


Figure 33. Meta-modèle holonique tel que implémenté dans l'environnement MEGA.

Dans ce meta-modèle, nous avons ajouté deux meta-associations, reliant directement un holon à une propriété ou à un attribut, ces deux meta-associations servent à simplifier les modèles instanciés. En effet, la partie informationnelle d'un holon, par déduction va contenir

l'ensemble des attributs et propriétés de cet holon, pour simplifier les modèles on peut choisir de ne pas représenter la partie informationnelle, et d'en faire abstraction en reliant directement un holon à l'ensemble de ses propriétés et attributs.

II.2.2. Une interface graphique pour les nouveaux concepts

Une fois que les nouvelles meta-classes eu été créées, il a fallu créer une représentation graphique permettant de les afficher sur un Diagramme et également de les différencier les unes des autres.

MEGA propose un ensemble d'outils (ou dialogues) pour la personnalisation de l'ensemble des diagrammes qu'il propose, parmi ceux-là, l'outil appelé « Paramétrage des diagrammes » permet, d'associer des meta-classes et les meta-associations entre elles à un diagramme *MEGA* donné, cette association permet ainsi d'instancier les nouvelles meta-classes au sein du diagramme voulu, dans notre cas le diagramme de processus. L'ensemble des graphiques associés à chacun des concepts que nous utiliserons par la suite dans nos modèles est illustré dans l'***Annexe D***.

Le dialogue « Paramétrage des diagrammes » permet aussi de classer l'ensemble des meta-classes utilisables dans un diagramme en vues, ainsi donc, l'utilisateur peut choisir d'affiche une vue, ou de la cacher.

Pour structurer notre modélisation, nous avons identifié 3 différentes vues :

- La vue « Flux » : permettant d'afficher l'ensemble des flux échangés entre les processus modéliser dans un diagramme de processus.
- La vue « Holons » : permettant d'afficher les holons, parties physiques ou parties informationnelles faisant partie d'un flux et les relations entre ces éléments.
- Et la vue « Propriétés et Attributs » : cette vue regroupe les propriétés et attributs des holons.

Dans la section suivante, nous montrons comment l'approche holonique de modélisation peut s'intégrer dans un cadre de modélisation de plus large envergure. Nous avons ainsi choisi de décortiquer l'approche holonique et l'analyser pour la positionner dans le cadre Zachman pour la modélisation d'entreprise.

III. L'approche holonique selon le cadre de modélisation Zachman

L'objectif de cette section est de montrer comment la modélisation holonique construite autour du concept de l'holon-produit ainsi que l'échange de modèles holoniques utilisant différents mappings de meta-modèles peuvent s'intégrer dans une méthodologie de modélisation de plus large envergure. Pour cela, nous avons choisi le cadre de modélisation Zachman introduit à la fin des années 80, par John Zachman. Zachman a constaté l'existence d'une multitude de diagrammes pour la représentation des connaissances de l'entreprise à différents niveaux sans relations claires entre les différents diagrammes.

III.1. Introduction au cadre Zachman

En 1987, John Zachman écrivit : "Pour préserver l'activité de la désintégration, le concept d'une architecture de systèmes d'information devient de moins en moins une option et de plus en plus une nécessité."(Zachman 1987). A partir de ce constat, naquirent les premières esquisses de l'un des cadres de modélisation d'entreprise les plus répandus, il s'agit du cadre Zachman. En 1992, le cadre fut raffiné et étendu suite aux travaux de (Sowa and Zachman 1992). Le cadre Zachman est une structure logique pour classifier et organiser les représentations des informations d'une entreprise, significatives aussi bien pour le management de l'entreprise que pour le développement des systèmes de l'entreprise. La représentation du cadre Zachman dans sa plus simple forme représente les artefacts de conception que constituent les intersections entre les rôles impliqués dans le processus de conception et les abstractions des produits. Pour Zachman un acteur peut être :

- Quelqu'un qui s'est engagé à faire des affaires dans une industrie particulière ;
- Un homme d'affaires qui dirige l'entreprise ;
- Un analyste qui veut représenter l'activité de l'entreprise sous une forme fonctionnelle ;
- Un concepteur qui applique des technologies spécifiques aux problèmes de l'activité ;
- Le constructeur du système ;

- Le Système lui-même.

Il a donc identifié les différents rôles que peut jouer un acteur dans l'entreprise : le Planificateur, le Propriétaire, le Concepteur, le Constructeur, et le Sous-traitant. Ces rôles représentent les lignes dans la matrice Zachman. Zachman a constaté que chacun des participants (rôle ou acteur) regardait les mêmes catégories d'information mais sous différentes perspectives, il identifie ainsi les différentes catégories d'informations, qui représentent les *colonnes* dans la matrice Zachman:

- « Quoi » : les données ou objets manipulés par une organisation (*What*).
- « Comment » : les processus permettant de manipuler les données et objets (*How*).
- « Où » : les endroits où se passe l'action (les processus, la direction des affaires, etc.). (*Where*)
- « Quand » : événements qui déclenchent des activités économiques. (*When*)
- « Qui » : les gens et les organismes impliqués. (*Who*)
- « Pourquoi » : les motivations qui déterminent le comportement les affaires. (*Why*)

Le cadre Zachman ne spécifie pas de modèles ou même de méthodes pour traiter chacune des cases le composant, il ne recommande pas l'utilisation de tel ou tel style de modélisation pour ces cases. Il est neutre vis à vis des méthodologies puisqu'il ne définit ni les tâches, ni leurs dépendances, ni les ressources impliquées, ni les délivrables et contrôles nécessaires. Zachman est aussi le premier à reconnaître que pour certaines cases de son cadre, il n'existe pas encore de techniques satisfaisantes. Plusieurs travaux, comme par exemple (Kleinberg and Merriman 1996; Rational Rose White Paper 2001), ont montré la compatibilité de l'approche Zachman ou sa complémentarité (Panetto et al. 2006) avec les méthodes de modélisation dans le monde de l'entreprise (UML[15], RUP[16], Idef x[17], etc.)

Dans la section, qui suit, nous tentons d'analyser chacune des facettes de la modélisation holoniques et identifier la case Zachman correspondant, en termes de rôles impliqués et des questions traitées selon la classification Zachman.

---

[15] *http://www.omg.org*
[16] *http://www-306.ibm.com/software/awdtools/rup/*
[17] *Méthode IDEF: http://www.idef.com/*

III.2. La modélisation holonique et Zachman

Tout d'abord nous identifions l'ensemble des étapes de l'approche holonique, par la suite nous tenterons de positionner chacune de ces étapes dans le cadre Zachman et ce en précisant le point de vue ou perspectives qu'elle traite (quoi, comment, quand, qui, quand, pourquoi), ainsi que le niveau d'abstraction auquel elle se situe (contextuel, conceptuel, logique, physique, etc.) chacun de ces niveaux d'abstraction et relatif à un rôle donné (planificateur, propriétaire, concepteur, constructeur, etc.).

- ***Etape 0 :***

  Avant de commencer la modélisation des produits de l'entreprise, étant donné que la modélisation holonique de l'entreprise s'effectue sur les bases d'un diagramme de processus. L'étape préliminaire à toute autre chose dans l'approche holonique consiste préciser la portée de la modélisation. Cette étape a pour objectif de designer les acteurs, les sites de productions ou d'activité à couvrir, pour chacun des sites identifier les processus concernés, et pour chaque processus nommer les objets manipulés (entrées/sorties). Cette étape est purement contextuelle, il n'y a pas de diagramme à établir, le but étant de lister l'ensemble des acteurs, sites, processus ou objets (produits, matières premières ou autres) pouvant intervenir dans les diagrammes réalisés par la suite. Cette étape préliminaire couvre en faite la première case, correspondant au contexte, de chacune des colonnes (de droite à gauche) :

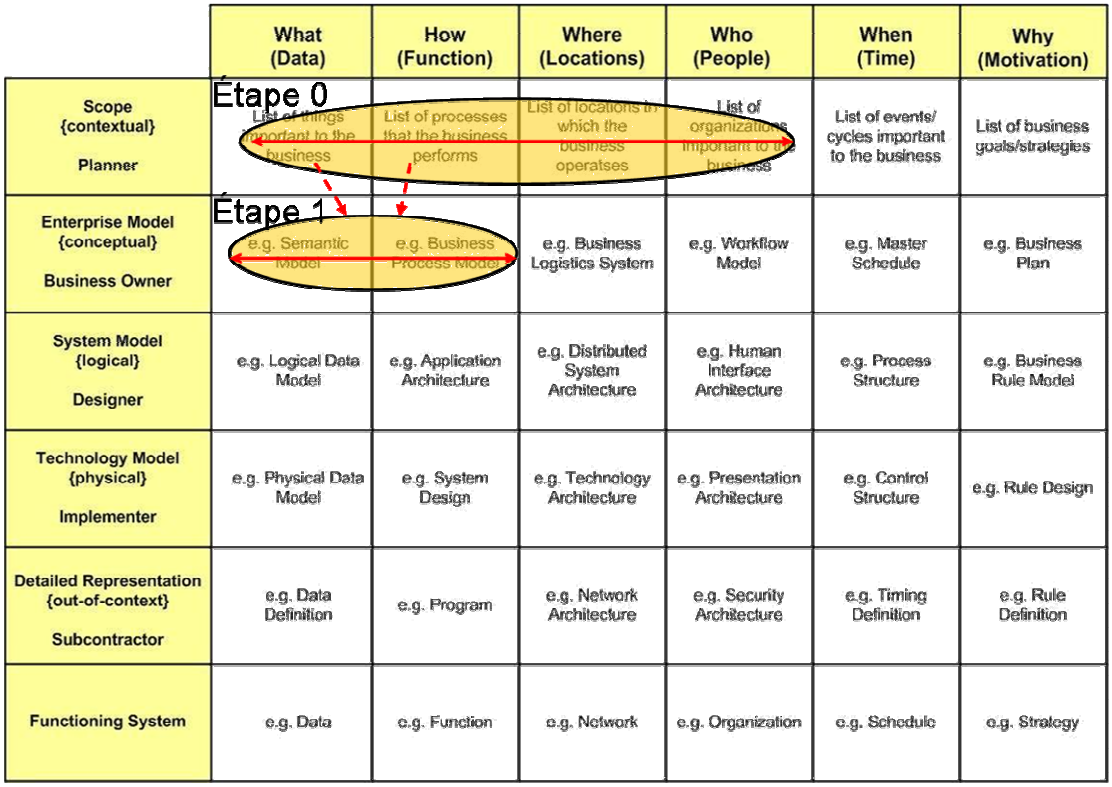
  - Qui (Who), pour les acteurs.
  - Où (Where), pour les sites.
  - Comment (How), pour les processus.
  - Et Quoi (what), pour les objets important pour l'entreprise.

- ***Etape 1 :***

  Dans cette étape, en combinant l'ensemble des éléments récoltés dans l'étape préliminaire (étape 0), les premiers diagrammes holoniques, décrivant la circulation des produits (holons) entre les processus à travers des Flux, peuvent être construits. La construction de ces diagrammes concerne en même temps les deux colonnes : Comment (How) et Quoi (what), elle se situe plus exactement au niveau conceptuel, étant donné que l'objectif et de construire un modèle conceptuel de l'entreprise. Un diagramme holonique décrit aussi les

propriétés et attributs de chaque objet (holon), les interactions entre les processus et les propriétés et attributs des holons en entrées ou sorties peuvent aussi être décrites.

La figure 34 positionne les deux étapes sur la grille Zachman afin de définir les aspects de l'entreprise couverts par l'approche de modélisation holonique.



	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner	List of things important to the business	List of processes that the business performs	List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} Business Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Figure 34. Etapes pour l'utilisation de la modélisation holonique selon le cadre Zachman.

Les cases de la même étape faisant partie de la même ligne sont visitées de préférence de droite vers la gauche. En effet, un déplacement vers la gauche correspond à un affinement de la granularité du domaine étudié (ex : Données ← Processus← Site). De même un déplacement vertical, du haut vers le bas correspond à une baisse de l'abstraction des éléments considérés. En effet, de haut en bas, les cases vont de la plus abstraite à la moins abstraite jusqu'à atteindre le niveau le plus bas relatif à l'implémentation physique. L'approche de modélisation holonique proprement dite, telle qu'elle a été décrite jusqu'à présent, n'adresse que les niveaux contextuel et conceptuel du cadre Zachman, elle ne suggère pas d'implémentation ni de technologies pour la suite de l'exploitation des modèles.

Dans la section suivante, nous proposons une application dans un contexte industriel afin de montrer l'applicabilité de l'approche de holonique.

IV. Mise en application de la modélisation holonique

Dans cette section, nous mettons en œuvres la modélisation holonique au profit d'un cas industriel réel, dans lequel l'objectif principal concerne la construction d'un système de gestion de traçabilité des produits. Le projet est une collaboration avec la branche meunerie d'un grand groupe industriel. L'objectif de cette application est de montrer que l'approche holonique pour la modélisation est applicable dans un cadre industriel proprement.

IV.1. Objectif et Contexte

L'objectif du projet réalisé en partenariat avec le groupe industriel est de concevoir, développer et mettre en place un système de gestion de traçabilité des produits de type « *farine »,* pour le suivi des informations relatives aux commandes des clients depuis leur réception, jusqu'à la livraison des produits chez le client, et ce pour permettre des opérations de rappel en cas d'observation d'incident ou de détérioration des produits lors de la production. Notre participation dans ce projet avait pour objectif, la proposition du modèle de données pour la représentation du produit, en prenant en compte l'ensemble des informations utilisées depuis la réception des commandes jusqu'à la livraison des produits en passant par la préparation des moutures. Les mécanismes et outils pour la classification des incidents et leur relation avec les produits sortants font partie d'une tâche indépendante, dont nous n'assurons pas la réalisation.

En 1987, une définition de la traçabilité a été éditée dans la norme NF X 50-1201. La traçabilité se définit alors de la manière suivante : «La traçabilité du produit est l'aptitude à retrouver l'historique, la localisation ou l'utilisation d'un produit au moyen d'une identification enregistrée». (NF X 50 120 1987).

Une définition de la traçabilité dans la norme ISO 8402 complète la définition de la norme NF X 50 120 ; «La traçabilité est l'aptitude à retrouver l'historique, l'utilisation ou la localisation

d'un article ou d'une activité, ou d'activités semblables, au moyen d'une identification enregistrée». (ISO 8402 1994).

Une entité tracée peut donc être une activité (ou processus), un produit, ou un organisme ou une personne. Lorsqu'il se rapporte à un produit, le terme " traçabilité " peut se référer à l'origine des matériaux et des pièces à l'historique des processus appliqués au produit ou à la distribution et l'emplacement du produit après livraison.

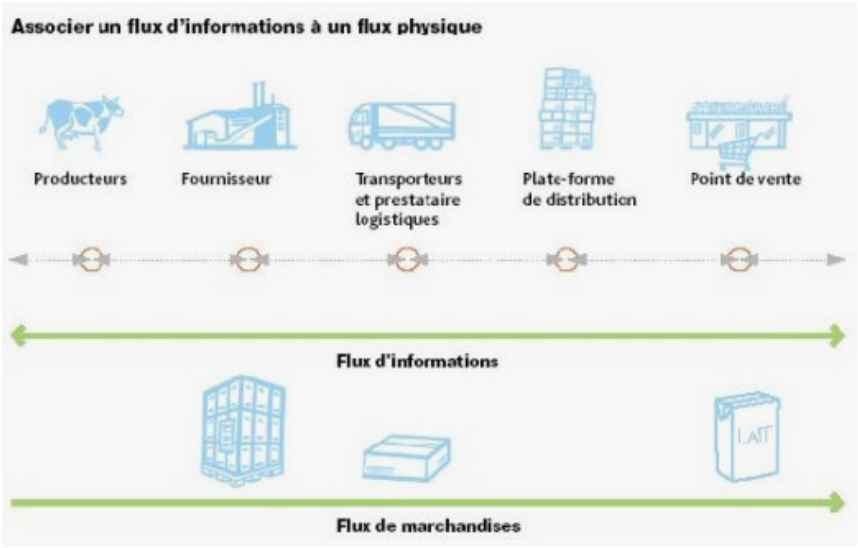


Figure 35. Objectif et principe de la traçabilité.

Du point de vue de l'utilisateur, la traçabilité peut être définie comme étant le fait de suivre des produits qualitativement et quantitativement dans l'espace et dans le temps. Du point de vue de la gestion de l'information, mettre en place un système de traçabilité dans une chaîne d'approvisionnement, c'est associer systématiquement un flux d'informations à un flux physique. L'objectif est de pouvoir retrouver, à l'instant voulu, des données préalablement déterminées relatives à des lots ou regroupements de produits, et ce, à partir d'un ou plusieurs identifiants clés.

Il existe autant d'utilisations différentes pour les systèmes de traçabilité que de définitions de la traçabilité (Viruéga 2005). Parmi tous les usages répertoriés nous citons :

- **L'usage en plan de rappel :** lorsqu'il se rapporte à un produit, le terme peut se référer à l'origine des matériaux et des pièces, l'historique des processus appliqués au produit, la distribution et l'emplacement du produit après livraison.

- **L'usage en recueil de données :** lorsqu'il se rapporte à la collecte de données, il relie les calculs et les données générales tout au long de la boucle de qualité, en remontant parfois aux exigences pour la qualité pour une entité.
- **L'usage en étalonnage** : au sens de l'étalonnage, le terme traçabilité s'applique au raccordement des équipements de mesure aux étalons nationaux ou internationaux, aux étalons primaires ou aux constantes et propriétés physiques de base

IV.2. Modélisation Holonique de l'installation

La traçabilité à mettre en œuvre au sein de l'entreprise correspond à une utilisation en plan de rappel en cas de produits défectueux. Cette utilisation peut donner lieu deux types de fonctionnement du système de traçabilité :

- Après détection d'anomalies dans la production, une recherche de l'ensemble des produits concernés est effectuée, et pour chaque produit livré, le client est informé de la défaillance.
- Le produit est déclaré défectueux en sortie du système de production (par le client ou par le service de qualité) ; il faut alors retrouvé l'ensemble des éléments impliqués dans sa production pour isoler le problème en cause.

Les deux cas correspondent à une traçabilité dite en arrière « *backward traceability* ». En effet l'utilisation des données recueillies s'effectue après la production, on remonte donc le cycle de vie du produit.

IV.2.1. Problématique de la traçabilité dans l'entreprise

Depuis déjà quelques années, le partenaire a mis en place l'identification des nombreux articles, documents, et produits manipulés dans le moulin, les commandes reçues sont identifiées de manière unique, les lots de produits de l'entreprise (palettes de 20-50Kg, Big bags) disposent tous d'une identification selon le type de farine concernée, et la composition de mouture utilisée.

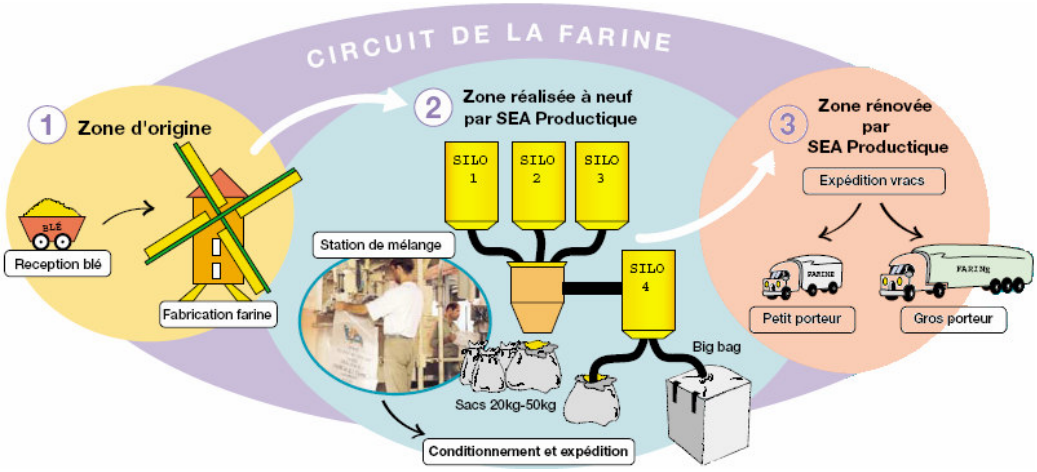


Figure 36. Evolution du processus de fabrication de farine dans le moulin étudié.

Cependant, l'identification à elle seule n'implique pas la traçabilité. En effet, c'est le traitement de l'information spécifique au système de traçabilité qui fait qu'un système de traçabilité est fiable ou pas. Les liaisons et relations existant entre les différents éléments et leurs identifiants forment une caractéristique principale dans un système de traçabilité, permettant ainsi de retrouver à l'historique de l'entité décrivant son traitement, son utilisation ou sa localisation. C'est dans ce contexte, que s'inscrit notre collaboration pour la modélisation des données nécessaires au suivi des produits depuis la réception des commandes, jusqu'à la livraison.

IV.2.2. Méthode de travail

Nous avons commencé par délimiter le domaine étudié. Il a été décidé que dans un premier temps, nous nous focaliserons sur les produits de type 20-50Kg et ce en couvrant les processus de réceptions des commandes, la planification de la production, et l'organisation de la livraison des produits. Le processus concernant l'exécution de la fabrication produit a été mis en dehors du domaine étudié.

Notre objectif est donc de construire les modèles de données nécessaires pour couvrir les informations utiles pour le suivi de la traçabilité des produits 20 et 50kg.

Pour obtenir les modèles de données escomptés, nous appliquons donc la modélisation holonique implémentée dans l'environnement de Modélisation MEGA. L'approche permettra donc de suivre les objets de l'entreprise (produits, matière première, commandes, documents

internes) en modélisant à chaque fois les informations produites, utilisées ou attendues par les différents processus, sans oublier de définir les liens de dépendances entre les objets. De l'ensemble de ces modèles sera déduit, par la suite, le modèle de données complet pour le suivi de la traçabilité. La figure 37 illustre les différents échanges mis en route depuis la réception d'une commande du client jusqu'à la livraison.

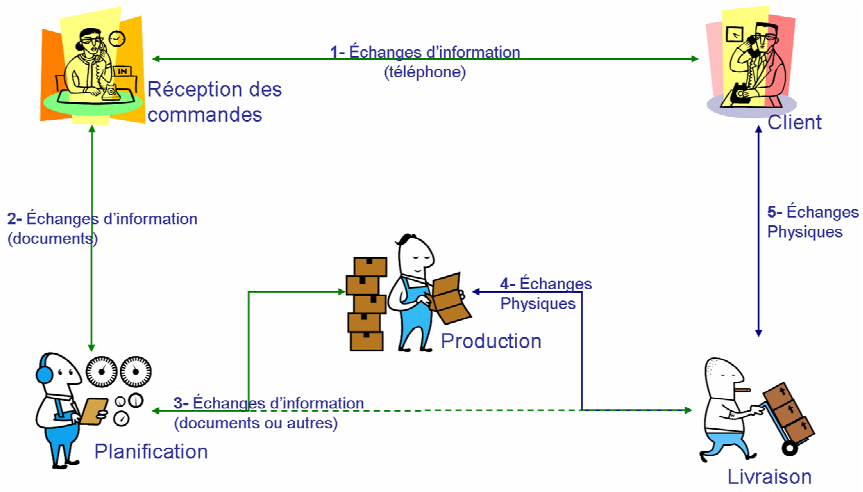


Figure 37. Description de la circulation des flux sur le terrain.

IV.2.3. L'approche appliquée

L'approche holonique a pour premier objectif réalisation d'une cartographie représentant l'ensemble des opérations que subit un produit dans son environnement lors de sa production, ainsi que l'ensemble des relations qui le relient aux autres objets de l'entreprise, produits ou matières premières, et ce pour permettre la réalisation de systèmes d'information décrivant au plus prés la réalité et les informations du produit tout au long de sa fabrication.

Appliquée à un contexte de traçabilité, l'approche holonique permet de générer automatiquement le schéma relationnel permettant de stocker l'ensemble des données relatives au produit, ses relations avec les autres produits, ses relations avec les matières premières utilisées lors de la fabrication et ses relations avec les ressources nécessaires à sa fabrication. Dans ce qui suit, nous présentons l'ensemble des opérations allant de la cartographie jusqu'à la réalisation d'un système cohérent avec la réalité du produit dans le but de réaliser la base de données représentant le noyau dur d'un système d'information pour la gestion de la traçabilité.

- **Phase 1 : Modélisation Holonique**

  Cette phase vise à réaliser la cartographie de l'ensemble des processus ou opérations que subit un produit lors de sa production, sans oublier les relations le reliant aux autres produits ou matières première. Les modèles produits restent cohérents avec la réalité physique observée sur les produits concrets.

- **Phase 2 : Génération du modèle de données dans formalisme UML**

  A partir des modèles générés par la phase 1, on synthétise en UML sous forme de diagrammes de classe (approche orientée objet) l'ensemble des informations relatives à chacune des classes de produits identifiées à la suite de la phase 1. Cette phase permet, de se détacher de la représentation holonique des produits, en structurant l'ensemble des données en utilisant un format standard, UML en l'occurrence.

- **Phase 3 : Génération du schéma relationnel**

  En vue de la réalisation d'une base de données relationnelle, cette phase dans ce sens consiste à générer à partir du résultat de la phase 2 (Diagramme de classe UML) un schéma relationnel restituant la structure conceptuelle définie en UML lors de la phase 2.

- **Phase 4 : Implantation**

  Lors de la phase 4, le schéma relationnel produit par la phase 3 est implanté tel quel dans un système de gestion de base de données (SGBD, Oracle, MYSQL, MS ACCESS, etc.), en vue d'une implantation logicielle. Cette phase permet, de créer l'ensemble des tables physiques qui contiendront les données récoltées en milieu réel.

- **Phase 5 : Peuplement de la base de données**

  Cette phase vise à formater convenablement les données récoltées sur le terrain, pour pouvoir par la suite les saisir par la suite dans la base de données L'objectif de cette phase consiste en l'utilisation de données réelles et non d'un jeu de données fictif et ceci pour un diagnostic correspondant à la réalité.

- **Phase 6 : Diagnostic**

  Le but de cette phase est d'identifier les lacunes et manques de renseignement de certaines informations primordiales pour le bon fonctionnement de la base de données.

Exemple : des clés primaires redondantes, des clés secondaires absente ou mal renseignée, des champs dans un mauvais format.

L'objectif de ce diagnostic étant de corriger et mettre à niveau le système de collecte des données afin de préserver le fonctionnement, et la fiabilité de la base de données obtenue. Cette dernière phase représente le moyen par le biais du quel, un retour constructif sur le système concernant la récolte de données et la gestion des données lors de la production est obtenu, permettant de compléter augmenter, ou repenser les données récoltées ainsi que la manière de les récolter.

La phase 1 correspond à ***l'étape 1*** identifiée précédemment dans le cadre de modélisation Zachman. Les Phases 2 à 4 correspondent aux lignes « Logique » et « Physique » de la colonne « Quoi » du cadre de modélisation Zachman. Quant aux phases 5 et 6, elles correspondent à l'utilisation du systèmes et non à son développement elles peuvent donc pas être représentées dans la cadre Zachman. - Chapitre 3 -

IV.3. Vers un système gestion de traçabilité

Dans cette section, nous montrons comment chacune des phases définies précédemment a été appliquée sur le cas étudié. Nous illustrons chaque phase par des exemples concrets réalisés d'après notre cas d'étude.

- **Les modèles OSSAD :**

Le point de départ de notre modélisation est l'identification de l'ensemble des données échangées entre les différents acteurs entrant en jeu dans le domaine étudié. Pour ce faire, l'entreprise a modélisé le traitement de données dans le domaine étudié en utilisant l'outil OSSAD[18]. La figure 38 montre un exemple de modèle réalisé sous l'outil OSSAD, représentant le modèle de circulation de données entre les différents acteurs de l'entreprise (humains ou machines) lors de la réception d'une commande de production.

[18] *http://www.ossad.org*

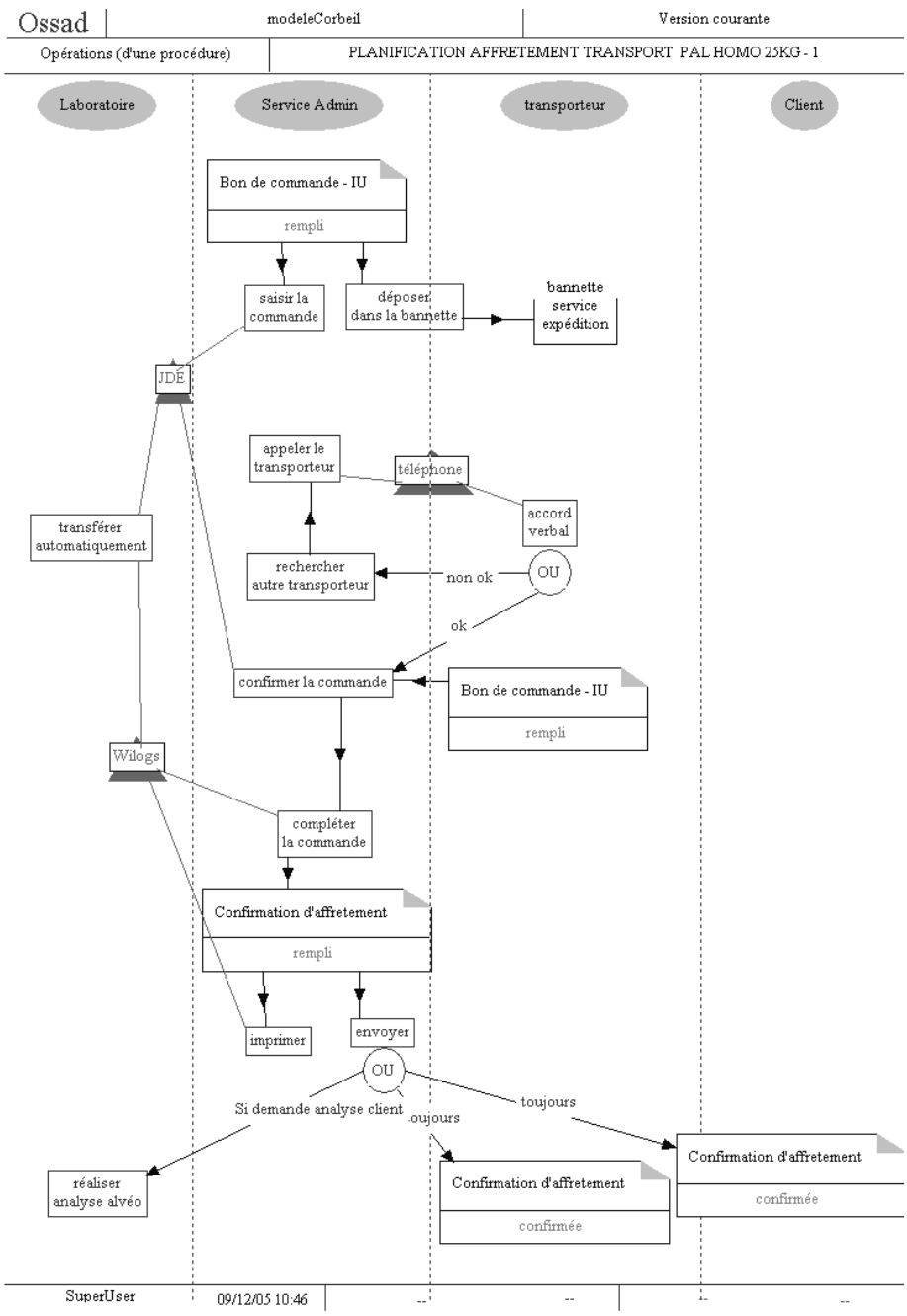


Figure 38. Exemple de schéma de traitement de données produit sous OSSAD.

La difficulté qui s'est posée lors de l'analyse de l'ensemble des modèles OSSAD est liée à l'identification des contenus des flux échangés, voire même à la nature des flux (informations, support d'informations ou matière). L'isolation des informations relatives au produit sur un modèle OSSAD n'est donc pas une chose triviale. D'où la nécessité, de construire des modèles reprenant là base des modèles OSSAD, en termes d'acteurs et échanges de flux, mais se focalisant plus sur la nature des flux échangés afin de mieux décrire le contenu de ces flux,

surtout quand il s'agit d'informations liées au produit. Dans ce but, nous utilisons le principe de la modélisation holonique.

- **Construction des modèles holoniques**

A partir des modèles réalisés sous OSSAD, mais aussi en capitalisant les connaissances des employés du moulin de Corbeil, nous avons projeté les données identifiés sur une modélisation de l'ensemble des processus de réalisée sous MEGA. Cette modélisation a permis d'identifier pour chacun des processus quels sont les flux reçus (matière ou données), et quels sont les flux émis. Etant donné que nous étudions la traçabilité des produits, en considérant non seulement les produits comme de simples objets mais également l'information qui leur est associé. Nous utilisons donc le concept d'holon tel qu'il est implémenté sous MEGA pour la représentation des produits, et des matières premières.

Dans notre modélisation, nous mettons en œuvre les notions de Processus, Flux entre processus (physique, informationnel ou holonique), et holon. Les holons représentent, ainsi, tout objet ou document circulant entre les différents processus de l'entreprise auquel nous attachons un certain nombre d'informations représentée que l'on représente par des attributs afin de bien identifier l'ensemble des informations associée à chaque élément modéliser, et de bien constituer la chaîne des relations de dépendance entre les différents objets. Cette chaîne permet de retracer l'ensemble des éléments relatifs à un produit donné, à une livraison ou à une commande. Pour exprimer la relation de dépendance entre les objets de l'entreprise, nous utilisons la relation de composition des holons.

Dans le cas d'étude, suite à la réception de la commande, le service administratif contacte les transporteurs répertoriés, pour désigner le transporteur qui pourra faire la livraison. Le client est ainsi informé par confirmation de commande, de la date de livraison et du transporteur assigné. Une fois la *commande* confirmée, le service administratif retransmet la *commande,* sous forme d'un *bon de chargement* (ou *bon de commande*), au service d'ensachage. Celui-ci, peut en fonction de sa connaissance des stocks de farine en cours, soit lancer directement un *programme d'ensachage* exécutable sur le champs, soit coordonner la production de nouvelles quantités de farine correspondant avant de préparer le *programme d'ensachage.* Un *programme d'ensachage* liste l'ensemble des *tâches d'ensachage prévues,* une *tache prévue* peut traiter plusieurs commandes client en même temps, si celles-ci concernent le même type

de farine et le même conditionnement (20 ou 50Kg). L'opérateur d'ensachage peut décider de réaliser une *tache prévue* en plusieurs fois, chacune des réalisations est nommée une *tâche réalisée,* chacune nécessitant un *ordre de fabrication* spécifique réalisé par l'opérateur. Chaque *tache réalisée* produit un certain nombre de palettes de sacs, disposant chacune d'une indentification unique. Une *palette de sacs* ne peut contenir qu'un seul type de farine. La *palette* est par la suite emballée puis préparée pour la livraison. Après emballage, chaque *palette* est disposée dans un emplacement spécifique, les *palettes* sont donc organisées dans des « *racks* » dédiés aux produits. Dans la figure ci-dessous, nous montrons un exemple d'échange de flux holoniques entre les processus de l'entreprise.

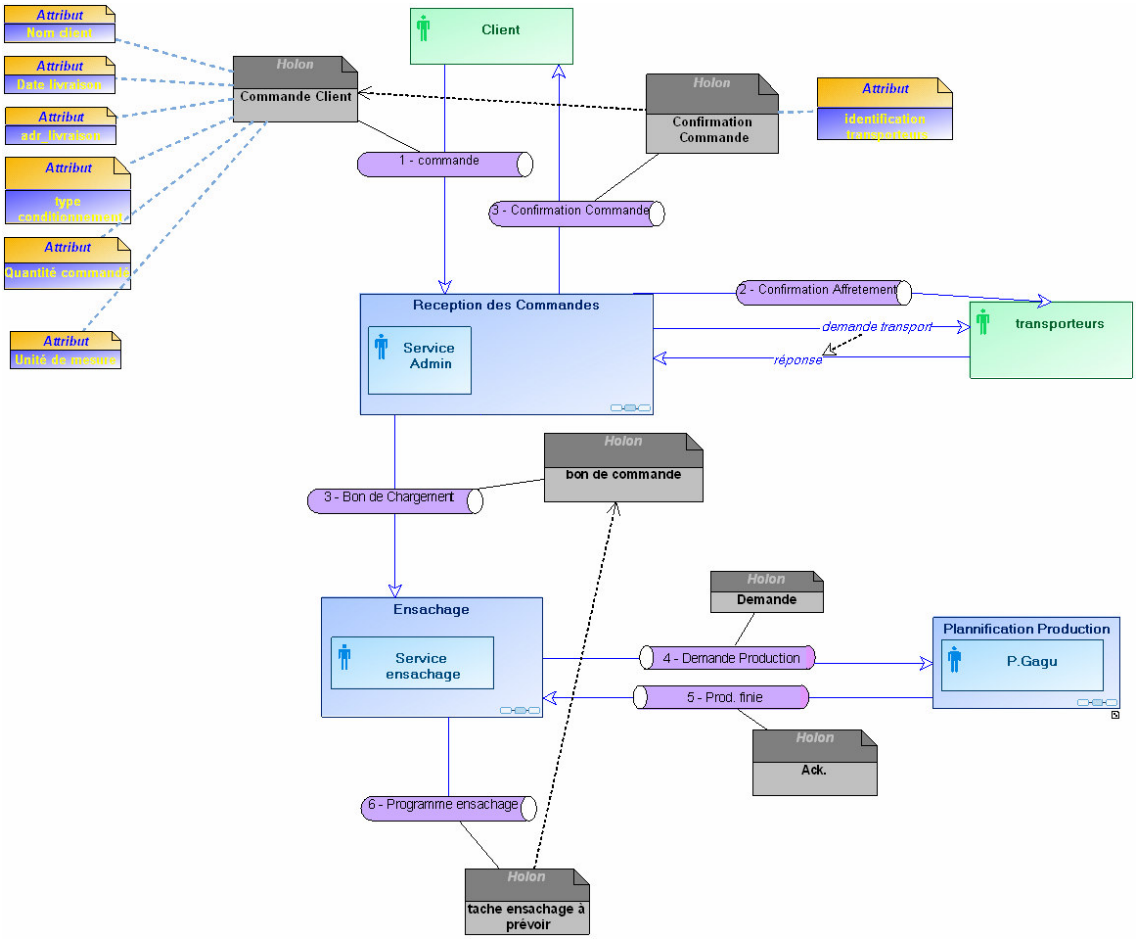


Figure 39. Extrait de la modélisation holonique réalisée dans l'environnement MEGA

La figure 39 représente les premiers échanges déclenchés par la réception d'une commande par le service administratif. Etant donné que la production dans l'entreprise est de type « batch », tout au long du cycle de production les instances particulières des produits ne sont pas identifiables (par exemple, la production de farine se fait par lot de 80 tonnes). La

difficulté principale consiste dans le fait que le produit farine est issu d'une production dite « en batch » et non une fabrication standard. L'identification des unités de produits étant difficile, la désignation des produits en fabrication se fait donc par *lots* et non par *unités*. Pour les produits en vrac, nous utilisons le concept d'holon, pour représenter les classes de produits et non les instances particulières d'une classe de produits. L'exemple de la figure 40 montre un modèle dans lequel on représente le concept d'holon, pour représenter le type d'un produit (farine) et les informations associées à ce type de produit. Dans cet exemple, lors de la préparation d'une mission de production, la farine est soutirée depuis les cellules de stockage, identifiée puis transférée vers la cellule 1, qui accueille la farine destinée à l'ensachage.

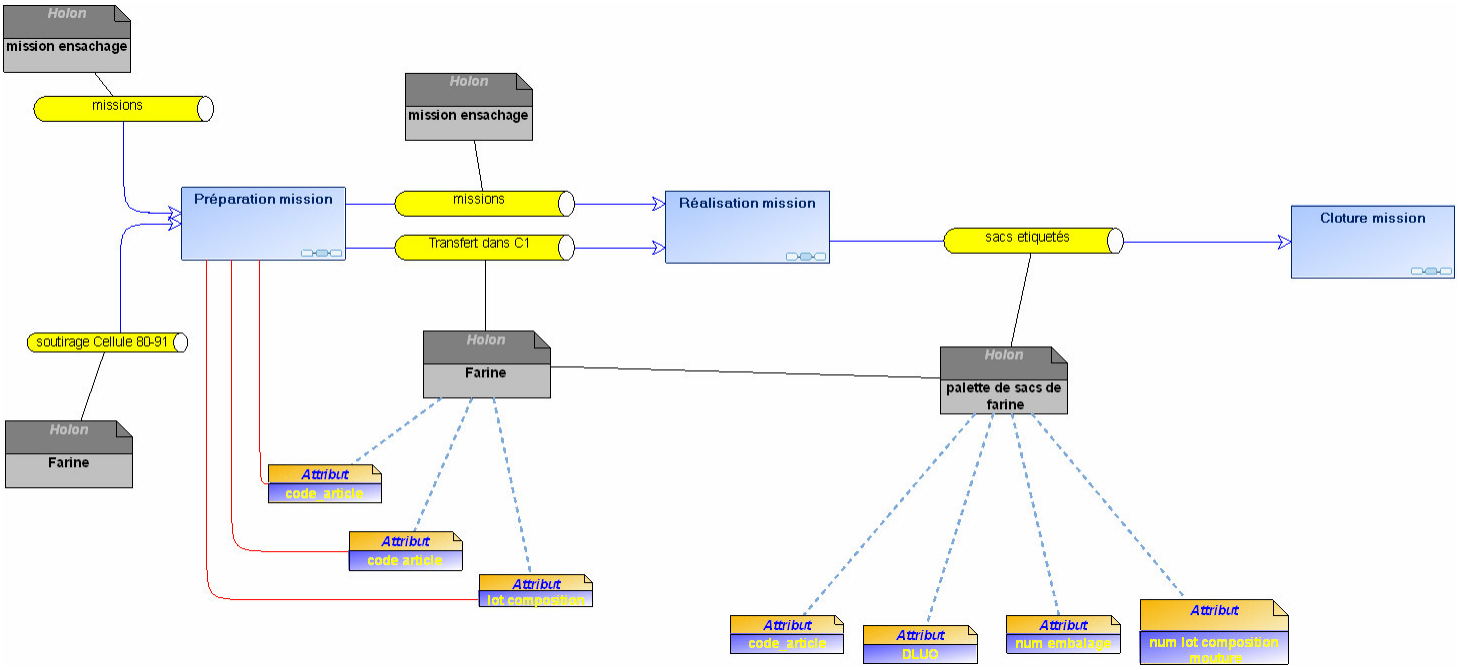


Figure 40. Exemple d'utilisation des holons pour la représentation du type d'un produit.

La construction des modèles holoniques de l'entreprise, a permis de constituer l'ensemble des informations relatives aux produits. La phase suivante consiste à capitaliser ces informations dans un schéma de données permettant de créer par la suite la base pour recueillir les données nécessaires pour la traçabilité.

Par la suite, nous montrons le passage des modèles holoniques vers le modèle de données pour la traçabilité. Puis, après peuplement de la base à l'aide de données réelles collectées sur le terrain. Nous déduisons les premiers constats et consignes pour l'amélioration de la définition et la collecte de données pour la traçabilité.

- **Des modèles holoniques vers les modèles de données**

En utilisant l'ensemble des modèles holoniques établis, nous extrayons automatiquement l'ensemble des données relatives à chacun des objets de l'entreprise défini par un holon. Nous construisons donc le schéma définissant l'ensemble des données identifiées pour chaque objet important issu de la modélisation holonique.

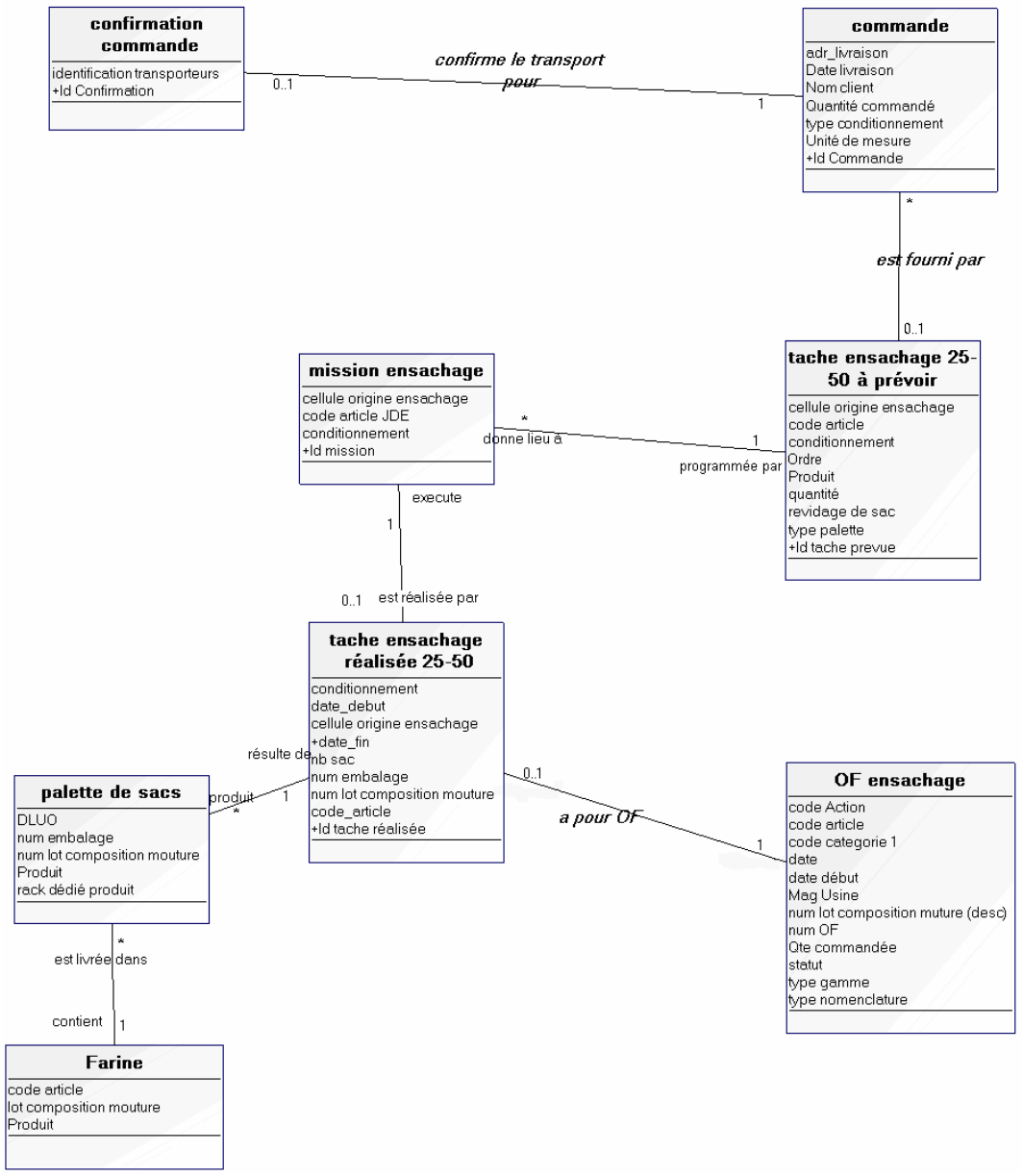


Figure 41. Diagramme de classes issu de la modélisation holonique.

La figure 41 montre le modèle représentant les d'objets identifiés dans l'entreprise. Ce modèle représente ainsi pour chaque type d'objets, l'ensemble des relations de dépendance ainsi que les caractéristiques (attributs) de chacune de ses classes, les cardinalités des associations ont été définies suivant les spécifications recueillies auprès des personnes concernées dans l'entreprise.

- **Une base de données pour la traçabilité**

A partir du diagramme de classes présenté dans le paragraphe précédent nous générons automatiquement la base de données destinée à contenir les données de traçabilité pour les palettes de sacs (*cf.* Figure 42).

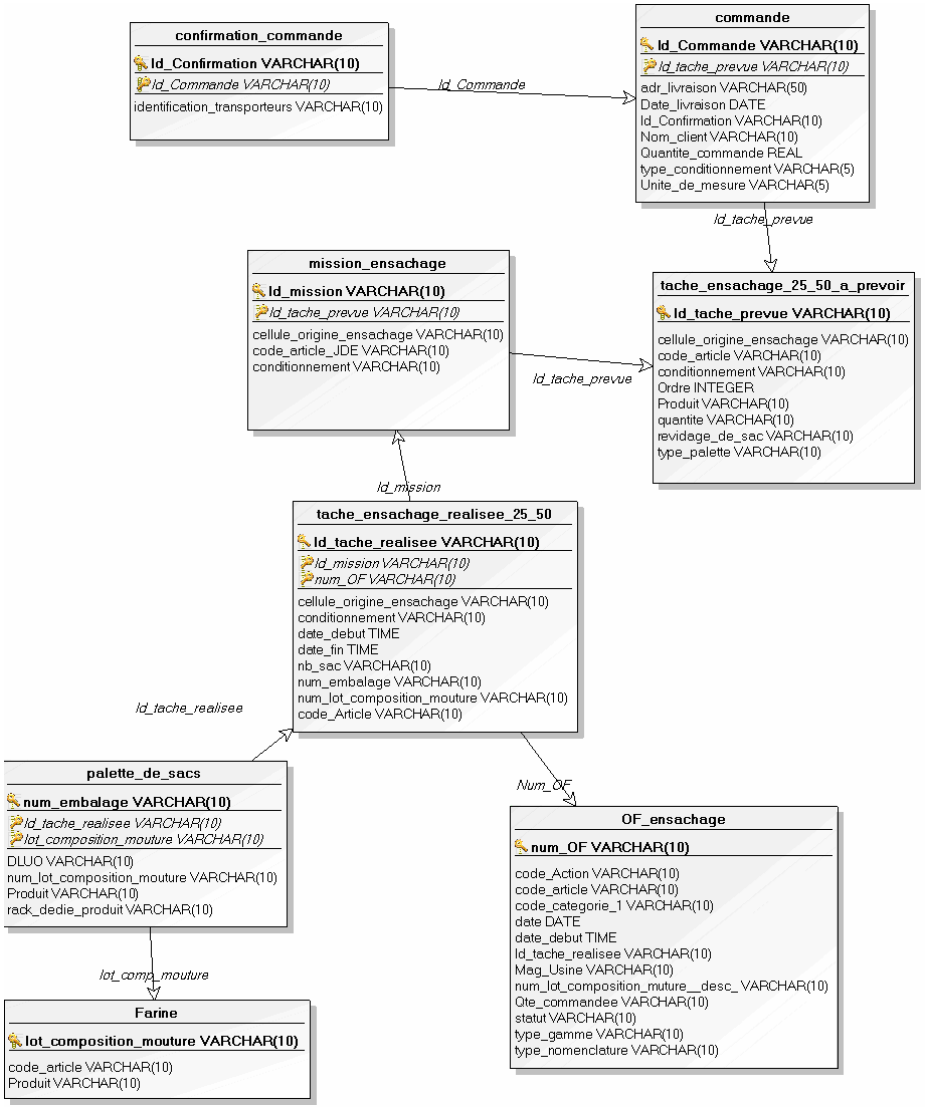


Figure 42. Schéma relationnel de la base de traçabilité.

- **Conclusions et constat**

Dans le contexte de la gestion de la traçabilité, l'approche holonique offre ainsi une démarche de modélisation structurée permettant d'obtenir un schéma de données pour la représentation des données relatives au produit telles qu'elles ont été identifiées lors de la modélisation des différents processus intervenant lors de la fabrication du produit.

Les outils de génération de code de ***MEGA Designer*** permette de générer du script SQL standard à partir à partir d'un modèle relationnel, nous avons donc utilisé cette fonctionnalité afin de préparer le script SQL permettant là création, des tables de la base de données destinée à recueillir les données de traçabilité des produits. Pour gérer la base de données, il a été décidé d'utiliser le système de gestion de base de données de Microsoft « *Microsoft SQL Server »* dans sa version 2005.

Pour le peuplement de la base, des routines *Visual Basic* ont été développées, permettant ainsi, la saisie automatique de données préalablement saisies par les différents services de l'entreprise dans un document Excel. En effet, l'échantillon de test contenait l'ensemble des données recueillies pendant 2 mois de production à temps complet (période Septembre-Octobre 2005), ceci équivaut à environ 15000 références de commandes client, 60 préparations de farine différentes, et la programmation de 200 programmes d'ensachage.

Pour le diagnostic et l'analyse des données recueillies dans la base, nous avons défini un ensemble de requêtes pour analyser traçabilité des produits, certaines incohérences dans les données ont été détectées.

Par exemple :

- *Requête a* – Retrouver les palettes issues provenant d'un lot de mouture.

  Après analyse des données de la base, il est possible de retrouver les palettes mais il est impossible de retrouver le client car le chemin menant vers le client n'est pas toujours correctement construit.

  Causes de la défaillance :

  - Le lien Commande/Tache est totalement inexistant dans les données collectées
  - Le lien Tache/mission est sommairement renseigné.

- *Requête b* – Retrouver les palettes suite à un problème sur une cellule d'ensachage

L'identification des palettes est possible, mais pas celle du client qui a reçu les palettes, ce problème est du aux mêmes causes que celles détectées lors de l'analyse de la *requête a*.

- *Requête c* – Retrouver les palettes suite à un problème sur un numéro d'emballage

  Pour les mêmes raisons citées auparavant (voir requête a), Il est impossible de trouver le client correspondant à une mission ou à une tache réalisée. Il est cependant possible d'identifier les palettes et les missions concernées en cas de problèmes sur un numéro d'emballage, mais aucun lien permettant de retrouver le client à contacter.

Le constat de cette première batterie de test, permet de dire que la récolte des données se faisant généralement manuellement reste très défaillante, et de ce faite la traçabilité qui en découlera sera biaisée suite au manque d'informations. Pour contrer ce problème, l'entreprise a décidé donc d'intégrer la base de données de traçabilité directement dans leur système d'information automatisé pour une récolte d'informations plus structurée et plus cohérente.

V. Conclusion

Dans ce chapitre, nous avons tout d'abord montré une implémentation de l'approche holonique dans un atelier de modélisation standard « *MEGA Suite* ». Dans cet outil, nous avons donc intégré les concepts issus de l'approche holonique pour la modélisation des produits de l'entreprise. Par la suite, nous avons personnalisé les diagrammes de processus proposé sous *MEGA* afin de consolider la modélisation de l'entreprise en utilisant les concepts intégrés.

Pour montrer la compatibilité de la méthode holonique et le reste des approches de modélisation utilisées en entreprise, nous avons montré comment les différentes étapes de la modélisation holonique peuvent être intégrés dans un cadre de modélisation d'entreprise couvrant l'ensemble des aspects de l'entreprise, le cadre *Zachman*.

Afin de tester l'applicabilité de l'approche holonique pour la modélisation sur un cas pratique dans un contexte industriel, un cas d'étude a été réalisé dans le cadre d'une collaboration avec un partenaire industriel, dans laquelle nous avons utilisé la modélisation holonique d'une partie des processus de la branche meunerie d'un grand groupe industriel, et ce afin d'étudier et de proposer une représentation des produits de l'entreprise pour la réalisation d'un système de gestion de la traçabilité. Les résultats obtenus montrent l'apport d'une modélisation centrée sur le produit et ses informations, représentée dans notre cas par l'approche holonique.

Dans le chapitre suivant, nous formalisons les mappings nécessaires pour la réalisation de d'une interopérabilité sémantique basée sur la transformation de modèles, permettant aux instanciations du meta-modèle holonique de jouer le rôle de sources unificatrices des données relatives au produit, et ce afin de mettre en cohérence l'ensemble des représentations du produits se référant à cette représentation holonique.

Chapitre 4 :
L'approche dirigée par les modèles pour une interopérabilité orientée produit

I. Introduction

L'objectif de ce chapitre est de montrer comment les modèles issus de l'approche de modélisation holonique peuvent interopérer avec des modèles spécifiques à d'autres contextes. Cette interopérabilité au niveau des modèles permet de mettre en œuvre l'interopérabilité orientée produit des applications au niveau de l'exécution, permettant ainsi la mise en cohérence des différentes représentations et vues du produit.

Tout d'abord, nous étudions l'apport de l'approche MDA et les transformations de modèles dans l'interopérabilité dans le cas général. Nous formalisons ainsi les propriétés mathématiques à prendre en compte lors de la spécification de mappings pour mettre en œuvre l'interopérabilité sémantique dirigée par les modèles entre applications. Dans cette formalisation, nous nous proposons une définition des mappings de modèles et meta-modèles basée sur la théorie des ensembles.

Par la suite, nous illustrons par des exemples, la réalisation des mappings pour l'interopérabilité orientée produit, basée sur le meta-modèle holonique. Ces exemples sont basés sur des langages de modélisation d'entreprise, le premier issu du monde de la modélisation organisationnelle de haut niveau de l'entreprise, le second concerne la modélisation des systèmes d'exécution de la production dans les ateliers de l'entreprise.

Nous montrons, en suite, comment ces mappings peuvent être implémentés et pris en compte au sein de notre outil de l'approche de modélisation holonique dans l'environnement ***MEGA.***

Enfin, nous montrons l'apport de ses mappings pour l'approche holonique dans le cadre de modélisation Zachman, puis nous montrons comment l'ensemble de l'approche est appliqué sur un cas d'application issu de l'environnement AIPL.

II. Interopérabilité et transformation de modèles

L'approche MDA offre un support méthodologique pour traiter le problème de l'interopérabilité des applications, à titre d'exemple nous citerons les travaux de (Bourey et al. 2005) réalisés dans le cadre du projet INTEROP Noe. Nous postulerons ainsi le fait que chaque application (ou système) possède un modèle représentant un univers du discours (ou une partie de cet univers). Chaque modèle étant lui-même construit sur les bases d'un meta-modèle (selon les quatres niveaux ontologiques de l'approche MDA *cf.* Figure 15), la définition qui suit essaie d'établir la relation qui existe entre l'interopérabilité sémantique des applications et les échanges de modèles (Baïna and Morel 2006).

Définition 10. *Etant données deux applications A et B ; nous dirons que A et B sont sémantiquement interopérables si et seulement si nous pouvons construire une instance du modèle de B à partir d'une instanciation du modèle de A et construire une instance du modèle de A à partir d'une instanciation du modèle de B.*

Etablir l'interopérabilité entre applications revient donc à définir les transformations possibles entre les modèles se trouvant à la base de ces applications. (Lemesle 1998) introduit la transformation de modèles en établissant des règles de transformation entre meta-modèles (*cf.* Figure 43).

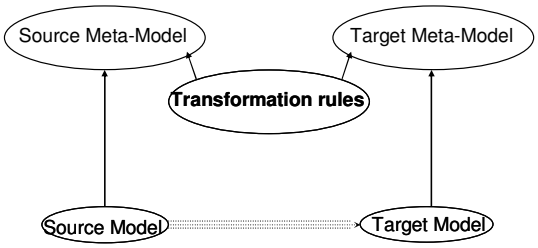


Figure 43. Transformation de meta-modèle (Lemesle, 1998)

Cependant la considération d'un meta-modèle de référence jouant le rôle d'une passerelle de communication permet de réduire le nombre de transformations nécessaires pour couvrir l'ensemble des connexions possibles entre les différents systèmes appelés à interopérer. En effet, les solutions d'interopérabilité basées sur un modèle de référence permettent de réduire

la complexité du réseau des connexions point à point entre les différents systèmes communiquant.

L'approche d'interopérabilité orientée produit telle qu'elle est définie peut être positionnée au niveau 2 du model LCIM. En effet, le niveau 2 suppose un modèle de référence commun permettant l'alignement des données statiques. Dans notre cas, le meta-modèle holonique joue le rôle de modèle de référence pour l'échange d'informations relatives au produit entre les différentes applications.

Dans ce qui suit nous identifions, tout d'abord, les mécanismes mis en place pour l'établissement de transformations entre modèles afin de spécifier l'interopérabilité entre applications ayant des modèles et meta-modèles différents. Par la suite, dans le chapitre suivant nous montrons comment ces mécanismes sont mis en place pour permettre l'échange d'information à travers des modèles holoniques basés sur le meta-modèle holonique.

II.1. Mappings de meta-modèles

Dans cette section, nous nous focalisons essentiellement sur l'étude de la relation entre mappings et transformations de modèles dans le contexte MDA.

Pour cela, nous commençons tout d'abord par proposer une définition plus formelle des notions de meta-modèle et de mapping de meta-modèles. Pour cela, nous adoptons une approche algébrique, basée sur la théorie algébrique pour les ontologies et les mappings d'ontologies définie par (Kalfoglou and Schorlemmer 2003).

Selon (Kalfoglou and Schorlemmer 2003), une ontologie *O* peut être définie sous forme d'un couple $O = (S, A)$, où *S* est la signature *(ontologique)* – décrivant un vocabulaire – et *A* est un ensemble d'*axiomes (ontologiques)* – spécifiant l'interprétation du vocabulaire dans l'univers du discours. Typiquement, une signature est munie d'une structure mathématique ; cette structure est généralement exprimée à l'aide d'une hiérarchie de concepts ou de symboles définissant un ensemble partiellement ordonné. Les relations entre les éléments d'une signature se résument en deux types de relation : « est » et « est sous type de ».

La figure 44 montre un exemple simple d'une relation sémantique liant les éléments de l'ontologie classifiant les types de flux et leur contenu définis par le meta-modèle holonique.

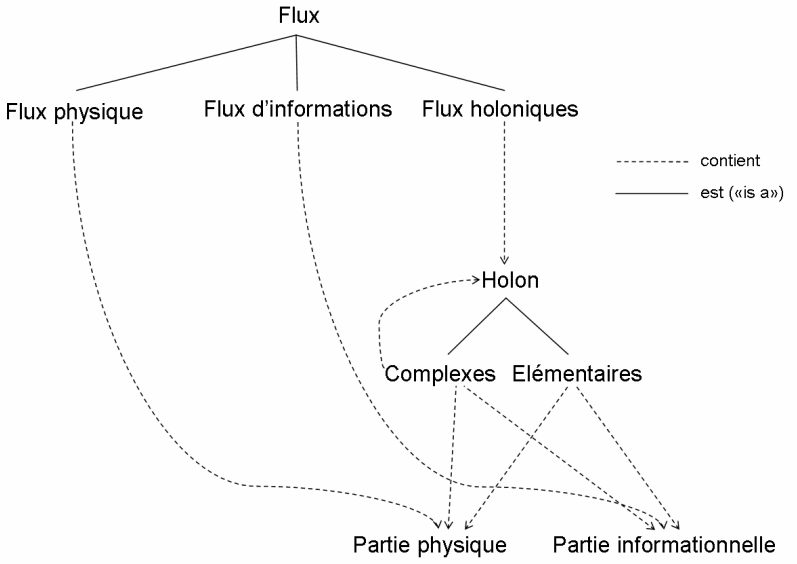


Figure 44. Exemple de signature partiellement ordonnée d'une ontologie

Par analogie avec la définition des ontologies, nous établissons de la même manière une définition ensembliste d'un meta-modèle. Un meta-modèle peut donc être défini comme étant un ensemble de concepts issus d'une ontologie, et un ensemble de relations ou associations entre ces concepts. Les relations entre concepts d'un meta-modèle peuvent être basées sur les axiomes définis dans l'ontologie. Nous définissons, donc, un meta-modèle comme un couple $M = (C, R)$ où C est l'ensemble des concepts utilisés dans M, et R l'ensemble des relations entre ces concepts. Etant donné que les concepts utilisés par le meta-modèle M sont issus d'une ontologie $O = (A, S)$, la structure sémantique définie sur A se retrouve donc projetée sur C. Cette structure sémantique est exprimée, en général, par un ordre partiel sur l'ensemble des concepts. Pour préserver la structure sémantique lors de la transformation d'un meta-modèle ou bien d'une ontologie il faut que l'ordre partiel soit conservé.

Un mapping entre meta-modèles est ainsi défini comme étant une fonction qui relie entre eux les concepts utilisés dans ces meta-modèles ; cette fonction doit permettre aussi de préserver la structure sémantique entre les concepts définis pas chacun des vocabulaires. Plus formellement, soit M = (C, R) et M' = (C', R') deux meta-modèles ; nous appelons mapping total de M vers M,' une application f de C vers C', préservant la structure sémantique (représentée par un ordre partiel). Un mapping est donc un morphisme entre deux vocabulaires dont le but est de conserver la sémantique des termes transcrits. Ce morphisme f

traduit l'ensemble des relations de correspondance existant entre l'ensemble des concepts définis par *C* et *C'* (équivalence, généricité, ou lien faible).

Généralement, il est difficile de d'identifier de mappings complet permettant de transformer tous les concepts définis par un meta-modèle en concepts appartenant à un autre meta-modèle (Madhavan et al. 2002). Pour s'adapter à des mappings plus faibles et moins couvrants, nous définissons la notion de mapping partiel entre meta-modèles. Un mapping entre deux meta-modèles M1 = (C1, R1) et M2 = (C2, R2) est dit partiel s'il se réduit à un mapping total entre une sous-partie du meta-modèle de départ M1 vers une sous-partie du meta-modèle d'arrivée M2.

II.2. Les règles de transformations

Dans cette section nous nous intéressons aux différents de transformations identifiables entre deux modèles ou meta-modèles différents. Pour illustrer les règles de transformation pouvant exister entre deux meta-modèles, (Roque 2005) définit la notion de composants d'un meta-modèle, ainsi que les correspondances sémantiques entre deux composants (ou concepts) Comp1 et Comp2 appartenant à différents meta-modèles de la façon suivante (Roque 2005).

Soient Comp1 et Comp2 deux concepts définis dans deux meta-modèles différents :

- **cas 1 :** Comp1 ≡ Comp2

Comp1 est équivalent à Comp2 ; cette équivalence ne prend pas en compte les relations reliant Comp1 et Comp2 au reste des concepts, mais uniquement la sémantique exprimée par les concepts Comp1 et Comp2 eux-mêmes ;

- **cas 2 :** Comp2 ⊇ Comp1

Comp1 est toujours utilisé quand Comp2 est utilisé ; l'inclusion ici exprime le fait que la sémantique de Comp2 couvre complètement la sémantique de Comp1 ;

- **cas 3 :** Comp1 ∩ Comp2 ≠ ∅

Comp1 est parfois utilisé quand Comp2 est utilisé, mais pas toujours ; l'intersection de Comp1 et Comp2 définit le sens commun aux deux concepts Comp1 et Comp2.

REMARQUE. — Le cas 2 est un cas particulier du cas 3 dans lequel l'intersection de Comp1 et Comp2 (Comp1 ∩ Comp2) serait égale à Comp2.

D'après les correspondances sémantiques définies par(Roque 2005), on peut dire qu'il y a 3 types de correspondances entre deux concepts Comp1 et Comp2 (cf. figure 5) :

- **équivalence :**

Comp1 et Comp2 sont équivalents : Comp1 ≡ Comp2

- **généricité :**

Comp2 est plus générique que Comp1 : Comp2 ⊇ Comp1

- **lien faible** (ni équivalence ni généricité) :

Comp1 ∩ Comp2 ≠ ∅ ∧ Comp1 ∩ Comp2 ≠ Comp2 ∧ Comp1 ∩ Comp2 ≠ Comp1

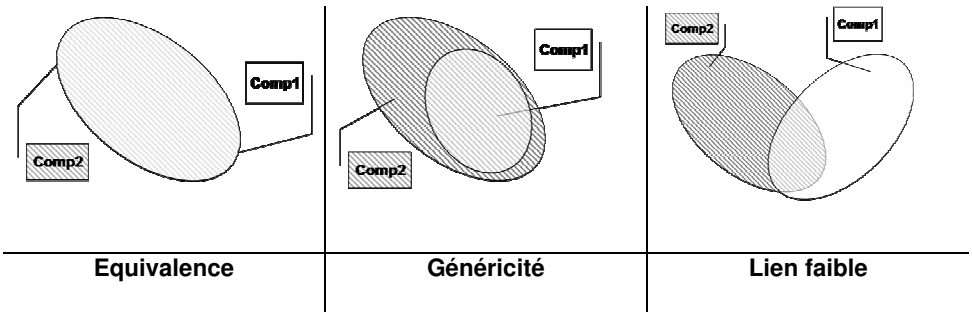


Figure 45. Illustrations des correspondances entre les sémantiques des composants

Pour parler d'interopérabilité sémantique, un simple morphisme unidirectionnel ne suffit pas. En effet, si un morphisme permet de passer d'une instance de M1 à une instance de M2, il ne permet pas forcement l'inverse ; or comme énoncée précédemment, la définition même de l'interopérabilité sémantique est basée sur cette symétrie (cf. définition 1) ; il faut donc retrouver la même dualité au niveau des mappings. La dualité des mappings existe sous deux formes distinctes.

- **Cas 1 :** deux mappings qui forment un mapping bijectif entre M1 et M2.

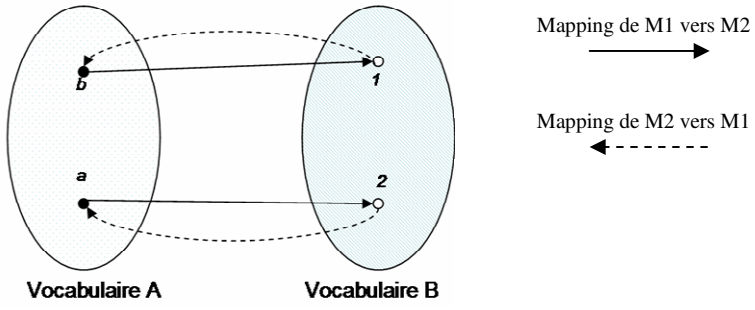


Figure 46. Mapping bijectif entre deux vocabulaires distincts A et B

- **Cas 2 :** Deux mappings indépendants ; un mapping de M1 vers M2 et un mapping de M2 vers M1 :

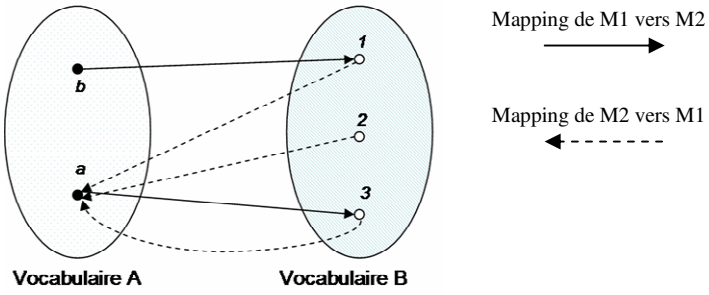


Figure 47. Mappings distincts non bijectifs entre deux vocabulaires distincts A et B

REMARQUE. — Le problème, dans ce cas, est que ces mappings ne conservent pas forcément la sémantique s'ils sont appliqués successivement (*i.e. instance de M1 → instance de M2 → nouvelle instance de M1)* :

$$b \xrightarrow{\text{M1 vers M2}} 1 \xrightarrow{\text{M2 vers M1}} a \ ;$$

II.3. Vers une classification interopérabilité sémantique

Dans cette partie, nous proposons une définition pour l'interopérabilité entre application dans un cadre MDA en nous basons sur les différents type de mappings identifiés entre les différents meta-modèles.

Définition 11. *Considérons A et B deux applications ; M_A et M_B leurs meta-modèles respectifs. A et B sont dites sémantiquement interopérables si et seulement si il existe un mapping bijectif (isomorphisme) de M_A vers M_B, que nous noterons f. La bijection de f assure que l'on peut construire une instance du modèle de B à partir d'une instanciation du modèle*

de A (en utilisant f) et construire une instance du modèle de A à partir d'une instanciation du modèle de B (en utilisant le mapping inverse f^{-1}).

Suite à cette définition, nous identifions trois niveaux d'interopérabilité basés sur les propriétés des mappings existants :

- **Niveau 0 :** interopérabilité faible

  Il n'existe pas d'isomorphisme partiel entre MA et MB. Cependant, il se peut que des mappings non bijectifs existent entre MA et MB ;

- **Niveau 1 :** interopérabilité sémantique partielle

  Il existe un isomorphisme partiel entre M_A et M_B. Il existe une sous-partie de M_A que l'on notera M'_A *et* une sous-partie de M_B *(M'_B),* telles que l'interopérabilité entre M'_A et M'_B est de niveau 2 (M'_A et M'_B sont équivalentes) ;

- **Niveau 2 :** interopérabilité sémantique totale

  Il existe un isomorphisme total entre M_A et M_B *:*

  Dans ce cas de figure, tout concept apparaissant dans M_A a son équivalent dans M_B et *vice versa*, ce qui signifie que M_A et M_B sont équivalents. L'interopérabilité sémantique totale (de niveau 2) reste la plus difficile à établir, étant donné qu'il est très rare que deux meta-modèles M_A et M_B soient totalement équivalents. En général, on cherche plutôt à établir une interopérabilité partielle entre les deux meta-modèles M_A et M_B.

Nous avons vu dans la section précédente que pour une interopérabilité sémantique totale (de niveau 2), il faut définir un mapping composé essentiellement de règles d'équivalence. En effet, les règles d'équivalence possèdent, par définition, la bijection nécessaire pour établir l'interopérabilité sémantique entre deux systèmes. Cependant, il arrive que l'on ne puisse pas identifier de règles d'équivalence entres les concepts des meta-modèles étudiés. Dans (Baïna et al. 2006a), nous proposons un méthode de raffinement, permettant de déduire une relation d'équivalence à partir de relations de généricité ou des liens faibles entre concepts.

III. Application

Dans ce qui suit, nous illustrons les mappings de meta-modèles pour l'interopérabilité en utilisons deux exemples de mapping entre le meta-modèle holonique et deux meta-modèles spécifiant chacun une représentation différente du produit dans l'entreprise. Le premier exemple est issu du monde de la modélisation organisationnelle de haut niveau de l'entreprise, le second concerne la modélisation des systèmes d'exécution de la production dans les ateliers de l'entreprise.

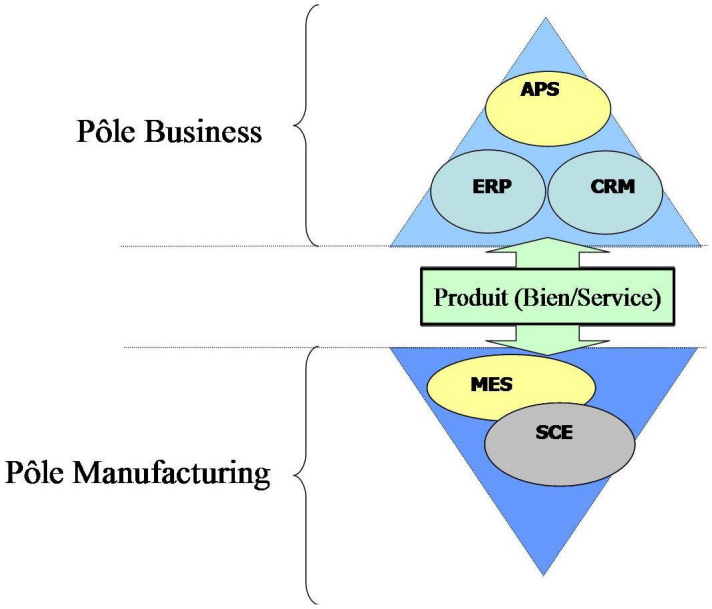


Figure 48. Structure des entreprises de production

Nous présentons deux exemples de mappings depuis le meta-modèle holonique vers deux meta-modèles se situant chacun à un niveau différents dans l'entreprise:

- Le premier exemple concerne le meta-modèle de UEML (Unified Enterprise Modelling Language) qui a été créé pour répondre à la disparité des innombrables langages et méthodes de modélisation d'entreprise (Vallespir et al. 2003). UEML est ainsi défini comme le langage générique permettant la fédération des différents modèles issus de la modélisation d'entreprise. Nous avons choisi UEML comme un exemple de langage pour la modélisation du domaine « Business » de l'entreprise.

- Le deuxième exemple concerne le meta-modèle du standard IEC 62264 utilisé pour des applications du niveau « Manufacturing ».

Nous avons choisi deux exemples hétérogènes pour montrer la polyvalence B2M que propose l'approche de modélisation holonique, permettant ainsi aux modèles holoniques d'être exploités, par la suite, soit dans des applications du pôle « Business » de l'entreprise soit dans des applications du pôle « Manufacturing ».

III.1. UEML

UEML (Unified Enterprise Modelling Language) est le résultat du projet UEML (UEML 2002) dont l'objectif principal était de résoudre le problème de l'existence d'une multitude de langages de modélisation d'entreprise (Vernadat 1996). Ce projet avait pour objectif unique de montrer la faisabilité d'un langage de modélisation d'entreprise (Panetto et al. 2004) décrivant l'ensemble les notions et concepts pertinents pour la modélisation d'un cas particulier. Les résultats de ce projet sont maintenant étendus dans le cadre d'une activité de recherche du REX INTEROP NoE. UEML (que l'on qualifiera de UEML 1.0) est construit autour d'un meta-modèle définissant un ensemble de concepts (Berio et al. 2003) identifié comme étant l'unification sémantique de quelques uns des langages de modélisations les plus répandus dans le monde de l'entreprise (GRAI, IEM, EEML, etc.).

L'objectif principal de UEML n'est pas de remplacer les différents langages de modélisation dans l'entreprise, mais de jouer le rôle d'un esperanto permettant les échanges entre différentes applications basées chacune sur un modèle différent issu d'un langage de modélisation. Pour le développement de UEML, trois langages de modélisation d'entreprise ont été retenus ; GRAI, IEM et EEML. Après analyse de ces différents langages, seuls les concepts apparaissant au moins dans deux d'entre eux ont été intégrés dans le meta-modèle principal de UEML (Panetto et al. 2004).

Le meta-modèle d'UEML est le résultat d'un effort pour unifier les concepts de plusieurs langages de modélisations. Cependant, il n'existe pas d'applications d'entreprise utilisant UEML, nous n'avons donc pas choisi UEML pour ses implémentations mais pour son aspect unificateur, en effet à lui tout seul il représente plusieurs langages de modélisation du niveau organisationnel de l'entreprise.

Dans ce qui suit, nous présentons quelques un des concepts principaux définis par UEML *:*

Activité (Activity). Une activité représente une description générique d'une partie du comportement de l'entreprise transformant un certain nombre d'entrée en sorties. Elle peut être décomposée en d'autres activités et regroupe plusieurs exécutions d'activité ayant des propriétés communes.

Objet: Un objet est toute sorte d'entité qui peut circuler via un flux; autrement dit, un objet est une entité qui peut être utilisée ou produite lors de l'exécution d'une activité, que ce soit une entité physique (Ressource) ou une entité purement informationnelle (Objet informationnel)

- *Objet Informationnel.* Un Objet informationnel est un objet constitué exclusivement d'information.
- *Ressource.* Une Ressource est un sous-type d'objet nécessaire pour l'exécution d'une activité. Il existe deux types de ressources:
    - les ressources physiques,
    - et les ressources humaines.

Flux. Un Flux représente l'écoulement d'un certain nombre d'objets depuis une source vers une destination. Un flux peut être de plusieurs types:

- *Flux d'entrées/sorties*. représente le trafic d'objets d'entrée/sorties entre activités. Si le contenu est un objet de sortie, alors le flux est connecté à un port de sortie *OutputPort* de l'activité émettrice, cela voudrait dire que l'objet est produit lors de l'exécution de l'activité. Si par contre, le contenu est un objet d'entrée ; le flux est alors lié à un port d'entrée *InputPort* au niveau de l'activité réceptrice, ceci signifie que l'objet est probablement consommé lors l'exécution de l'activité.
- *Flux de ressources.* véhicule les ressources nécessaires pour l'exécution d'une activité.
- *Flux de contrôle.* véhicule l'information permettant d'établir le contrôle d'activité. Il définit également la relation de précédence entre activité, une activité envoyant un flux de contrôle à une autre activité s'exécute forcément avant celle-ci.

La figure 49 représente le meta-modèle définissant les concepts de base de UEML ainsi que les relations entre ces concepts.

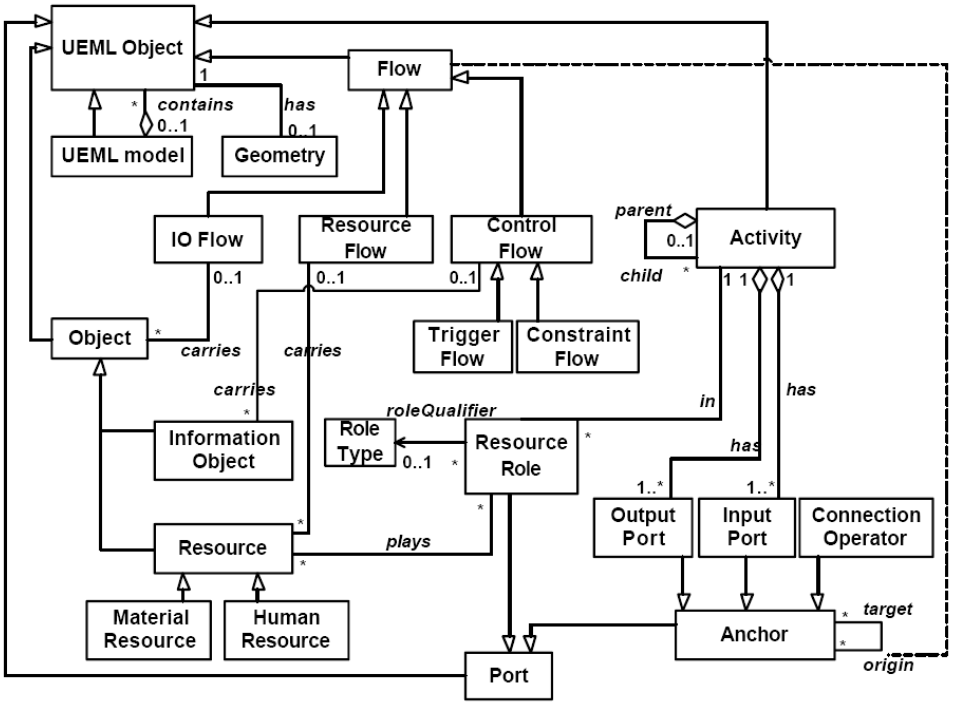


Figure 49. Meta-modèle UEML (UEML 2002)

En vue de construire les mappings pour l'interopérabilité basé sur le meta-modèle holonique comme modèle de référence, nous tentons par la suite d'étudier la correspondance entre le vocabulaire et les concepts définis par UEML et les concepts définis dans le meta-modèle holonique. Après avoir étudié, un à un, les concepts du meta-modèle holonique et les concepts d'UEML, nous proposons un mapping entre les deux vocabulaires composé de correspondances que nous résumons dans le Tableau 3 qui présente une à une l'ensemble des règles du mapping, en précisant le type de la règle : équivalence (≡), inclusion (⊆, ⊇) ou lien faible (∩).

Tableau 3: Correspondance entre Holons et les concepts UEML

Meta-modèle Holonique	Type de la correspondance	Meta-modèle UEML
Holon	⊆	Objet
Partie Informationnelle	⊆	Objet Informationnel
Partie Physique	⊆	Ressource Physique
Flux physique	⊆	Flux de ressource
Flux informationnel	⊆	Flux de contrôle'
Flux d'holons	⊆	Flux d'entrées/sorties

Ce mapping n'est qu'un mapping partiel, il ne couvre pas la totalité des concepts définis par les deux meta-modèles. Les correspondances entre les deux meta-modèles ne sont pas forcément des relations d'équivalence, le mapping n'est donc pas isomorphe.

III.2. IEC 62264

Le standard IEC 62264 (2002) spécifie l'interface d'échange de données et de modèles avec le niveau atelier dans l'entreprise. Il définit les modèles et interfaces entre les activités de l'entreprise et les activités du niveau contrôle. Chaque niveau traite une vue particulière de l'intégration du niveau contrôle avec le reste de l'entreprise. Ces modèles peuvent être classes en deux catégories: modèles opérationnels et modèles de ressources. L'IEC 62264 couvre plusieurs aspects du domaine du contrôle de production ; les capacités de production, la définition des produits, la planification et les performances de production (*cf.* Figure 50). Pour ce faire, elle définit tout un ensemble de modèles jouant le rôles d'interfaces d'échange entre les activités du niveau contrôle les autres activités de l'entreprise. Dans ce qui suit, nous introduisons brièvement les modèles les plus importants.

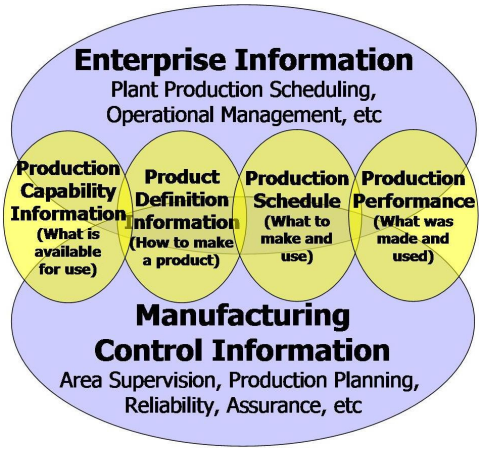


Figure 50. Aspects de l'intégration "Business to Manufacturing" dans l'IEC 62264

- Le modèle matériel (*material model)* est un modèle de ressource qui concerne les matières premières, les objets, les produits et les groupements de produits (lots et sous-lots) circulant dans le niveau atelier. Ce modèle spécifie la définition des classes de matière et les informations concernant ces classes. L'information traitée par ce modèle inclut l'inventaire des éléments bruts, finis, et intermédiaires.

- Le modèle de définition du produit (*product definition model)* est le modèle de l'IEC 62264 qui définit l'information relative à la nomenclature des produits, aux règles de production, les ressources utilisées et les besoins spécifiques pour la réalisation d'une tache.

- Le modèle d'équipement (*equipment model*) contient les informations spécifiques aux équipements, classes d'équipement, leurs capacités, ainsi que les tests et la maintenance qui leur sont associés.

- Le modèle de personnel (*personnel model*) définit les informations décrivant le personnel, les classes de personnel, leurs compétences et leurs qualifications.

Pour le mapping de l'holon dans les modèles de l'IEC 62264, nous intéressons tout spécialement au « *Material Model*», étant donné qu'il correspond mieux au contenu des modèles holoniques (*cf.* Figure 51). Le modèle matériel représente les produits finis, et les produits intermédiaires ou matières premières utilisés pour leur fabrication.

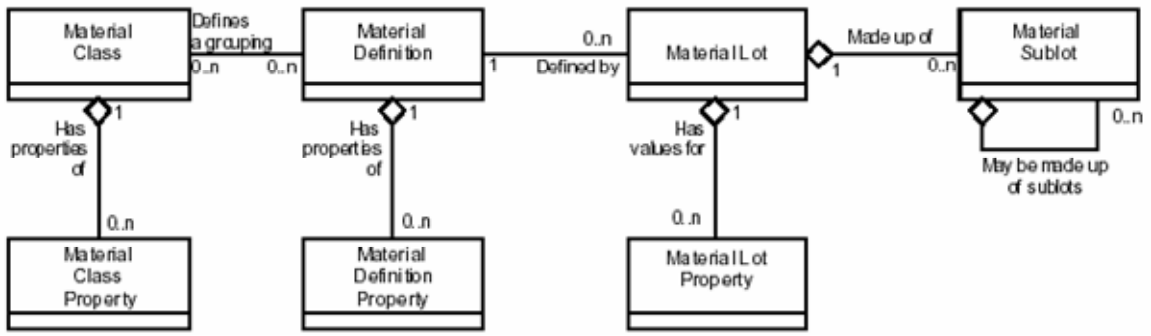


Figure 51. Le modèle matériel l'IEC 62264

De même, que pour le meta-modèle holonique et UEML, nous proposons dans le Tableau 4 un mapping entre les concepts holoniques et le modèle matériel du standard IEC 62264.

Tableau 4: Mapping les concepts holoniques et les concepts IEC 62264

Meta-modèle Holonique	Type de la correspondance	IEC 62264
Holon	∩	Material Lot ∪ Material sublot
Partie Informationnelle	⊇	Material Definition
Partie Informationnelle	⊇	Material Class
Propriétés et Attributs	⊇	Material Class property
Propriétés et Attributs	⊇	Material Definition property
Propriétés et Attributs	≡	Material lot property definition

Certains concepts du meta-modèle holonique correspondent à plusieurs concepts dans le *Material Model*, ce qui donne lieu à des règles *1→ n.* Par exemple, une partie holonique d'un holon peut être transformée en un élément de type *Material Définition* ou bien un élément de type *Material Class*. Ceci est dû au fait que les propriétés et attributs spécifiés dans une partie informationnelle peuvent être décrire la classe de l'objet et non une instance du produit, ce cas de figure arrive lorsque les valeurs de ces propriétés ou attributs sont conditionnées par les caractéristiques de classe de l'objet en question et non à une instance précise du produit.

Lors de cette étude de correspondance, des incohérences ont été identifiées au niveau du modèle matériel de l'IEC 62264. En effet, dans le modèle matériel nous avons remarqué que, contrairement au type Material Lot, les éléments de type Material Sublot ne sont pas décrits par des propriétés Material Lot Property, ni par des Material Classes, ni Material Definitions. Or un Material Sublot doit lui aussi pouvoir être correctement défini. Suite à nos travaux, une proposition pour amender le *Materiel Model* de l'IEC 62264 a été déposée auprès de l'organisme concerné (IEC/ISO JWG15).

III.3. Synthèse

Les mappings identifiés entre UEML et IEC 62264 d'un coté et le meta-modèle holonique de l'autre, ne couvrent pas la totalité des concepts définis dans ces meta-modèles, en effet, étant donné que chaque meta-modèle couvre des domaines qui ne concerne pas les autres modèles il est normal que les mappings définis ne puissent pas couvrir l'ensemble des concepts. De plus, ces mappings ne sont pas isomorphes, en effet, certaines relations de correspondance ne représentent pas des relations d'équivalence. Généralement, l'établissement de mappings isomorphes est une tâche plutôt laborieuse voir impossible. L'interopérabilité sémantique établie en utilisant la technique des mappings dépend essentiellement de la bijection des mappings identifiés. Pour que la bijection soit effective, il faut séparer et isoler les données provenant de la modélisation holonique du reste des données provenant d'autres sources.

Dans le cas du mapping UEML ↔ Meta-modèle Holonique, toutes les « parties informationnelles » sont des « Information Objects », mais tous les « Information Objects » ne sont pas des « parties informationnelles », puisque les applications utilisant UEML vont aussi utiliser le type « Information Objects » pour représenter d'autres types d'objets, autres

que des « parties informationnelles ». Il suffit donc de restreindre les transformations uniquement aux données issues de la modélisation holonique. On peut donc considérer les mappings comme étant isomorphes mais seulement en ce qui concerne les données résultant de la modélisation holonique du produit. Par la suite, pour établir l'interopérabilité entre applications, il est possible d'utiliser la transitivité des mappings identifiés avec le meta-modèle holonique. Etant donné deux concepts C et C' appartenant à deux meta-modèles d'applications M et M', et qui sont tous deux associés au même concept du meta-modèle holonique; C et C' sont considérés sémantiquement reliés dans le cadre de la modélisation holonique, la relation (C, C') fait donc partie du mapping déterminant l'interopérabilité entre M et M'. En établissant le lien entre un meta-modèle quelconque et le meta-modèle holonique, on définit indirectement les relations entre ce meta-modèle et l'ensemble des meta-modèles se référant au meta-modèle holonique.

III.4. Implémentation des mappings dans MEGA

Dans cette section, nous abordons l'implémentation des mappings pour l'interopérabilité dans l'environnement de modélisation holonique sous ***MEGA***. Ces mappings assurent l'échange d'information entre applications manipulant des données relatives au produit. Selon l'approche holonique pour une interopérabilité orientée produit, le meta-modèle holonique joue le rôle de modèle de référence pour les échanges d'informations de produits entre les différentes applications. Dans ce qui suit, nous montrons l'utilisation des mappings relatifs au meta-modèle holonique, pour l'exportation des modèles réalisés dans l'environnement de modélisation holonique implémenté sous ***MEGA***, en vue de leur réutilisation dans d'autres applications. L'utilisation de ces mappings dans l'environnement ***MEGA*** est doit permettre l'exportation des modèles holoniques depuis ***MEGA***, tout en permettant la réutilisation du résultat dans d'autres applications utilisant d'autres meta-modèles (*cf.* Figure 52). La fonctionnalité de la traduction des modèles s'ajoute ainsi aux autres fonctionnalité ***MEGA*** pour permettre ainsi la génération de modèles dans les bons formats (UEML, IEC 62264, ou autre) exploitables ainsi par des applications externes. Pour implémenter les mappings du meta-modèle holonique en d'autres meta-modèles, nous commençons tout d'abord par établir un format d'extraction des données depuis un modèle basé sur le meta-modèle holonique

réalisé sous MEGA. Nous avons choisi donc choisi le formalisme XML (XML 1998) pour spécifier la structure des documents échangés.

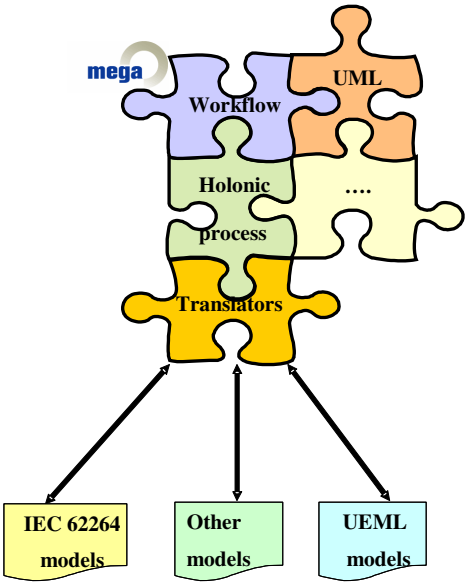


Figure 52. Mécanisme de traduction basé sur les mappings

MEGA offre la possibilité de construire des fichiers décrivant les objets définis dans un environnement MEGA avec une structure spécifique définie auparavant. Le schéma de construction des fichiers générés est formalisé dans ***MEGA*** par des *descriptions*. Nous avons ainsi défini la structure permettant de représenter un modèle fait sous ***MEGA*** à l'aide du meta-modèle holonique. Le résultat étant représenté par fichier XML dont la structure est exprimée sous forme d'un schéma DTD (Walmsley 2001) dans la figure 55.

Les mappings identifiés dans la section III du chapitre précédent sont utilisés pour construire les règles de transformations permettant ainsi de transformer un document XML représentant une instanciation du meta-modèle holonique, en un document XML valide des instanciation des meta-modèles cibles (UEML ou IEC 62264). Ces règles de transformations sont formalisées grâce au langage XSLT (XSLT 1999) permettant de décrire dans une syntaxe particulière les transformations nécessaires pour passer d'une structure XML d'origine à la structure cible. Ainsi donc, nous dérivons un ensemble de règles de transformation XSLT à partir de chacun des mappings définis antérieurement dans le chapitre 3, voir l'exemple de la figure 53.

```
<!ELEMENT Diagram (ProcessList, FlowList, HolonList)>
      <!ATTLIST Diagramme name CDATA >
<!ELEMENT Flow (ProcessusOut, ProcessusIn, Holons?)>
      <!ATTLIST Flow Id CDATA >
      <!ATTLIST Flow name CDATA >
<!ELEMENT Flow_List ( Flow+) >
<!ELEMENT Holon (attributes, properties, components)>
      <!ATTLIST Holon Name CDATA >
      <!ATTLIST Holon Classe CDATA  >
      <!ATTLIST Holon Complexe ( false | true) >
      <!ATTLIST Holon Etat CDATA >
      <!ATTLIST Holon Id CDATA >
<!ELEMENT Holon_List ( Holon+) >
<!ELEMENT Holons ( Holon+) >
<!ELEMENT Process EMPTY >
      <!ATTLIST Process Id NMTOKEN >
      <!ATTLIST Process Name CDATA >
      <!ATTLIST Process Type CDATA >
<!ELEMENT Process_List ( Process+) >
<!ELEMENT ProcessusIn ( #PCDATA) >
<!ELEMENT ProcessusOut ( #PCDATA) >
<!ELEMENT att EMPTY >
      <!ATTLIST att name CDATA >
      <!ATTLIST att type CDATA >
<!ELEMENT attributes ( att+) >
<!ELEMENT components ( Holon+) >
<!ELEMENT prop EMPTY >
      <!ATTLIST prop name CDATA >
<!ELEMENT properties ( prop+) >
```

Figure 53. Extrait du schéma DTD des fichiers XML générés par MEGA

La définition des règles XSLT nécessite l'identification des schémas XML d'origine et de destination. Dans le cas UEML, (Berio et al. 2003) définit une structure XML pour le langage UEML sous forme d'un schéma XSD (Walmsley 2001), la norme IEC 62264 possède quant à elle plusieurs implémentations sous forme de schéma XML la plus répandue est B2MML (B2MML 2003) qui définit les schémas XSD de l'ensemble des modèles de données de la norme. Pour faciliter l'utilisation des transformations XSLT, nous avons développés une petite application en JAVA présentant une interface graphique simplifiée (*cf.* Figure 54).

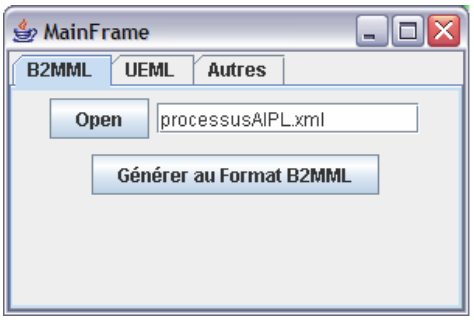


Figure 54. Interface graphique pour l'application des règles de transformations XSLT.

Cette application permet de spécifier le fichier XML contenant les données dans le format holonique, puis par la suite permettant de choisi le format destination UEML, ou IEC 62264, ou autres. Le résultat est présenté sous forme d'un fichier XML correspondant au format

choisi. La spécification des règles de transformations XSLT permet ainsi d'automatiser la transformation des modèles définis sous l'environnement holonique en des modèles compréhensibles et réutilisables dans des applications ayant leur propre meta-modèle.

```
<xsl:template match="Holon">
        <MaterialsubLot>
                <ID>    <xsl:value-of select="@Id"/>  </ID>
                <Description>
                        <xsl:value-of select="@Name"/>
                </Description>
                <Status >
                        <xsl:attribute name="composite">
                                <xsl:value-of select="@Composite"/>
                        </xsl:attribute>
                </Status>
                <xsl:apply-templates/>
                </MaterialsubLot>
</xsl:template>
<xsl:template match="attributes">
        <xsl:comment>
        <xsl:text> Attributes list </xsl:text>
        </xsl:comment>
        <Any>
                <xsl:for-each select="att">
                        <xsl:element name="att">
                                <xsl:value-of select="@name"/>
                        </xsl:element>
                </xsl:for-each>
        </Any>
</xsl:template>
<xsl:template match="properties">
        <xsl:comment>
                <xsl:text> Properties list </xsl:text>
        </xsl:comment>
        <Any>
                <xsl:for-each select="prop">
                        <xsl:element name="prop">
                                <xsl:value-of select="@name"/>
                        </xsl:element>
                </xsl:for-each>
        </Any>
</xsl:template>
<xsl:template match="components">
        <xsl:comment>
                <xsl:text> references to components </xsl:text>
        </xsl:comment>
                <xsl:for-each select="Holon">
                <MaterialsubLot>
                        <ID>    <xsl:value-of select="@Id"/>  </ID>
                </MaterialsubLot>
                </xsl:for-each>
</xsl:template>
        <xsl:template match="Partie_Info">
        <xsl:comment>
                <xsl:text> references to Partie_Info </xsl:text>
        </xsl:comment>
                <xsl:for-each select="Holon">
                <MaterialsubLot>
                        <ID>    <xsl:value-of select="@Id"/>  </ID>
                </MaterialsubLot>
                </xsl:for-each>
        </xsl:template>
```

Figure 55. Exemple de règles XSLT réalisées à partir des mappings vers l'IEC 62264.

IV. L'approche holonique et les mappings dans le cadre Zachman

Dans cette section, nous poursuivons l'analyse de l'approche de modélisation holonique selon le cadre Zachman. Cette analyse s'intéresse à l'échange des données entre les modèles résultant de la modélisation holonique de l'entreprise et les modèles relatifs aux applications réelles de l'entreprise : ERP, MES ou autres. En référence au cadre de modélisation Zachman, nous spécifions une nouvelle étape permettant de relier les étapes de conceptualisation propre à l'approche holonique (***étape 0*** et ***étape 1*** de la figure 34) aux couches basses du cadre Zachman représentant les niveaux relatifs à l'implémentation des applications ou systèmes préexistants dans l'entreprise. L'objectif étant de s'approcher le plus possible des couches "*Functioning System*" et "*Detailed Representation*" du cadre Zachman. En effet, les applications et systèmes de l'entreprise se situent dans les deux dernières lignes du cadre Zachman. L'étude et la réalisation des mappings relatifs à l'approche holonique de modélisation, permettent ainsi de faire le lien entre les modèles holoniques se situant à un niveau plus conceptuel aux représentations détaillées des données telles qu'elles sont implémentées dans chacun des systèmes de l'entreprise. Pour passer des étapes conceptuelles (***étapes 0*** et ***1***) à l'implémentation, le premier point à traiter concerne le choix du schéma logique dans lequel seront exportés les descriptions des produits et les objets représentés par des holons dans les diagrammes holoniques réalisés dans l'***étape 1*** (par exemple: UEML ou IEC 62264). Ce schéma logique correspond au modèle du système cible. Cette première tâche représente la case se trouvant à l'intersection du niveau logique et la colonne « *What* » dans le cadre Zachman.

Ensuite, les données seront transformées grâce au mappings prédéfinis puis intégrées selon l'implémentation du système cible (fichiers XML, bases de données relationnelles, bases de données objets ou autres technologies.). Dans Zachman, la phase d'implémentation des données correspond à la ligne « *Technology Model* » et la colonne « *What* ».

Nous définissons ainsi l'***étape 2*** du déroulement de l'approche holonique selon le cadre de modélisation Zachman, comme étant l'ensemble de la démarche relative à l'étude des correspondances sémantiques et la spécification des mappings entre le meta-modèle holonique et les modèles spécifiques aux applications d'entreprise. Dans la figure qui suit nous montrons la correspondance entre cette étape et les différents points de vues couverts par le cadre de modélisation.

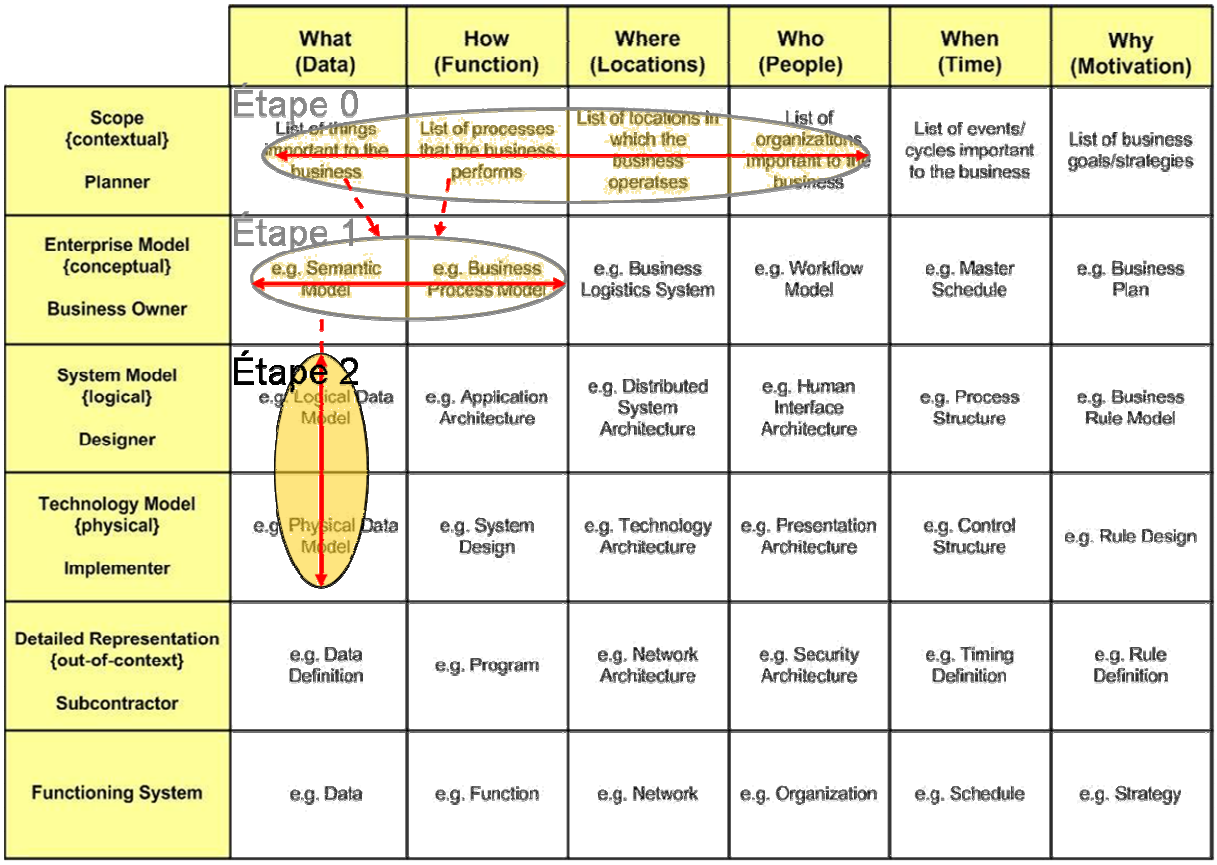


Figure 56. Positionnement de l'approche d'interopérabilité sur le cadre de modélisation Zachman

V. Mise en oeuvre

Pour montrer l'utilisation, à travers le cadre Zachman, des mappings et de l'approche holonique dans un contexte de production réel, nous illustrons notre approche sur un cas pratique issu de l'environnement AIP Lorrain.

Pour la gestion et le contrôle du procédé de production au sein de l'atelier, l'AIPL dispose d'un ensemble d'applications, que nous représentons dans la figure 57.

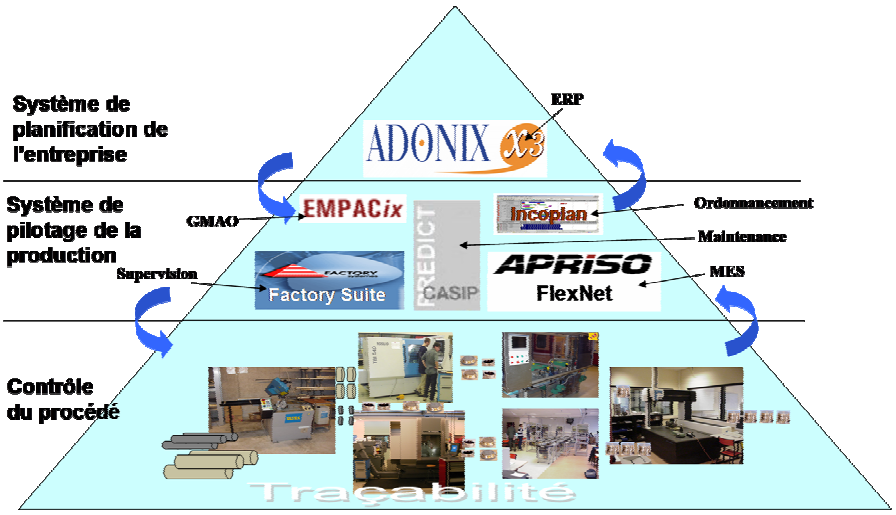


Figure 57. Intégration en pyramide des applications de l'AIPL.

Nous classons ces applications en deux sous-groupes, les applications du pôle « manufacturing » (Systèmes de pilotage de la production) et les applications du pôle « business » (Systèmes de planification de l'entreprise). L'objectif de cette section est de montrer l'utilisation de l'approche holonique pour l'interopérabilité orientée produit dans un contexte B2M. Dans le premier pôle, nous regroupons l'ensemble des systèmes de pilotage de la production (Factory Suite[19], Flexnet[20], EMPACix[21], Casip[22], Incoplan[23]) ; dans le second

[19] Factory Systems, *http://www.factory-syst.fr*
[20] Apriso-Flexnet, *http://www.apriso.com*
[21] INDUS International – EMPACix, *http://www.indus.fr*
[22] PREDICT-CASIP, *http://www.predict.fr*
[23] Incoplan, *http://www.incotec.fr/incoplan/incoplan_main.htm*

pôle nous plaçons l'ERP SAGE-ADONIX[24]. La figure 58 illustre les applications de l'AIPL mises en œuvre dans chacun des deux pôles.

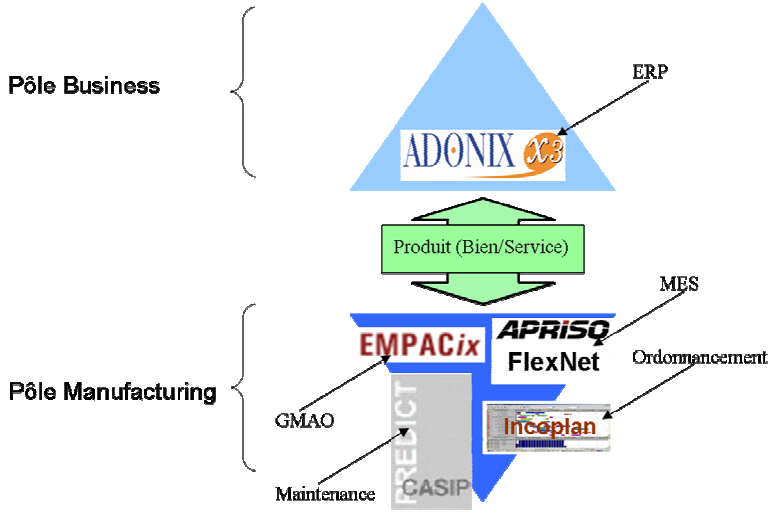


Figure 58. La séparation des pôles « manufacturing » et « business » appliquée au contexte AIPL.

Dans ce contexte, nous tentons de montrer l'apport de notre approche pour l'interopérabilité orientée produit. Nous allons appliquer l'approche holonique à notre approche étape par étape :

- ***Etape 0 : Analyse du contexte et identification des différentes entités mises en œuvre.***

L'objectif de cette étape est d'identifier l'ensemble des entités importantes pour le reste de la modélisation, ces entités peuvent être des acteurs humains, des sites de production, des processus ou des produits. Dans le cadre de l'AIP, nous avons identifié l'ensemble des entités et objets qui nous semblent pertinente pour la réalisation de notre objectif, mettre en pratique l'interopérabilité orientée produit de l'approche holonique.

A titre d'exemple, voici un extrait des éléments qui ressortent lors de l'analyse de l'environnement AIPL:

<u>Objets de l'entreprise :</u>

Produits : *Produit 01,09 ; Produit 60,88,09 ; Produit 60,88,11,10….*

Pièces : *Pièce 01, Pièce 10, Pièce 11, Pièce 09, Pièce 88, Pièce 60….*

Autres : *Ordre de Fabrication, Commande client, Commande de matières premières.*

[24] ADONIX, *http://www.sageadonix.fr*

Processus :

Tronçonnage des barres brutes ; Découpage des plaques (galvanisées ou aimantées) en rondelles ; Tournage des pièces ; Collage des pièces ; Assemblage des produits finaux, stockage, livraison, etc.

Chacun de ces processus se décompose lui-même en un ensemble de sous processus de plus fine granularité. Par exemple le processus d'assemblage se décompose en trois sous processus : préparation de l'assemblage, l'assemblage proprement dit puis le contrôle visuel de l'assemblage.

Sites :

IUT Brabois : c'est dans ce site que sont découpée les plaques galvanisées, utilisées pour la fabrications des pièces de l'AIPL

Le site AIPL : ici se déroule le reste du la fabrication des produits et des pièces.

Acteurs : *Le Client, Administration AIPL, opérateurs, …*

- ***Etape 1 : Construction des modèles holoniques correspondant au contexte spécifié par l'étape 0.***

Dans cette étape, les premiers modèles mettant en œuvre les différents éléments listés dans ***l'étape 0***. L'objectif étant de définir l'ensemble des holons échangés dans l'environnement étudié à partir de l'analyse réalisée dans l'***étape 0***. Pour procéder, nous commençons tout d'abord par construire un modèle de processus où apparaissent les différents processus, sites, acteurs et bien sur les différents flux circulants entre ces entités.

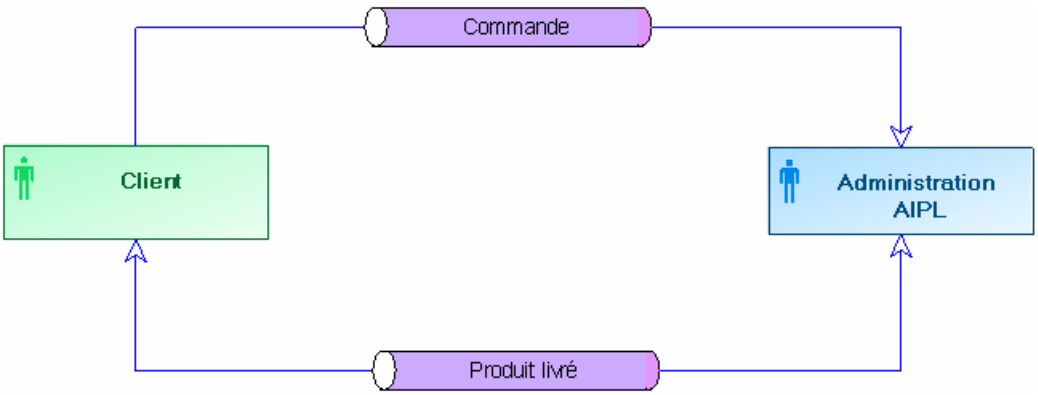


Figure 59. Point de départ pour la modélisation de l'AIPL.

Dans la figure 59, nous montrons le diagramme préliminaire de la modélisation holonique à l'AIPL. Dans ce modèle, nous décrivons le premier échange entre l'administration de l'AIPL et le client. En effet, dans un système contrôlé par le produit, la demande du produit est souvent l'élément déclencheur de l'ensemble de processus qui en découle. A cette demande correspond un retour pour satisfaire la demande, dans notre cas la demande correspond à la commande d'un ensemble de produits de la part du client, le retour correspond ainsi à la livraison de l'ensemble de produits commandés sous forme d'un colis.

Cependant, il faut noter que l'ensemble ses échanges (ou flux) n'ont pas la même nature, et n'ont donc pas la même incidence sur leur récepteur. En effet, dans notre exemple le flux « Commande » correspond à un stimuli qui contient un ensemble de données décrivant la commande, le flux « Produit livré » correspond quant à lui à un flux d'objets physiques et des informations relatives à ces produits. Les flux doivent ainsi pouvoir être différencier non seulement par rapport à leur contenu mais également par rapport à leur rôle (stimuli, événements, savoir faire, pouvoir faire, devoir faire, etc…).

A partir de l'échange entre le client et l'AIPL, notre approche de modélisation incite à remonter le cycle de production des produits livrés qui se situent eux à la fin de la chaîne de production dans l'environnement AIPL. Pour essayer de remonter l'historique lié à la production des produits finis dans cet environnement. Cette analyse permet ainsi de couvrir l'ensemble des flux et processus, faisant partie dans la boucle de production (Commande / Livraison) instaurée entre l'AIPL et le client. A la fin de cette analyse, l'ensemble des flux échangés doit former une boucle allant de la réception de la commande formulée par le client, jusqu'à la livraison des produits finaux (*cf*. Figure 60).

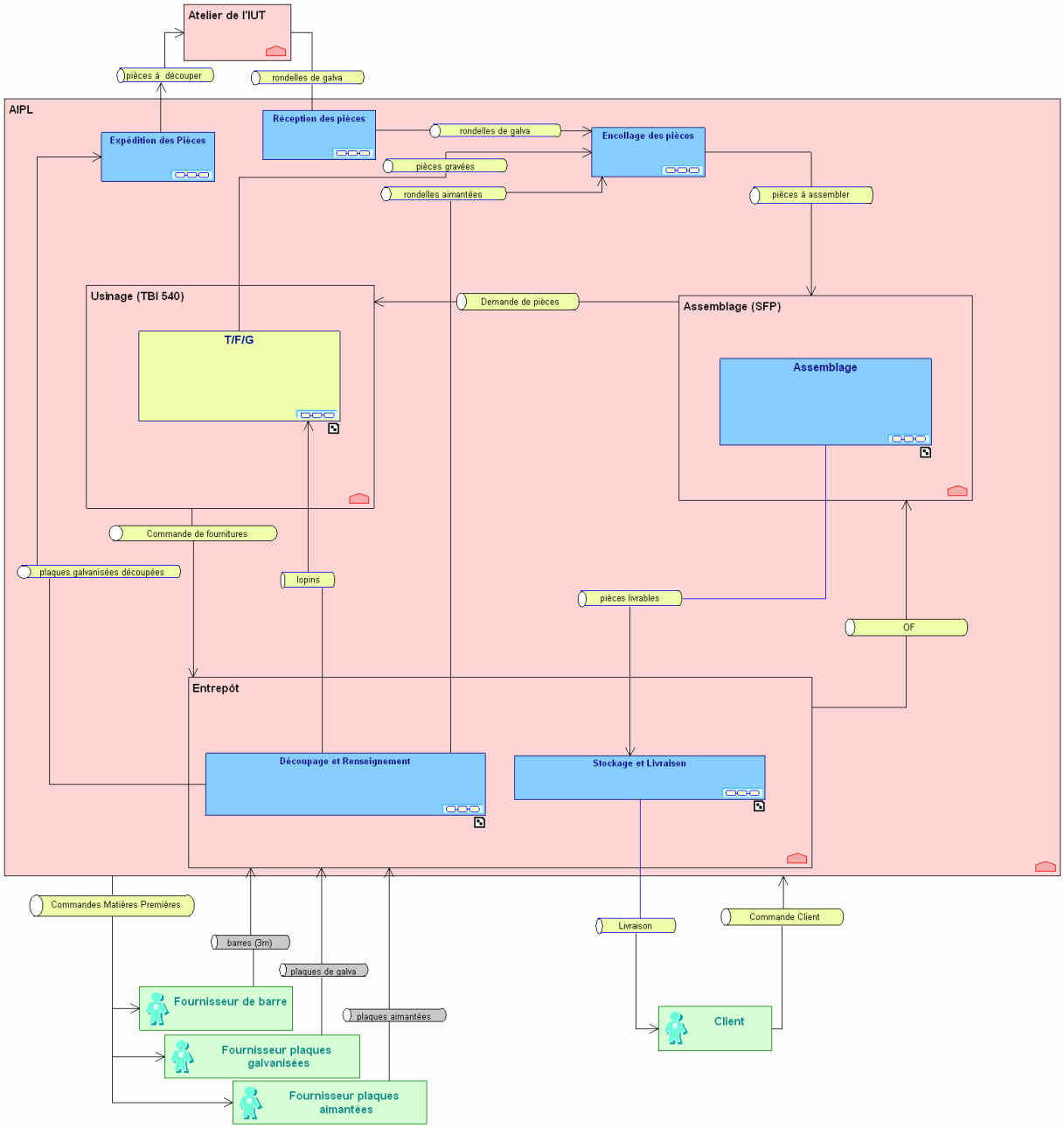


Figure 60. Vue générale du processus de fabrication de l'AIPL

Par raffinement, pour chacun des flux nous précisons le type : physique, informationnel ou holonique. Puis par la suite, nous faisons apparaître les différents holons ou objets physiques brutes qui y circulent. Ces holons, objets physiques ou objets informationnels sont dérivés des objets de l'entreprise qui ont été identifiés dans ***l'étape 0***.

Pour chacun des flux nous spécifions l'ensemble des holons qu'il véhicule, chaque holon est aussi relié aux autres holons impliqués dans sa fabrication. Ces holon pour l'instant font

référence à la partie physique des produits, pièces et autres objets véhiculés dans l'environnement AIPL (*cf.* exemple Figure 61).

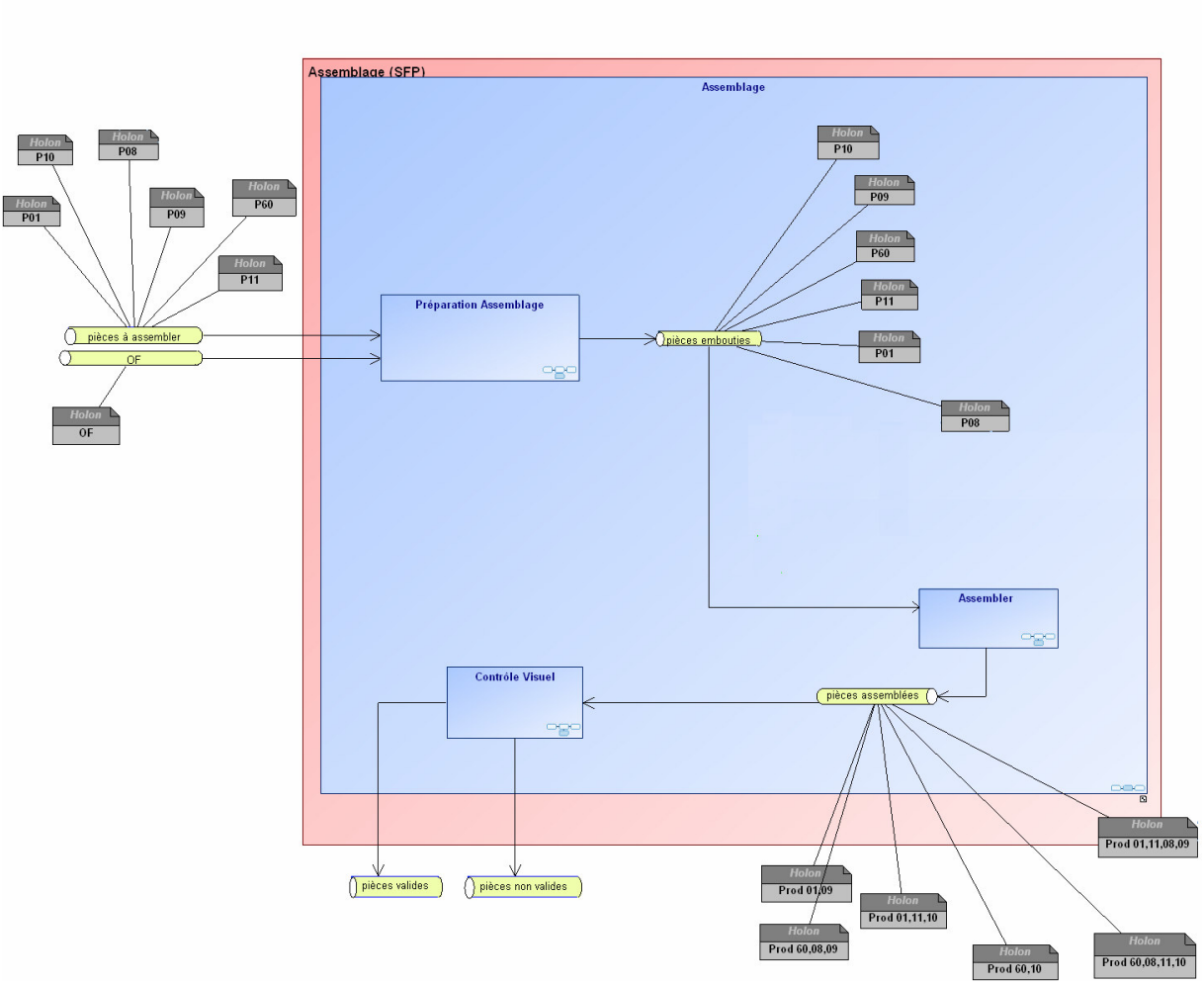


Figure 61. Exemple de flux d'holons échangés sur la cellule d'assemblage des produits

Le raffinement se poursuit par la définition des différentes informations (attributs ou propriétés) associés à chacun des holons identifiés. Chaque holon est ainsi décrit par un ensemble d'attributs ou propriétés. Ces attributs et propriétés sont aussi utilisés pour définir les interfaces informationnelles des processus mis en œuvre pour la fabrication du produit, la relation entre processus et informations en entrées puis en sorties est ainsi spécifié au terme de ***l'étape1.***

Pour l'identification des propriétés et attributs, à chaque processus nous identifions l'ensemble des informations relatives aux holons et qui sont donc pertinentes pour le processus en question. La différenciation attribut/propriété se fait sur la nature de

l'information, s'il s'agit d'une information intrinsèque à la partie physique du produit elle sera représentée par un attribut, sinon elle sera représentée par une propriété. Les attributs ne peuvent être modifiés que par des processus physique, les propriétés quant à elles sont manipulées et mises à jour par des traitements informationnels.

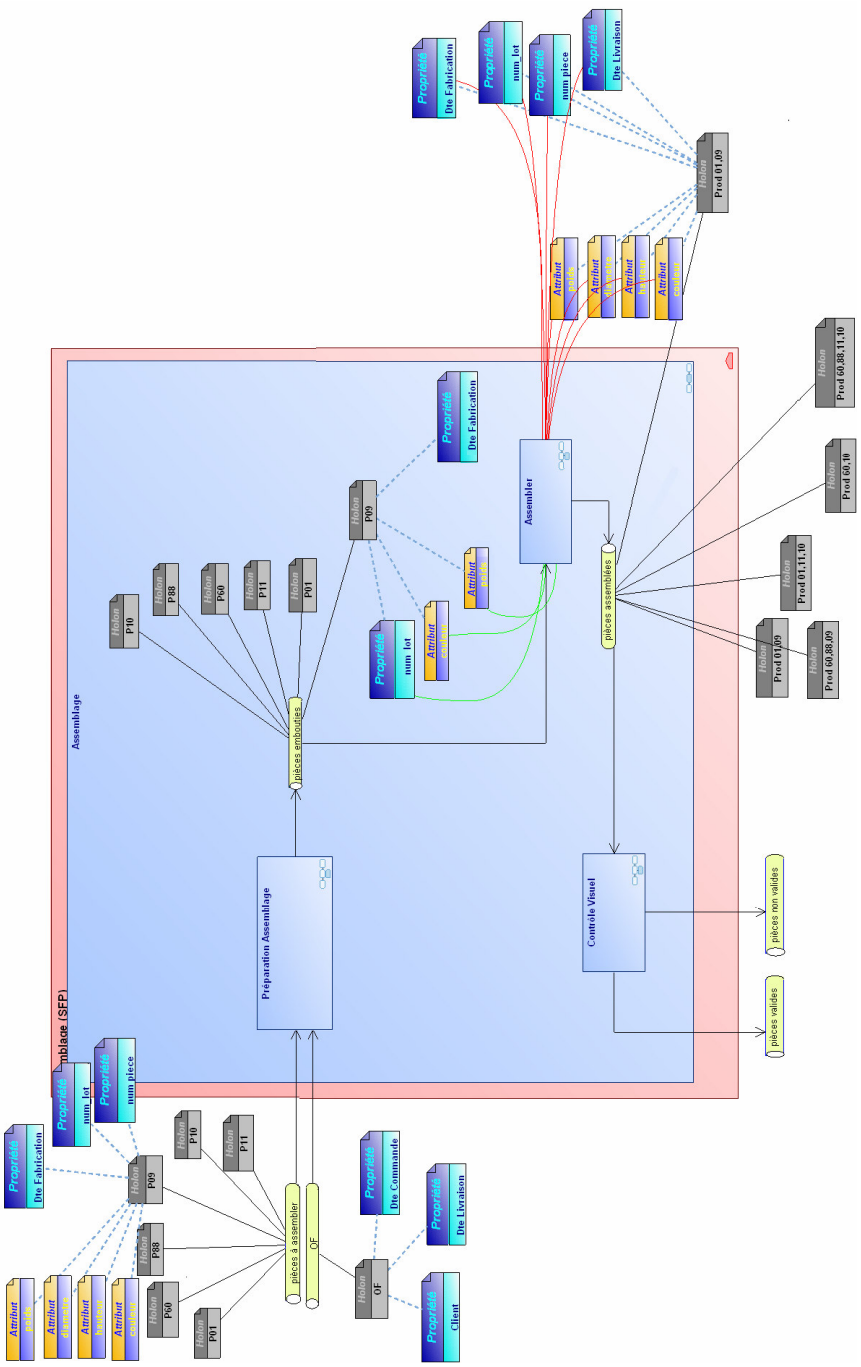


Figure 62. Exemple de l'identification des informations (propriété/attribut) relatives aux produits échangés dans l'environnement AIPL.

Du point de vue du produit, le résultat est une description détaillée de la composition du produit représentée sous forme d'holon, ainsi que des attributs « physiques » du produit et des propriétés « informationnelles » qui lui sont associés (*cf.* l'exemple de la figure 63).

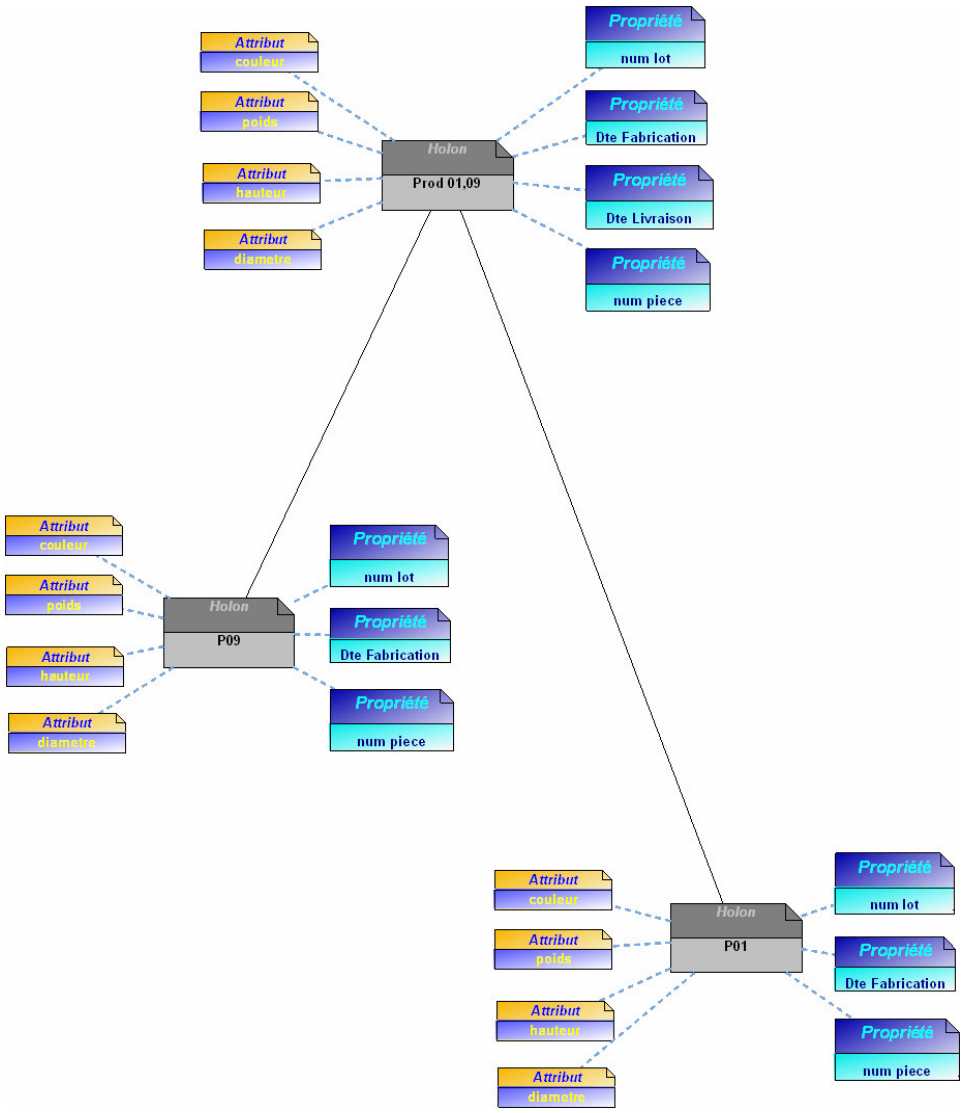


Figure 63. Exemple de représentation holonique du produit « Prod 01,09 ».

Cette représentation permet, d'une part, de configurer les systèmes d'entreprise (MES ou ERP ou autre) en prenant en compte la définition holonique du produit « Prod 01,09 » représentée ci-dessus comme étant le modèle suffisant et nécessaire pour représenter des produits de type « Prod 01,09 ». D'autre part, l'instanciation de cette représentation en utilisant les données propres à une instance du produit « Prod 01,09 » peut jouer le rôle d'un pivot central pour la

synchronisation des différentes vues du produit implémentées par l'ensemble des applications manipulant une instance donnée de ce produit.

- ***Etape 2 : Mise en correspondance des modèles respectifs de chacune des applications.***

L'objectif de cette étape est de réaliser le passage du modèle holonique du produit, décrivant l'ensemble des informations relatives à un produit donné (*cf.* Figure 63) à ***l'étape 1*** vers les modèles de niveau technique implémentés dans chacune des applications de l'environnement AIPL. Dans cet environnement, les applications manipulent différentes représentations du produit, à titre d'exemple nous utiliserons la représentation du produit implémentée dans l'ERP « ADONIX », et celle implémentée dans le MES « FLEXNET ». Ces représentations ont pu être restituées par un processus de rétro ingénierie appliquée directement sur les bases de données utilisées par ses deux progiciels. Nous avons identifié quelques uns des éléments relatifs au produit faisant à la fois partie de modèle de données implémenté dans ADONIX et celui implémenté dans FLEXNET, à titre d'exemples, nous pouvons citer :

- La description des caractéristiques du produit;
    - Adonix: *ITMMASTER; ITMFACILIT ;*
    - Flexnet: *PRODUCT* ; *PRODUCT_DIMENSON ;*
- L'identification et la traçabilité du produit physique au niveau des stocks;
    - Adonix: *STOLOT* ; *STOCK* ;
    - Flexnet: *INVENTORY*; *INVENTORY_SERIAL_NO* ; *NVENTORY_COUNT* ;
- La gestion des informations de routage des produits dans le processus de production;
    - Adonix: *ROUTING; TABROUALT ;*
    - Flexnet: *PRODUCT_ROUTING ;*
- Les informations relatives au prix de vente du produit et les conditionnements de vente.
    - Adonix: *ITM_SALES ;*
    - Flexnet: *PACKAGING_INSTR_DETAIL* ; *PACKAGING_INSTR_USAGE ;*

Dans la figure 64 et la figure 65, nous représentons des extraits des modèles de données correspondant à la représentation du produit respectivement dans FLEXNET et dans ADONIX.

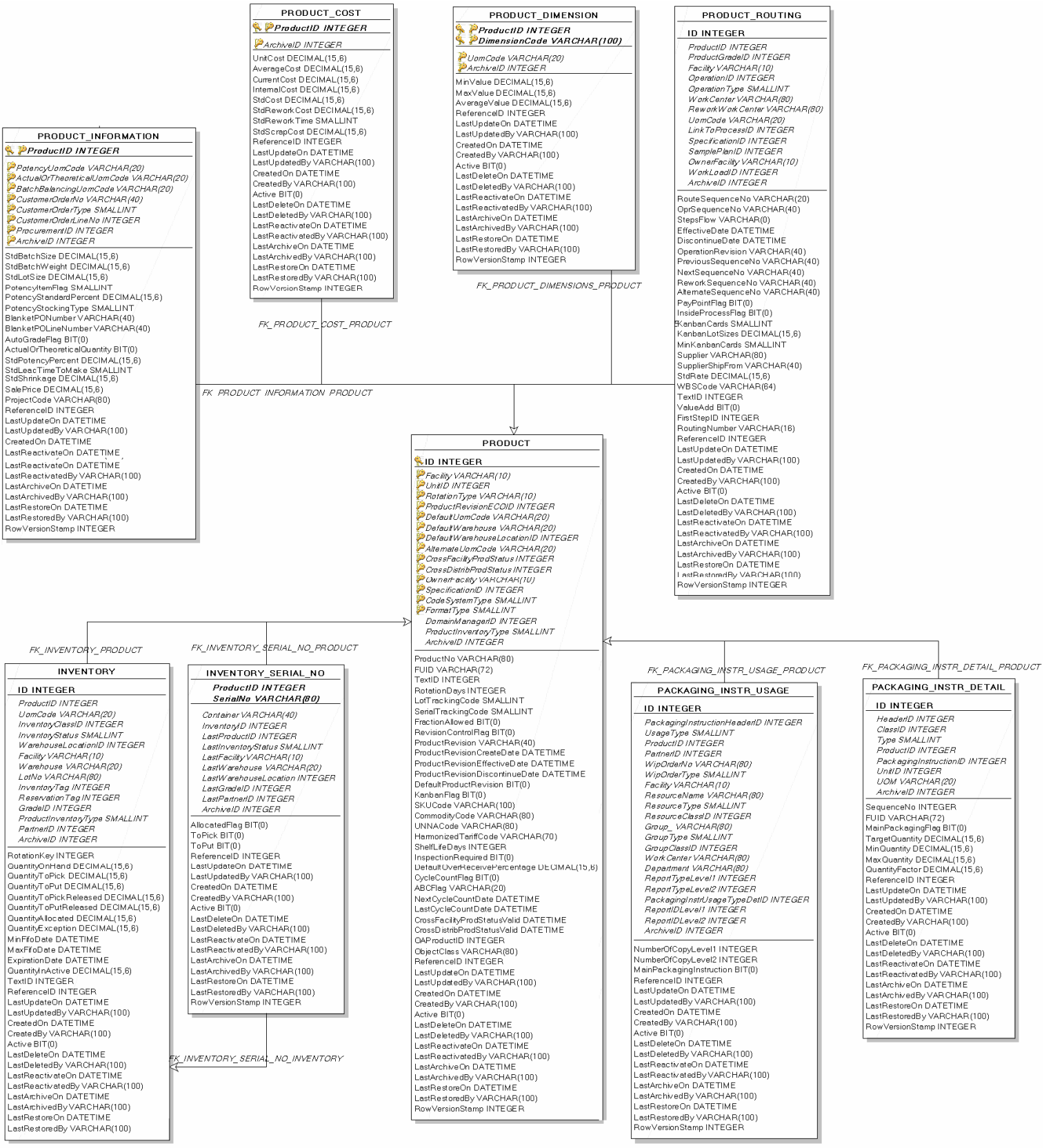


Figure 64. Extrait du modèle de données FLEXNET pour la représentation du produit.

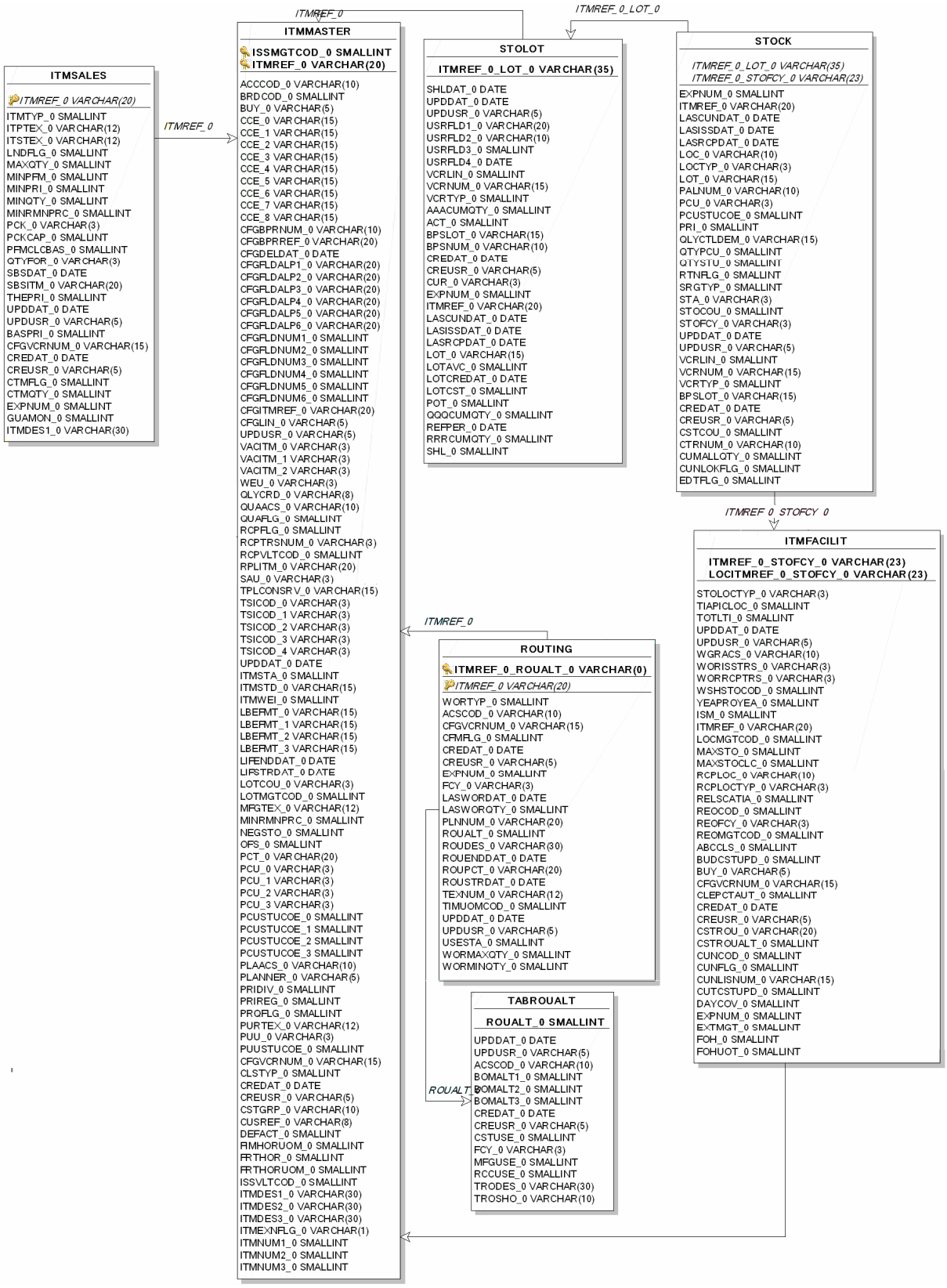


Figure 65. Extrait du modèle de données ADONIX pour la représentation du produit.

La configuration d'un ERP ou d'un MES est précédée par une phase d'ingénierie visant à modéliser l'existant de l'environnement dans lequel le progiciel (ERP, MES ou autre) aura à évoluer. Cette modélisation vise à spécifier les éléments que le MES ou ERP devra prendre en compte, et surtout la façon dont il va représenter ces éléments. Dans notre cas, nous supposons qu'en phase d'ingénierie, les données relatives au niveau manufacturing destinées ainsi au MES sont modélisées par le biais de la norme IEC 62264 et que les données du niveau business destinées à l'ERP sont quant à elles modélisées grâce à UEML. Les modèles obtenus permettront de configurer correctement les ERP et MES de l'entreprise. Nous avons dans les ***étapes 0*** et ***1*** de notre approche, que la modélisation holonique offre une méthode et un outillage permettant la modélisation des produits de l'entreprise sous forme d'holons. Cette modélisation permet de synthétiser, pour un produit donné, l'ensemble des informations relatives à ce produit ainsi que l'arborescence structurelle du produit permettant la description de l'ensemble des matières premières et produits intermédiaires utilisés dans la composition du produit en question.

L'étape 2 quant à elle vise à transformer les modèles de données issus de l'approche holonique en éléments réutilisables que ce soit pour la configuration d'un ERP ou d'un MES (ADONIX et FLEXNET par exemple). Pour cela les modèles de données issus de la modélisation holonique doivent être traduits en IEC 62264 d'un coté pour la configuration du MES, et en modèles de UEML de l'autre pour la configuration de l'ERP. Ces transformations et traductions depuis les modèles holoniques vers l'IEC 62264 puis UEML de l'autre sont réalisés grâce au mappings définis précédemment. La configuration du MES à partir des modèles IEC 62264, et de l'ERP à partir de modèles UEML ne fait pas partie du travail présenté dans cette thèse.

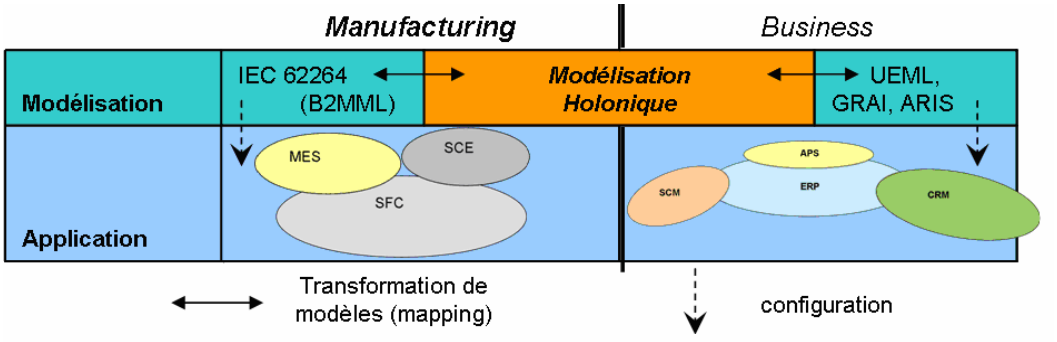


Figure 66. La modélisation holonique pour l'ingénierie B2M du produit.

La figure 66 montre l'apport de la modélisation holonique pour la configuration de la partie relative au produit des progiciels d'entreprise. Les exemples de ce chapitre montrent que les modèles issus de l'approche holonique pour la modélisation des produits de l'entreprise peuvent interopérer avec d'autres modèles spécifiques à différents domaines de l'entreprise que ce soient des modèles du « Manufacturing » ou du domaine « Business ». La modélisation holonique permet ainsi la modélisation du produit à l'intersection des deux pôles en offrant la possibilité à l'utilisateur de se focaliser sur des applications du pôle « Manufacturing » ou bien des applications du pôle « Business ». Elle fait ainsi partie des outils pour l'ingénierie de l'interopérabilité « B2M » en ce qui concerne le produit.

Les mappings que nous avons réalisés avec l'IEC62264 et UEML seront utilisés comme passerelles nous permettant, d'une part pour l'IEC 62264 de transformer les modèles holoniques en modèles réutilisables au niveau manufacturing, et d'autre part pour UEML d'établir le lien avec le niveau business. Le résultat peut par la suite être utilisé en phase d'ingénierie pour la configuration des applications ADONIX et FLEXNET afin de les adapter à l'environnement AIPL, ou en phase d'exploitation pour permettre l'échange de données entre les instances de modèles holoniques maintenues cohérentes avec la réalité des produits et les différentes représentations du produits au sein des différents progiciels. Ainsi, le modèle holonique réalisé en phase de modélisation joue un double rôle. En effet, il est utilisable, à la fois, à la conception des produits dans le but de spécifier l'ensemble des informations relatives à un produit donné, et en phase d'exploitation, puisqu'un modèle holonique doit être instancié en utilisant les données propres aux instances de produits et devenir ainsi la vue holonique du produit tel qu'il doit être vu par l'ensemble des systèmes et garantir ainsi la cohérence entre les différentes vues du produit dans ces différents systèmes.

Dans ce qui suit nous illustrerons nos propos en utilisant les mappings définis entre le meta-modèle holonique et l'IEC 62264, plus précisément le « *Material Model* » de l'IEC 62264. Dans un premier temps nous nous intéresserons à l'instanciation relative aux classes de produits et à la définition de ces produits correspondant à l'utilisation du modèle holonique en phase de conception. Cette première transformation permet la configuration d'un MES (ou ERP dans le cas d'UEML), en utilisant les informations relatives à la définition des classes d'objets et non les valeurs relatives aux instances des objets. Puis par la suite, nous montrerons une instanciation du « *Material Model* » prenant en compte des valeurs et

données propres à une instance précise d'un produit. La figure 67 résume l'enchaînement des différentes étapes afin de relier la représentation conceptuelle du produit sous la forme holonique aux différentes représentations du produit telles qu'elles sont implémentées dans les différents systèmes ADONIX et FLEXNET à titre d'exemple.

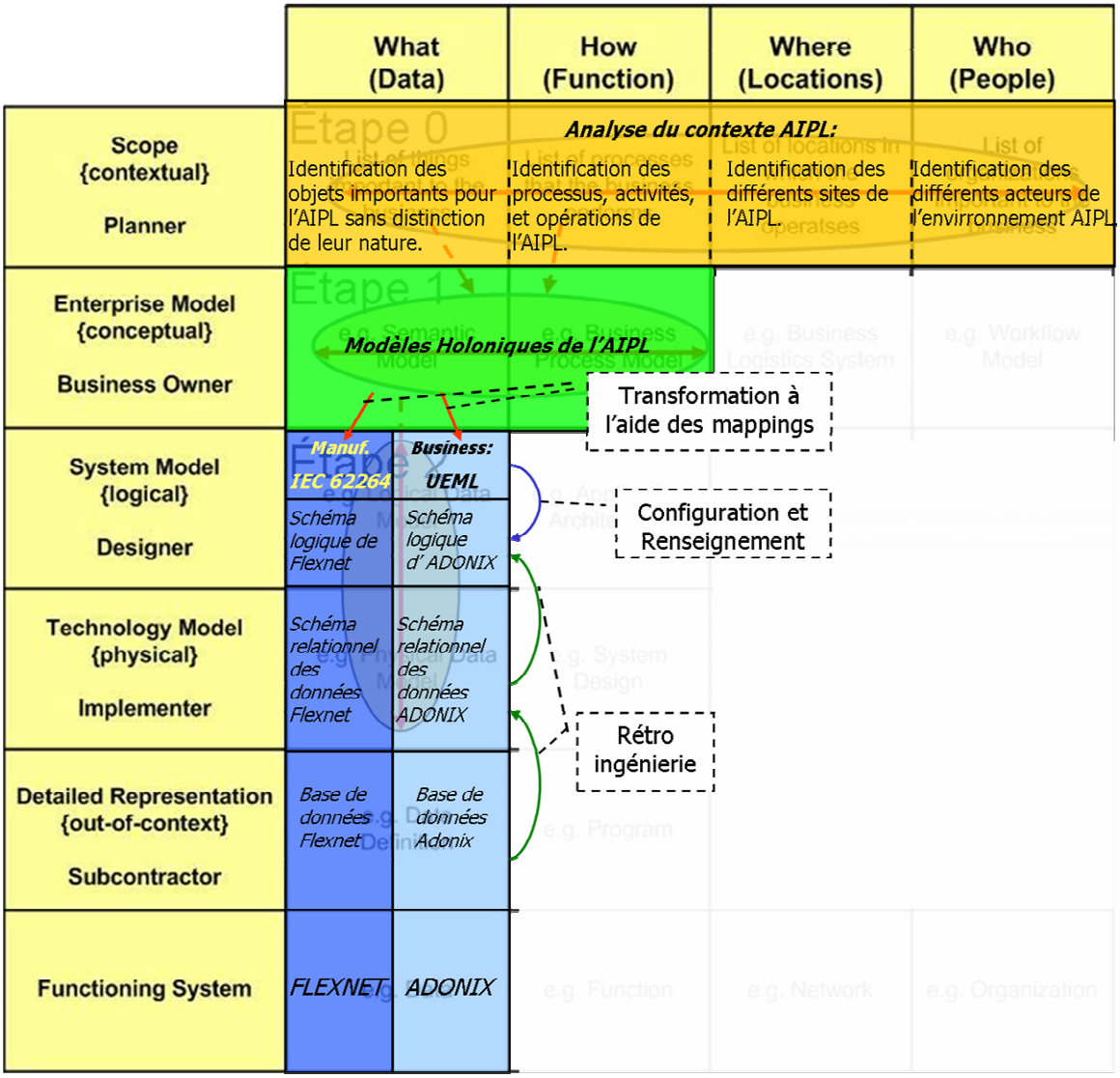


Figure 67. Illustration des différentes étapes suivies lors de l'application de l'approche holonique dans le cadre de l'environnement AIPL.

Dans la figure 68, nous illustrons le déroulement de l'ensemble de l'approche. En effet, la méthode holonique offre à l'ingénieur un outil lui permettant de modéliser le produit et ses interactions dans le système en question. Le guide construit sur la base du cadre Zachman constitue une aide pour l'utilisation de l'ensemble de la méthode. Cependant, la définition des

différents mappings est une tâche que seul cette ingénieur pourra effectuer, car ces mappings dépendent de l'ensemble des choix de modélisation (langages ou outils) et l'ensemble des contraintes liées à la réalité des systèmes mises en œuvre. Ces mappings permettront à l'Ingénierie de fournir les bons paramètres pour la configuration des systèmes utilisés, puis lors de la phase d'exploitation permettront l'échange d'informations relatives au produit entre les différents systèmes. Ces échanges se basent sur une représentation holonique du produit, maintenues cohérente à la réalité physique de ce produit tout au long de son cycle de vie.

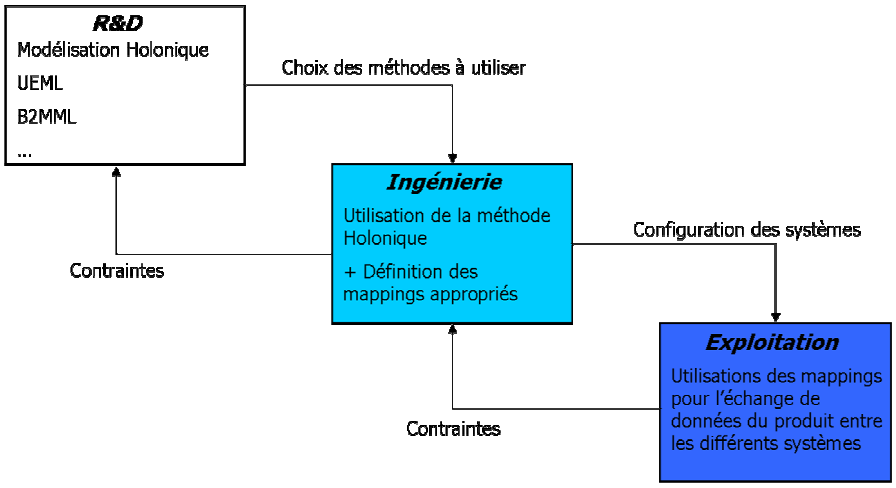


Figure 68. Illustration du déroulement de l'ensemble de la démarche à travers les différentes sections de l'entreprise.

Pendant phase d'ingénierie :

En appliquant les mappings définis dans précédemment dans ce chapitre entre le meta-modèle holonique et l'IEC 62264, nous réalisons une instanciation du modèle de « *Material Model* » à partir du modèle holonique du produit « Prod01,09 » synthétisé dans la figure 64. L'objectif de cette instanciation est de fournir la définition des classes d'objet et non les objets eux même. Il faut donc identifier les informations associées au produit qui pourraient avoir une signification par rapport à la classe même du produit :

Par exemple, pour une instance du produit « Prod 01,09 », l'attribut « poids » permet de spécifier la valeur exacte du poids de l'instance en question. Cependant, au niveau de la classe du produit, ce même attribut permet de définir l'intervalle de valeurs que peut prendre la valeur du poids d'une instance donnée. Un attribut ou une propriété, identifiés sur un produit lors du processus de modélisation, peut être vu comme un élément relatif à la classe du

produit, ou aux instances de ce produit ou aux deux en même temps. En étudiant un à un les attributs et les propriétés associés au produit « Prod01,09 », nous identifions l'ensemble des informations qui peuvent être utilisées pour la description de la classe de produit. Nous traduisons ensuite ces éléments vers le vocabulaire du modèle « *Mateial Model*» tel que le préconisent les mappings.

La figure 69 montre le résultat de la transformation, elle représente ainsi les données correspondant à une instanciation du « *MaterialModel*» de l'IEC 62264.

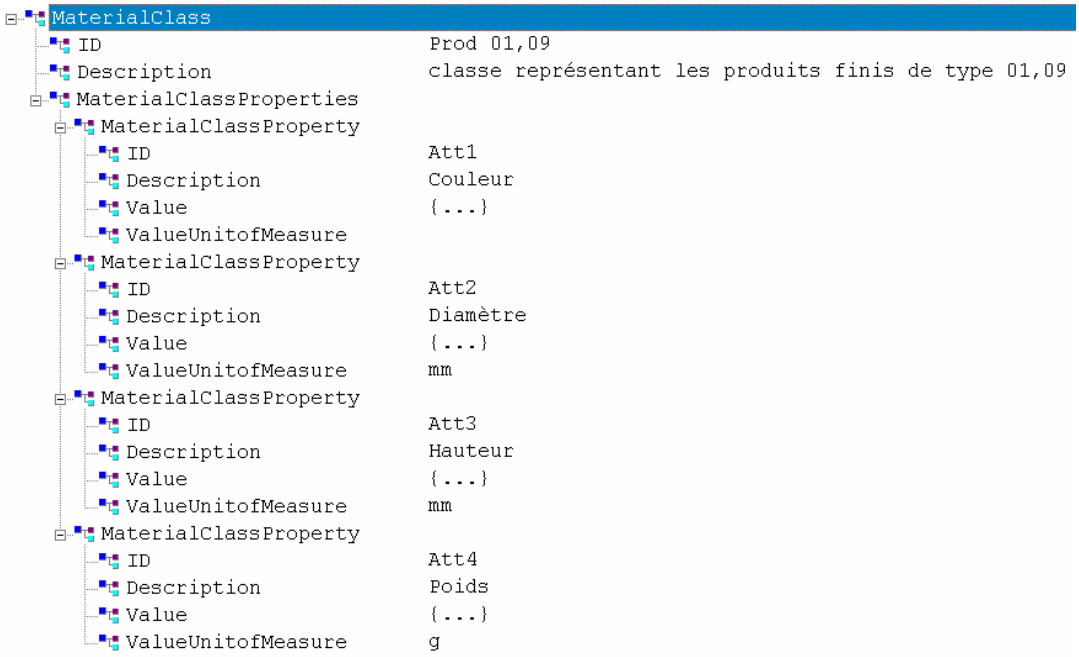


Figure 69. Instanciation des éléments *MaterialClass* et *MaterialClassProperty* correspondant à la classe du produit « Prod01,09 ».

Pendant la phase d'exploitation :

Les modèles holoniques conçus en ingénierie peuvent être instanciés en phase d'exploitation en utilisant les données cohérentes avec la réalité du produit mises à jours tout au long du cycle de vie du produit. Pour que le modèle holonique instancié puisse jouer le rôle de pivot d'échange et de synchronisation des données relatives au produit pendant l'exploitation, il est nécessaire que les transformations de modèles permettent l'échange des données instanciées relatives à un produit bien précis. Il faut donc appliquer les mappings pour obtenir une vue du produit et non une spécification de sa classe.

Dans la figure 70, nous montrons une instanciation du modèle holonique du produit «Prod 01,09» en utilisant les valeurs que prennent chacun des attributs et chacune des propriétés pour cette instance précise.

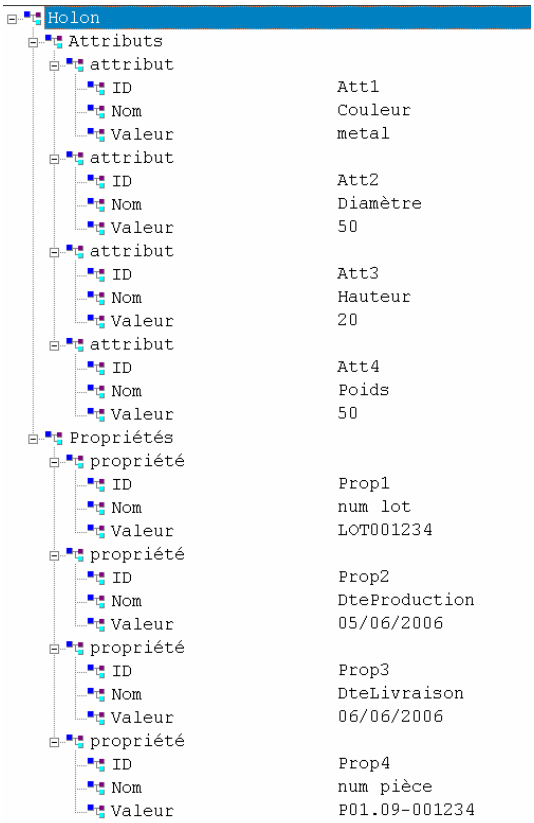


Figure 70. Instanciation d'un holon en utilisant les valeurs effectives correspondant à une instance du produit « Prod 01,09 ».

Nous allons maintenant appliquer les mappings pour passer de cette instanciation à une instanciation du « *MaterialModel*» de l'IEC 62264. Comme le préconise notre mapping, les données relatives à l'holon permettent de définir le *MaterialSublot* correspondant. Cependant, dans ce modèle il n'est pas possible d'associer des propriétés à un *MaterialSublot*, ce qui nous contraint à considérer un lot de produit, groupant les produits qui ont les mêmes valeurs pour l'ensemble de leurs attributs et propriétés. Le résultat du mapping est présenté dans les figures 71 et 72.

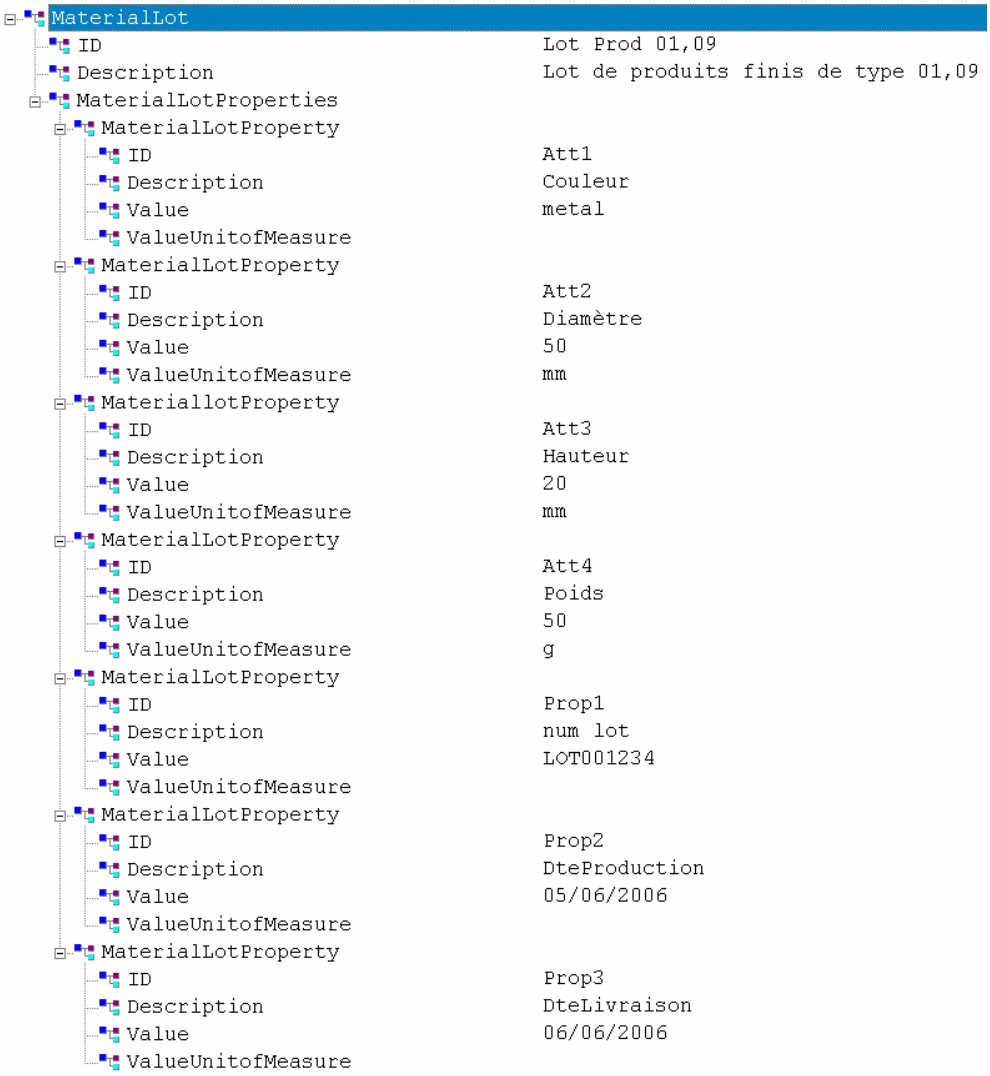


Figure 71. Instanciation d'un élément *MaterialLot* à partir des données de l'holon « Prod 01,09 ».

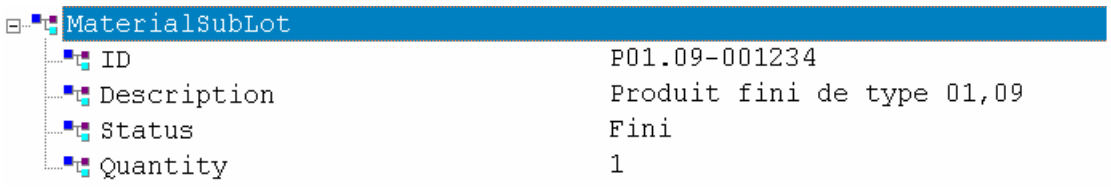


Figure 72. Instanciation d'un élément *MaterialSubLot* à partir des données de l'holon « Prod01,09 ».

Lors de la phase d'exploitation, les mappings permettent ainsi de renseigner des modèles au format de la norme IEC 62264 en utilisant des données issues d'instanciations effectives des modèles holoniques.

VI. Conclusion

Dans ce chapitre, nous avons formalisé les mappings entre meta-modèles tout en spécifiant les propriétés indispensables à la mise en place d'une interopérabilité sémantique en se basant sur les règles de transformation de modèles. La formalisation que nous proposons se base sur une description ensembliste des notions de meta-modèles et mappings entre meta-modèles. Nous avons établi une classification de l'interopérabilité basée sur les transformations de modèles. Cette classification est basée sur les propriétés mathématique, couverture et réflexivité, des mappings identifiés entre les différents modèles ou meta-modèles impliqués dans les transformations.

Nous avons ensuite proposé la mise en pratique de l'approche d'interopérabilité dirigée par les modèles pour définir des mappings pour la transformation de modèles entre le meta-modèle holonique d'un côté et deux exemples de meta-modèles issus du monde de l'entreprise :

- Le premier exemple concerne le meta-modèle de UEML, relatif au niveau organisationnel de l'entreprise.
- Le deuxième exemple concerne le meta-modèle du standard IEC 62264, plus proche du monde opérationnel de l'entreprise.

L'ensemble de l'approche holonique peut se décomposer en deux composantes distinctes :

- Une approche de modélisation permettant la conception et la spécification d'une vue unifiée du produit regroupant l'ensemble des informations qui lui sont relatives.
- Une approche de transformation de modèles pour permettre l'échange d'informations entre les modèles issus de la modélisation holonique et d'autre modèles appartenant à des applications spécifiques.

La première composante offre lors de la phase d'ingénierie une méthode permettant de décrire dans le détail le produit tel qu'il doit être vu par l'ensemble des applications. Par la suite, les techniques et règles de transformation sont utilisées pour transformer ce modèle holonique en

modèles informationnels et ainsi permettre une implémentation vers des modèles spécifiques destinés à une application ou un ensemble d'applications. Ce qui permettrait tout d'abord la configuration de ces applications de façon cohérente avec la réalité du système existant puis lors de l'exploitation, de capitaliser l'ensemble des informations du produit sous sa forme holonique puis utiliser les transformations et les mappings préétablis, pour faire de cette vision la référence partagée par les autres applications ayant besoin d'information relative à ce produit.

Conclusions et perspectives

I. Conclusion Générale

Dans cette thèse nous avons traité le problème de l'interopérabilité orientée produit dans un environnement de production. Cette interopérabilité permet de maintenir la cohérence entre les représentations du produit manipulées par les différents composants du système d'information de l'entreprise, tout en garantissant leur correspondance avec l'état réel du produit physique tout au long de son cycle de vie (conception, fabrication, commercialisation, utilisation…).

L'approche vise une interopérabilité basée sur l'échange d'informations entre les différentes applications en relation avec le monde du produit en prenant en compte la dimension physique du produit, en se basant sur le meta-modèle holonique comme modèle de référence pour l'échange des données et informations relatives au produit. Notre démarche permet ainsi de :

- Maintenir la cohérence entre l'ensemble des représentations du produit avec la réalité physique du produit, grâce aux propriétés transactionnelles définies pour les processus holoniques.
- Maintenir la cohérence entre les différentes représentations du produit existant dans les différents systèmes, grâce aux mécanismes de transformations de modèles basés sur le meta-modèle holonique comme modèle de référence.

Dans ce chapitre nous présentons le bilan général ainsi que les conclusions et les perspectives liées à la poursuite de ce travail, aux prochaines applications ainsi qu'aux nouvelles pistes de recherche.

II. Bilan de l'approche

Dans le chapitre 2 de ce manuscrit, nous avons proposé le meta-modèle holonique que nous avons mis en place comme base pour notre approche de modélisation pour l'interopérabilité orientée produit. En effet, pour mettre en œuvre cette interopérabilité nous avons opté pour une approche d'interopérabilité dirigée par les modèles construite autour d'un modèle de référence permettant la représentation de façon générique de l'ensemble des informations associées au produit courant son cycle de vie.

L'interopérabilité dirigée par les modèles se base sur le principe de l'architecture dirigée par les modèles (Mellor et al. 2004) pour mettre en œuvre l'échange d'informations et communication entre applications. En effet, dans ce contexte la mise en œuvre des échanges entre applications passe d'abord par l'étude des correspondances, ou mappings, entre les modèles de chacune de ces applications. Les échanges sont ainsi assurés grâce à des mécanismes de transformation de modèles, permettant l'échange d'informations entre différents modèles.

Dans notre approche, nous avons adapté le concept d'holon introduit par (Koestler 1967), nous appelons ainsi « holon » le concept de modélisation nous permettant d'agréger les représentations conceptuelles d'un produit et de la partie physique reliée a ce produit. L'holon est donc la fusion du produit en tant qu'entité réelle et de ses images virtuelles telle qu'elles sont perçues par les systèmes ambiants. Il joue dès lors le rôle d'interface informationnelle entre les différents systèmes ambiants dans un environnement de production (machines, applications, systèmes d'information, opérateurs humains, etc.), et le produit ; unité évoluant au sein même de cet environnement de production.

En plus du meta-modèle holonique que nous avons proposé pour la description générique des produits, basé sur l'ontologie descriptive BWW (Bunge-Wand-Weber) (Wand and Weber 1993; Wand and Weber 1995), nous avons aussi formalisé les mappings nécessaires pour spécifier l'interopérabilité sémantique entre les différents modèles impliqués.

Pour illustrer nos propos, nous avons réalisé deux exemples de mappings entre notre meta-modèle holonique d'un coté, et de l'autre coté le meta-modèle d'UEML pour le premier exemple et les modèles de l'IEC 62264 dans le deuxième exemple. L'objectif des deux exemples est de montrer que les modèles issus de l'approche holonique pour la modélisation des produits de l'entreprise peuvent interopérer avec d'autres modèles spécifiques à différents domaines de l'entreprise que ce soient des modèles du « Manufacturing » ou même du domaine « Business ». La modélisation holonique permettrait ainsi l'ingénierie du produit dans l'intersection des deux pôles faisant ainsi partie des outils pour l'ingénierie de l'interopérabilité « B2M » en ce qui concerne le produit

L'outillage et le prototypage de l'approche de modélisation holonique ainsi que la prise en charge des mappings pour l'interopérabilité s'appuient sur l'intégration du meta-modèle holonique dans un environnement professionnel pour la modélisation d'entreprise, les mappings ont quant à eux été implémentés en utilisant les fonctionnalités d'export XML de MEGA et la spécification dans le formalisme XSLT des règles de transformation induites.

Une collaboration a été menée avec le département meunerie d'un grand groupe industriel, pour une application en environnement réel de la modélisation holonique. L'objectif de cette application est de concevoir un système de traçabilité pour les différents produits par les moulins du groupe. Notre participation dans ce projet, a consisté en l'application de l'approche de modélisation holonique pour la spécification, a priori, de l'ensemble des données et informations relatives aux produits devant être prises en compte dans le système de traçabilité, et ainsi de générer de manière automatique le schéma de données préliminaire du système (*cf.* figure 5). Les résultats de cette application ont permis, pour notre part, de mettre en œuvre puis valider le meta-modèle et l'approche proposée et pour l'entreprise, de bénéficier d'un prototype exécutable pour leur système de traçabilité et de déterminer les améliorations à apporter lors de la récolte et la saisie des données sur le terrain, en détectant les redondances, défaillances ou mauvaise utilisation des données primordiales à la traçabilité.

Une autre application s'est déroulée, quant à elle, au sein d'une entreprise pilote semi-industrielle supportée par l'Atelier Inter-établissements de Productique Lorrain (AIPL). L'objectif de cette application a été montré l'apport de la modélisation holonique au sein d'un environnement, dans lequel différents systèmes doivent synchroniser la vue qu'ils ont du

produit. Lors de cette application nous avons modélisé l'ensemble des processus mis en œuvre pour la production des produits de l'AIPL pour définir qu'elle est la structure des informations relatives à chacun des produits, puis en appliquant les mappings définis précédemment nous montrons comment ces modèles peuvent être réutilisés que ce soit pour la configuration des applications de l'entreprise pendant la phase d'ingénierie du système ou alors pour l'échange de données entre les différents systèmes et la vue holonique du produit pendant phase d'exploitation du système. Cette partie a comporté notamment l'encadrement de projets de Master visant à appliquée la modélisation holonique dans l'environnement de production de l'AIPL. Ceci nous a permis d'observer l'acclimatation de l'utilisateur avec l'ensemble des concepts introduits dans l'environnement de modélisation holonique, et le retour lié à la satisfaction de l'outil.

Globalement, pour un utilisateur non expert, le concept d'holon reste une abstraction difficilement assimilée. Cependant, le pragmatisme de l'approche de modélisation holonique que nous proposons simplifie, tout en le formalisant, le processus de représentation des produits, de leurs données et de leur cycle de vie.

Grâce, au cadre de modélisation *ZACHMAN* nous avons construit un guide d'utilisation de l'approche holonique, lors de la modélisation des produits puis par la suite lors de la transformation de modèles. Ce guide décrit l'ensemble des aspects couverts par les différentes étapes relatives à l'approche holonique de modélisation et les mappings pour l'interopérabilité orientée produit.

III. Perspectives

En termes de perspectives de recherche, nous pressentons deux pistes intéressantes : d'une part, dans la formalisation présentée dans cette thèse nous n'avons pris en compte que les correspondances entre les concepts définis par les modèles et meta-modèles. En effet, une perspective liée au problème de définition de mappings et de règles de transformation entre meta-modèles peut s'énoncer comme suit :

> *Comment prendre en compte les relations et les associations entres concepts lors de l'établissement des mappings ?*

Si l'équivalence entre composants est une notion intuitive, la prise en compte des associations et relations lors de l'étude de l'équivalence de composants (concepts) de meta-modèles peut rendre la tâche inextricable. L'autre question que l'on peut se poser concerne le type même des correspondances : dans la majorité des travaux de recherche réalisés dans ce domaine, les règles de correspondances correspondent à des transformations *un à un* (Lemesle, 1998 ; Roque, 2005), autrement dit, partir d'un concept pour arriver à un autre concept (au mieux). Qu'en est il des autres cas ; 1 vers n ou n vers 1 ? Dans ces cas là, il est nécessaire de prendre en compte les relations entre concepts que se soient les relations au niveau du meta-modèle source dans le cas d'une règle de transformation $1 \rightarrow n$, ou au niveau du meta-modèle destination dans le cas $n \rightarrow 1$. Plus généralement, la transformation de concepts doit aussi traiter les relations et les associations entre concepts. Est-il possible de raisonner sur l'un sans prendre l'autre en considération ? La question reste ouverte.

D'autre part, une deuxième perspective pour la suite des travaux de recherches concerne l'évolution du concept d'holon dans l'application que nous en faisons. En effet, jusqu'à maintenant nous nous sommes contenté de considérer l'holon ou le produit holonique comme un produit ayant une intelligence réduite ; de niveau 1 selon (Wong et al. 2002). Il serait ainsi intéressant d'étudier l'évolution du produit pour lui permettre de prendre lui-même les décisions le concernant, et lui fournir ainsi l'ensemble des fonctionnalités lui permettant de

communiquer avec son environnement de façon autonome. Cette perspective vise à donner au produit l'ensemble des propriétés faisant de lui un objet intelligent de niveau 2.

1. Une identification unique
2. aptitude à communiquer avec l'environnement
3. aptitude à retenir ou enregistrer de l'information le concernant
4. aptitude à utiliser un certain langage pour afficher ses caractéristiques et ses besoins de production.
5. aptitude à prendre ou participer à la prise de décision concernant sa propre destinée.

Cette perspective pourrait être réalisée en mettant en relation les modalités du processus ; Pouvoir-faire, Devoir-faire, Savoir-faire, Vouloir-Faire (Mayer 1995) ; et ces mêmes modalités mais du point de vue du produit cette fois-ci. Les échanges d'informations entre le produit et la ressource exécutant le processus tiendraient ainsi compte des correspondances entre les modalités du processus d'un côté et celles du produit de l'autre.

D'autres travaux de recherche au sein du groupe thématique SCP cherchent également à rendre actif le produit dit « intelligent » en lui conférant non seulement une connaissance sur ses données techniques mais aussi sur les services qu'il nécessite depuis sa conception jusqu'à son exploitation, en référence à une « ontologie produit » (Dassisti et al. 2006). Ce produit deviendrait ainsi acteur de l'interopérabilité des applications qui manipulent sa vue informationnelle et participerait donc à l'« intelligence ambiante » du système de production . Ces travaux s'inscrivent dans le cadre des thèses respectives de Melle Angela TURSI et M. Jean-Philippe AUZELLE.

Bibliographie

Abrial, J. R. (1974). "Data semantics." Data Base Management., e. Klimbie and Koffeman, ed., North-Holland Publisher.

Alban, D. (1996). "Organisation du système d'information et stratégies d'entreprise étendue, les systèmes d'information coopératifs," Thèse de doctorat à l'Institut de l'Administration des Entreprises de Paris.

Alcouffe, C. (2001). "Formes de coopération interentreprises : l'organisation de la R & D dans l'aéronautique et le spatial." *Note n°356*, LIRHE - Université des Sciences Sociales, Toulouse.

ATHENA. (2003). "Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications." ATHENA IP: IST-507849.

B2MML. (2003). "The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 2.0." *http://www.b2mml.org*.

Baïna, S., and Morel, G. (2006). "Product Centric Holons For Synchronisation And Interoperability In Manufacturing Environments." The 12th IFAC Symposium on Information Control Problems in Manufacturing, May 17-19, 2006. St-etienne, France.

Baïna, S., Panetto, H., and Benali, K. (2006a). "Apport de l'approche MDA pour une interopérabilité sémantique : Interopérabilité des systèmes d'information d'entreprise." *Ingénierie des Systèmes d'Informations* 11/3, 11-29, Hermès, Lavoisier, Juin 2006, ISBN : 2-7462-1524-1.

Baïna, S., Panetto, H., and Benali, K. (2006b). "A Product Oriented Modelling Concept: Holons for systems synchronisation and interoperability." ICEIS'06, 8th International Conference on Enterprise Information Systems, 23 - 27, May 2006. Paphos, Cyprus.

Baïna, S., Panetto, H., Benali, K., and Morel, G. (2005). "Adapting HPM to B2M interoperability issues: Towards Interoperability between Business Management Level and Shop Floor Level." IFIP/ACM SIGAPP INTEROP-ESA'05 Doctoral Symposium, February 21st 2005. Geneva, Switzerland.

Baïna, S., Panetto, H., and Morel, G. (2006c). "A Towards a Product Oriented Process Modelling for Enterprise Applications Synchronisation and Interoperability." IFAC/IFIP I-ESA'06, 2nd Interoperability for Enterprise Software and Applications Conference, March, 2006. Bordeaux, France

Bajic, E., and Chaxel, F. (1997). "Towards a holon-product oriented management." The 4th IFAC Workshop on Intelligent Manufacturing Systems (IMS'97), Seoul, Korea.

Barthes, J.-P., Vayssade, M., and Znamierowska, M. (1979). "Property driven databases." The 6th International Joint Conferences on Artificial Intelligence, August 20-23, 1979. Tokyo, Japan.

Benchimol, G. (1993). *L'entreprise étendue*, Hermès. ISBN 2866013549.

Berio, G., Anaya, V., Boudjlida, N., Krogstie, J., and Petit, M. (2003). "D3.2: Core constructs, architecture and development strategy." UEML TN IST- 2001- 34229.

Berio, G., Opdhal, A., Anaya, V., Dassisti, M., Panetto, H., Wohed, P., Baïna, S., and et al. (2005). "DEM1: UEML 2.1, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, n° IST 508-011."

Bézivin, J. (2004). "In Search of a Basic Principle for Model-Driven Engineering." *Novatica Journal* vol. 2. No 2(Special Issue, March-April 2004).

Bézivin, J. "On the Unification Power of Models." *In Software and Systems Modeling*, pp: 171-188.

Bézivin, J., and Gerbé, O. (2001). "Towards a Precise Definition of the OMG/MDA Framework." 16th IEEE International Conference on Automated Software Engineering ASE'01., Loews Coronado Bay, San Diego, USA. November 26-29, 2001.

Bézivin, J., J. Lanneluc, and R. Lemesle. "sNets : The Core Formalism for an Object-Oriented CASE Tool." *COODBSE'94 Proceedings of the Colloquium on Object Orientation in Databases and Software Engineering*, May 1994. Montreal, Quebec, pp 224-239.

Birtwistle, G., Dahl, O., Myhrhaug, B., and Nygaard, K. (1973). *Simula begin*, Petrocelli Charter. ISBN 91-44-06211-7, New York.

Blaha, M., and Premerlani, W. "A Catalog of Object Model Transformations." *3rd Working Conference on Reverse Engineering*, November 8-10, 1996. Monterey, California.

Boitard, L. (1998). "Contribution à l'intégration d'outils du génie automatique autour d'un système d'information unifié.," Thèse de doctorat de l'Université Henri Poincaré, Nancy I, France. Septembre 1998.

Boudjlida, N., Dong, C., Baïna, S., Panetto, H., Haussmann, K., Tomas, J., Abian, M., and Nunez, M. J. (2006). "DTG4.1: A practical experiment on semantic enrichment in a homogeneous environment. ." *IST 508-011, July 2006*, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence.

Bouessel du Bourg, A. (1992). "L'échange de données informatisées (EDI), un choix stratégique de management." paru dans le n°17 du Magazine "Brises".

Bourey, J.-P., Grangel, R., Berre, A., Doumeingts, G., K., K., M., B., Pondrelli, L., and Daclin, N. (2005). "DTG2.1: Report on model establishment." *IST 508-011*, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence (INTEROP Noe).

Bunge, M. (1977). *Treatise on Basic Philosophy. Volume 3, Ontology I: The Furniture of the World*, Reidel, Boston.

Bunge, M. (1979). *Treatise on Basic Philosophy. Volume 4, Ontology II: A World of Systems*, Reidel, Boston.

Candea, C., Staicu, M., and Barbat, B. (2000). "Holon like approach for robotic soccer." RoboCup Euro 2000 Workshop, 28 May - 2 June 2000.

Casati, F., Ceri, S., Pernici, B., and Pozzi, G. (1996). "Deriving active rules for workflow enactment." The 7th International Conference on Database and Expert Systems Application, September 9-13, 1996. Zurich, Switzerland.

Centum. "Centum Production Control System ", Yokogawa Electric Corporation.

Chen, D. (2005). "Modélisation d'entreprise pour l'intégration et l'intéropérabilité des systèmes industriels," Habilitation à diriger des Recherches à l'Université de Bordeaux 1, France.

Chen, D., and Doumeingts, G. (2003). "European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap." *Annual Reviews in Control*, 27, pp 153–162, December.

Dahl, O., and Nygaard, K. (1966). "Simula, an Algol-based simulation language." *Communication of the ACM*, Vol 9, pp. 671-678.

Dassisti, M., Panetto, H., and Tursi, A. (2006). "Product-driven Enterprise Interoperability for Manufacturing Systems Integration. Proceedings of the BPM2006 Business Process

Management Workshops. ." 2nd ENEI Workshop, Springer Verlag, , Lecture Notes in Computer Science, LNCS 4103, 249-260, ISBN 3-540-32595-6, Vienna, Austria, September 4, 2006. .

Davis, B. R., Smith, S., Davies, M., and St. John, W. (1983). "Integrated Computer-aided Manufacturing (ICAM) Architecture Part III/Volume III: Composite Function Model of "Design Product" (DES0)." *Technical Report AFWAL-TR-82-4063 Volume III*, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, , Ohio 45433.

Denivaldo, L., Hammoudi, S., Bézivin, J., and F., J. "Mapping Specification in MDA: from Theory to Practice." *Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications (IFIP/ACM SIGAPP INTEROP-ESA'2005).* Geneva, pp: 253-264.

Doan, A., Madhavan, J., Domingos, P., and Halevy, A. "Learning to map between ontologies on the semantic web." *the Eleventh International WWW Conference*, May 7-11, 2002. Honolulu, Hawaii, USA.

Doumeingts, G. (1984). "Methode GRAI: Methode de Conception des Systemes de Productigue," Thése de doctorat à l'Universite de Bordeaux 1, France, Bordeaux, France.

Doumeingts, G., Vallespir, B., Zanettin, M., and Chen, D. (1992). "GIM, GRAI Integrated Methodolgy, A Methodology for Designing CIM Systems." Report of the LAP/GRAI, University Bordeaux 1, Bordeaux, France.

EIF. "European Interoperability Framework for pan-European eGovernment Services, Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC)." November 2004, Luxembourg.

El Hadj Mimoune, M. (2004). "Contribution à la modélisation explicite et à la représentation des données de composants industriels : application au modèle PLIB," Thèse de doctorat à l'Université de Poitiers, France. Juillet 2004.

Elbyed, A. "Un algorithme de mapping d'ontologies pour l'interopérabilité d'entrepôts de ressources pédagogiques." *1ères Rencontres Jeunes Chercheurs en Environnements Iformatiques pour l'Apprentissange Humain, RJC-EIAH'2006*, 11-12 mai, 2006. Paris, France.

Euzenat, J. "Towards a principled approach to semantic interoperability." *Workshop on Ontologies and Information Sharing, IJCAI*, August 6th, 2001. Seattle, Washington, USA.

Favre, J.-M. "Foundations of the Meta-pyramids: Languages and Metamodels - Episode II, Story of Thotus the Baboon." *Dagsthul Seminar on Language Engineering for Model-Driven Software Development, DROPS proceedings 2004*, February 29- March 05, 2004, Dagsthul, Germany. *http://drops.dagstuhl.de/portals/04101/.*

Favre, J.-M. "Megamodelling and Etymology - A Story of Words: From MED to MDE via MODEL in five milleniums." *Dagstuhl Seminar 05161 on "Transformation Techniques in Software Engineering"*, May 2005. Dagsthul, Germany.

Fettke, P., and Loos, P. "Ontological Evaluation of Reference Models using the Bunge-Wand-Weber Model." *Proceedings of the 9th Americas Conference on Information Systems*, August 4-5, 2003. Florida, USA pp. 2944–2955.

FIPA. (2002). "FIPA ACL Message Structure Specification." *FIPA Standard SC00061*.

Frachet, J. P. (1987). "Une introduction au génie automatique: faisabilité d'une chaîne d'outlis CAO pour la conception et l'exploitation des machines automatiques industrielles," Thèse de doctorat de l'Université Henri Poincaré, Nancy I, France. 1987.

Gerber, C., Siekmann, J., and Vierke, G. (1999). "Holonic multi-agent systems." *Research Report RR-99-03*, DFKI.

Gouyon, D. (2004). "Contrôle par le produit des systèmes d'exécution de la production : apport des techniques de synthèse," Thèse de doctorat de l'Université Henri Poincaré, Nancy I, France. Décembre 2004, Nancy, France.

Gouyon, D., Simão, J. M., Alkassem, K., and Morel, G. (2004). "Work in progress for product driven manufacturing automation." the 11th IFAC INCOM Symposium, April 7th-9th, 2004. Salvador de Bahia, Brazil.

Gruber, T. R. (1993). "Towards Principles for the Design of Ontologies Used for Knowledge Sharing " Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publisher.

Gustas, R. (1995). "A Basis for Integration within Enterprise Modelling." The 2nd International Conference on Concurrent Engineering: Reasearch and Applications, August 23-25, 1995. Washington, DC Area, USA.

HMS. (1994). "HMS Requirements." http://hms.ifw.uni-hannover.de/: HMS Server.

IDEAS. (2002). "Interoperability Development for Enterprise Application and Software." IST-2001-37368.

IEC 62264. (2002). "IEC FDIS 62264-1:2002, Enterprise-control system integration." Geneva, Switzerland.

IEC 62390. (2005). "Common automation device. Profile guideline, IEC TC 65/290/DC Device Profile Guideline (2002), TC65: Industrial Process Measurement and Control." IEC, Geneva, Switzerland.

IEEE. (1990). "IEEE: Standard Computer Dictionary- A Compilation of IEEE Standard Computer Glossaries."

IGES. (1980). "Initial Graphics Exchange Specification - ANSI Y 14.26M." A. N. S. I. (ANSI), ed., USA.

INTEROP. (2003). "Interoperability research for Networked Enterprises Applications and Software." INTEROP-NoE IST 508011.

ISO 8402. (1994). "Vocabulaire pour le management et l'assurance de la qualité." AFNOR.

ISO 10303-11. (1994). "Industrial automation systems: and integration — Product data representation and exchange." Geneva, Switzerland.

ISO 10303-21. (1994). "Industrial automation systems and integration — Product data representation and exchange." Geneva, Switzerland.

ISO 10303. (1994). "Industrial Automation Systems and Integration - Product Data Representation and Exchange." Geneva, Switzerland.

Janis, A. (2005). "Keynote: A Historical Perspective on Conceptual Modelling: from Information Algebra to Enterprise Modelling and Ontologies." Tenth CAiSE/IFIP8.1/IINTEROP/AIS-SIGSAND International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'05), June 13-14, 2005. Porto, Portugal.

Jorysz, H. R., and Vernadat, F. B. (1990a). "CIM-OSA Part 1: total enterprise modelling and function view." *International Journal of Computer Integrated Manufacturing*, Vol. 3, pp. 144 - 156.

Jorysz, H. R., and Vernadat, F. B. (1990b). "CIM-OSA Part 2: information view." *International Journal of Computer Integrated Manufacturing*, Vol. 3, 157-167.

Kalfoglou, Y., and Schorlemmer, M. (2003). "Ontology Mapping: The State of The Art." *The Knowledge Engineering Review, Cambridge University Press*, vol. 18, pp. 1-31.

Kalfoglou, Y., and Schorlemmer, M. (2004). "Formal Support for Representing and Automating Semantic Interoperability." 1st European Semantic Web Symposium (ESWS'04), Heraklion, Crete, Greece, 45-61.

Kleinberg, K., and Merriman, M. (1996). "Business Process Re-engineering Techniques and Tools : Bridging to Implementation." Gartner Group.

Klischewski, R. (2004). "Information integration or process integration: How to achieve interoperability in administration." EGOV04 at DEXA conference, 30 August - 3 September, 2004. Zaragoza, Spain.

Klittich, M. (1990). "CIM-OSA Part 3: CIM-OSA integrating infrastructure - the operational basis for integrated manufacturing systems." *International Journal of Computer Integrated Manufacturing*, Vol. 3, 168 - 180.

Koestler, A. (1967). *The Ghost in the Machine*, Arkana, ISBN 0140191925, London.

Kosanke, K. "ISO Standards for Interoperability: a comparison." *INTEROP-ESA'05*, Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications (IFIP/ACM SIGAPP INTEROP-ESA'2005).

Lemesle, R. (1998). "Transformation Rules Based on Meta-Modelling." EDOC'98, November 3rd-5th, 1998. La Jolla, California, 113-122.

Lemesle, R. (2000). "Techniques de Modélisation et de Méta-modélisation," Thèse de doctorat de l'Ecole Centrale de Nantes. Octobre 2000

Linthicum, D. S. (1999). *Enterprise Application Integration*, Addison Wesley. ISBN 0201615835.

LISI. (1998). "Levels of Information Systems Interoperability." C4ISR Interoperability Working Group, Department of Defense, Washington, DC.

Madhavan, J., Bernstein, P., Domingos, P., and Halevy, A. "Representing and reasoning about mappings between domain models",." *In Eighteenth National Conference on Artificial Intelligence (AAAI'2002)*, July 30-August 1, 2002. Edmonton, Canada.

Manset, D., Verjus, H., McClatchey, R., and Oquendo, F. (2006). "A Formal Architecture-Centric Model-Driven Approach for the Automatic Generation of Grid Applications." Proc of the 8th International Conference on Enterprise Information Systems (ICEIS06), May 2006. Paphos, Cyprus.

Martin, C., Nowlin, A., St. John, W., Smith, S., Ruegsegger, T., and Small, A. (1983). "Integrated Computer-aided Manufacturing (ICAM) Architecture Part III/Volume VI: Composite Information Model of "Manufacture Product" (MFG1)." *Technical Report AFWAL-TR-82-4063*, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio.

Martin, C., and Smith, S. (1983). "Integrated Computer-aided Manufacturing (ICAM) Architecture Part III/Volume IV: Composite Information Model of "Design Product" (DES1)." *Technical Report AFWAL-TR-82-4063 Volume IV*, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433.

Maturana, F. (1997). "MetaMorph: an adaptive multi-agent architecture for advanced manufacturing systems," Ph.D. dissertation at the University of Calgary, Canada.

Maturana, F., Shen, W., and Norrie, D. H. (1999). "MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing." *International Journal of Production Research*, vol. 37(10), pp: 2159-2174.

Maurino, M. (1995). *La gestion des données techniques. Technologie du concurrent engineering*, Editions Masson.

Mayer, F. (1995). "Contribution au Génie Productique: Application à l'ingénierie Pédagogique en Atelier Inter-établissements de Production Lorrain," Thèse de doctorat de l'Université Henri Poincaré, Nancy I.

Mc Farlane, D., S. Sarma, J.L. Chirn, C.Y. Wong, and Ashton, K. "The Intelligent Product in Manufacturing Control and Management." *The 14th IFAC World Congress*.

McFarlane, D. (2002). "Auto-ID Based Control - An Overview." *CAM-AUTOID-WH-004*, Auto-ID Center, University of Cambridge.

McFarlane, D., S. Sarma, J.L. Chirn, C.Y. Wong, and Ashton, K. "The Intelligent Product in Manufacturing Control and Management." *The 14th IFAC World Congress*, July 5-9. 2005. Beijing, China.

Mellor, S. J., Kendall, S., Uhl, A., and Weise, D. (2004). *Model Driven Architecture*, Addison-Wesley Pub Co. ISBN 0201788918.

Minsky, M. L. (1975). "A framework for representing knowledge " The Psychology of computer vision, McGraw Hill. ISBN 0-262-62101-0, New-York.

Molina, A., Chen, D., Panetto, H., Vernadat, F., and Whitman, L. (2004). "Enterprise Integration and Networking: Issues, trends and vision." International Conference on Enterprise Integration and Modeling Technology (ICEIMT'04), Kluwer Academics Publisher, Toronto, Canada.

Morel, G., Panetto, H., Zaremba, M. B., and Mayer, F. (2003). "Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues." *IFAC Annual Reviews in Control*, 27/2, 199-209.

Morris, E., Levine, L., Meyers, C., Plakosh, D., and Place, P. (2004). "Systems of Systems Interoperability (SOSI)." *CMU/SEI-2004-TR-004 (ESC-TR-2004-004).* Software Engineering Institute, Carnegie Mellon University. .

NATO C3. (2003). "NATO C3 Technical Architecture (NC3TA) Reference Model for Interoperability." The Hague, The Netherlands: 2003. *http://www.nc3a.nato.int/index.html*.

Naumenko, A., and Wegmann, A. (2003). "Two Approaches in System Modelling and Their Illustrations with MDA and RM-ODP." ICEIS 2003, the 5th International Conference on Enterprise Information Systems, April 23-26, 2003. Angers, France, 398-402.

NF X 50 120. (1987). "Vocabulaire pour le management et l'assurance de la qualité." AFNOR.

Opdahl, A. L., and Henderson-Sellers, B. (2002). "Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model." Software and Systems Modeling, Springer, pp. 43-67.

Panetto, H. (2006a). "Meta-modèles et modèles pour l'intégration et l'interopérabilité des applications d'entreprises de production," Habilitation à Diriger des Recherches à l'Université Henri Poincaré, Nancy I, France.

Panetto, H. (2006b). "Towards a Classification Framework for Interoperability of Enterprise Applications." *Int. J. of CIM, Taylor & Francis*.

Panetto, H., Baïna, S., and Morel, G. (2006). "Mapping the IEC 62264 models onto the Zachman framework for analysing products information traceability: a case study " *Journal of Intelligent Manufacturing, Springer Verlag. ISSN: 0956-5515.*

Panetto, H., Berio, G., Benali, K., Boudjlida, N., and Petit, M. (2004). "A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models." IFAC INCOM 2004 Symposium, April 5th-7th, 2004 . Bahia, Brazil.

Petrie, C. (1992). *Entreprise Integration Modeling*, The MIT Press, Cambrindge, MA.

Pierra, G. "Représentation et échange de données techniques." *Conférence de Mécanique Industrielle 2000 (Mec Ind 2000).* vol. 1. pp. 397-414.

Pohl, K. "The three Dimensions of Requirements Engineering." *International Conferance on Advanced Information System Engineering CAiSE'93*, Paris, France.

Rahm, E., and Bernstein, P. A. (2001). "A Survey of Approaches to Automatic Schema Matching." *VLDB Journal*, vol. 10, pp 334-350.

Rational Rose White Paper. (2001). "The Zachman Framework for Enterprise Architecture and Rational Best Practices and Products." *http://se2c.uni.lu/tiki/se2c-bib_download.php?id=688*.

Roque, M. (2005). "Contribution a la définition d'un langage générique de modelisation d'entreprise," Thèse de doctorat de l'Université de Bordeaux 1, France. November 2005.

Rosemann, M., and Green, P. (2002). "Developing a meta model for the Bunge–Wand–Weber ontological constructs." *Information Systems Journal*, Vol. 27(2), pp 75-91.

Rothenberg, J. (1989). "The Nature of Modeling " Artificial Intelligence, Simulation and Modeling, John Wiley & Sons, Inc., New York, pp 75-92.

Seidel, D., and Mey, M. (1994). "IMS - Holonic Manufacturing Systems: Glossary of Terms." IMS - Holonic Manufacturing Systems: Strategies, Seidel D. and Mey M. (eds), University of Hannover, Germany.

Seidewitz, E. (2003). "What Models Mean." *IEEE Software*, September 2005.

Selk, B., Kloeckner, S., and Albani, A. (2005). "Enabling interoperability of networked enterprises through an integrative information system architecture for CRM and SCM." The 3rd Business Process Modelling Conference, September 5-8, 2005. Nancy, France.

SEMATECH. (1995). "Device Interoperability Guideline for Sensors, Actuators, and Controllers." Technology Transfert 94102567A-STD, *http://www.sematech.org*.

SET. (1989). "SET: Z 68-300, Industrial Automation – External Representation of Product Definition Data – Data Exchange and Transfert Standard Specification." A. F. d. N. AFNOR), ed., France.

Shen, W., and Barthes, J. P. (1995). "Description and Applications of an Object-Oriented Model PDM, Modeling Complex Data for Creating Information: Real and Virtual Objects." Dubois J.E. & Gershon N. (eds.), Springer-Verlag, pp 15-24.

Silva, N., and Rocha, J. A. "Ontology mapping for interoperability in semantic web." *in Proceedings of the IADIS International Conference WWW/Internet 2003 (ICWI'2003)*, 5-8 Nov, 2003. Algarve, Portugal.

Sowa, J. F., and Zachman, J. A. (1992). "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal,* , Vol. 31(3), pp. 590 - 616.

Stumme, G., and Maedche, A. "Ontology Merging for Federated Ontologies on the Semantic Web." *In Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001)*, September, 2001. Viterbo, Italy.

Terzi, S. (2005). "Elements of Product Lifecycle Management: Definitions, Open Issues and Reference Models," Thèse de doctorat de l'Université Henri Poincaré Nancy I en cotutelle avec le Politecnico di Milano (en Italie), 27 mai 2005.

Terzi, S., Cassina, J., Chiari, G., and Panetto, H. (2004). "Traçabilité des produits: une apporche holonique." The 5th French Speaking Conference on Modelling and Simulation, MOSIM'04, Nantes, France.

Tolk, A., and Muguira, J. A. (2003). "The Levels of Conceptual Interoperability Model." Simulation Interoperability Workshop, Orlando, Florida, USA.

UEML. (2002). "Unified Enterprise Modelling Language (UEML) Thematic Network." IST-2001-34229.

Uschold, M., King, M., Moralee, S., and Y., Z. (1997). "The Enterprise Ontology." Artificial Intelligence Applications Institute (AIAI), Report of the Artificial Intelligence Applications Institute (AIAI). The University of Edinburgh.

Valckenaers, P. (2001). "Special issue on Holonic Manufacturing Systems." *Computers in Industry*, 46(3), October.

Vallecillo, A., Hernandez, J., and Troya, J. M. (2000). "Component interoperability." *Tech. Rep. ITI-2000-37*, Departmento de Lenguajes y Ciencias de la Computacion, University of Malaga.

Vallespir, B., Chen, D., and Ducq, Y. (2005). "Enterprise Modelling for interoperability." 16th IFAC world congress, Prague, République Tchèque, 4-8 juillet 2005.

Vallespir, B. (2003). "Modélisation d'entreprise et architecture de conduite des systèmes de production," Habilitation à Diriger des Recherches à l'Université Bordeaux 1, 19 décembre 2003.

Vallespir, B., Braesch, C., Chapurlat, V., and Crestani, D. (2003). "L'intégration en modélisation d'entreprise : les chemins d'UEML." La 4ème conférence francophone de Modélisation et Simulation, Organisation et conduite d'activités dans l'industrie et les services, MOSIM'03, Toulouse, France, 23-25 avril 2003.

Van Brussel, H., J. Wyns, Valckenaers, P., Bongaerts, L., and Peeters, P. (1998). "Reference Architecture for holonic manufacturing systems: Prosa." *Computers in Industry*, 37(3), 255-274.

Vega, G. (2005). "Développement d'applications à grande échelle par composition de métamodèles," Thèse de doctorat à l'Université Joseph Fourier à Grenoble, France, December 2005.

Vernadat, F. B. (1996). *Enterprise modelling and integration: principles and applications*, Chapman & Hall. ISBN: 978-0-412-60550-5.

Viruéga, J.-L. (2005). *Traçabilité: Outils, méthodes et pratiques*, Éditions d'Organisation. ISBN : 2-7081-3260-1.

Visser, U., Stuckenschmidt, H., Wache, H., and Vogele, T. "Enabling technologies for interoperability." *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, Bonn, Germany, 2000., 35--46.

Walmsley, P. (2001). *Definitive XML Schema* Prentice Hall. ISBN: 0130655678.

Wand, Y., and Weber, R. (1993). "On the ontological expressiveness of information systems analysis and design grammar." *Information Systems Journal*, 3, 217 - 237

Wand, Y., and Weber, R. (1995). "On the deep structure of information systems." *Information Systems Journal*, 5, 203 - 223.

Warmer, J., and Kleppe, A. (1999). *The Object Constraint Language: Precise Modeling with UML.*, Addison-Wesley. ISBN 0-201-37940-6.

Wegner, P. (1996). "Interoperability." *ACM Computing Survey*, vol. 28(1), pp 285–287.

WFMC. (1995a). "Workflow Management Coalition: The Workflow Reference Model." *TC-1003*.

WFMC. (1995b). "Workflow standard – Interoperability abstract specification." *Document Number WFMC-TC-1012*, Workflow Management Coalition.

WFMC. (2000). "Workflow Management Coalition: Workflow standard – Interoperability Wf-XML Binding." *Document Number WFMC-TC-1023*.

Willars, H. (1993). "TRIAD Modelleringshandbok: Business Modelling Overview." SISU.

Wong, C. Y., McFarlane, D., Ahmad Zaharudin, A., and Agarwal, V. "The intelligent product driven supply chain." *IEEE International Conference on Systems, Man and Cybernetics*, October 10-13 2004 The Hague, The Netherlands.

WordNet. "Princeton University Cognitive Science Lab."

Wyns, J. (1999). "Reference architecture for holonic manufacturing system," PhD Thesis at the "Katholieke Universiteit" of Leuven.

XML. (1998). "Extensible Markup Language (XML) 1.0." World Wide Web Consortium.

XSLT. (1999). "XSL Transformations (XSLT) Version 1.0, November, 1999." *http://www.w3.org/TR/xslt*.

Yutaka, S. (1998). "Information unification between enterprise resource planning system and production control system." Yokogawa Technical Report, English Edition, N° 25.

Zachman, J. A. (1987). "A Framework for Information Systems Architecture." *IBM Systems Journal* vol. 26(3), pp. 276-292.

Zur Muehlen, M., and Klein, F. (2000). "AFRICA: Workflow Interoperability based on XML-messages. ." CAiSE 2000 Workshop on Infrastructures for Dynamic Business-to-Business Service Outsourcing (ISDO '00), June 5-6, 2000. Stockholm, Sweden

Annexes

Annexe A

Afin d'instaurer les règles permettant l'instanciation de modèles valides selon la sémantique du meta-modèle présenté dans le chapitre 2, nous définissons un ensemble de contraintes que nous formalisons en OCL. OCL (Object Constraint Language) est le langage formel de prédicats du 1er ordre standardisé dans les modèles UML. Il existe deux types de contraintes OCL

- Invariant: les invariants sont des conditions qui portent sur toutes les instances de classificateurs. Ce sont souvent des conditions supplémentaires au sein d'une classe, d'un type ou sur l'association entre les classes que l'on ne peut pas spécifier en utilisant les notations graphiques UML.
- Pré-conditions et post-conditions: OCL fournit une syntaxe spéciale pour spécifier les pré- et post-conditions d'opérations dans le modèle UML. Pré- et post-conditions sont des contraintes qui définissent un contrat que l'implémentation d'opérations doit satisfaire. La pré-condition décrit la condition que l'on doit respecter avant l'exécution de l'opération. La post-condition décrit les effets produits par l'exécution de l'opération.

Etant donné que nos contraintes concerne uniquement les classes et les instances de classes, nous n'utiliserons pour leur formalisation que les blocs « *invariant* » représenté par le mot clé OCL « **inv » ;**

- *Contraintes sur la composition des Holons*

  contrainte 1 :

  « Un holon qui ne possède pas de partie physique, est obligatoirement un holon complexe composé d'autres holons. »

  En effet, si un holon est obtenu par composition de plusieurs holon, il peut ne pas posséder de partie physique qui lui est propre, par contre, il possède une partie informationnelle.

```
context Holon
inv contrainte1:
```

```
self.Partie_Physique -> size()=0 implies self.oclIsTypeOf(Holon_Complexe) and
self.children ->size() > 0)
```

contrainte 2 :

« Un holon élémentaire possède nécessairement une partie physique et une partie informationnelle. »

Cette propriété est nécessaire pour assurer que les feuilles dans l'arborescence de composition d'un holon sont des holons élémentaires qui possèdent nécessairement une partie physique.

Ocl :

```
context Holon_Elementaire

inv contrainte2:

self.oclAsType(Holon). Partie_Physique -> size()=1

     and self.oclAsType(Holon).Partie_Informationnelle -> size()=1
```

- *Contraintes sur les Attributs*

contrainte 3 :

« Un attribut qui décrit une partie physique d'un holon fait nécessairement partie de la partie informationnelle de ce même holon. »

En effet, on peut retrouver les attributs d'une partie physique d'un holon soit en listant l'ensemble des attributs qui lui sont associés directement, soit en sélectionnant les attributs de la partie informationnelle du même holon. Grâce à cette contrainte, les deux requêtes donnent le même résultat.

Ocl :

```
context Attribut

inv contrainte3:

self.PartiePhysique.holon -> size()=1 implies
       (self.Partie_Informationnelle -> size()=1 and
        self.Partie_Informationnelle.holon= self.Partie_Physique.holon)
```

contrainte 4 :

« Si un attribut décrit une partie physique ne faisant pas partie d'un holon, alors il ne peut pas être associé à une partie informationnelle. »

En effet, étant donné qu'une partie physique peut exister indépendamment d'une entité *Holon,* les attributs intrinsèques à cette partie physique ne sont décrit par aucune partie informationnelle.

Ocl :

```
context Attribut
inv contrainte4:
self.Partie_Physique.holon -> size()=0 implies
      self. Partie_Informationnelle -> size()=0
```

- <u>*Contraintes sur les Etats*</u>

contrainte 5 :

« Un état relatif à un holon ne fait référence qu'aux attributs ou propriétés reliés à ce même holon. »

Cette contrainte, assure la cohérence entre les éléments décrivant l'état d'un holon, et les caractéristiques de l'holon lui-même.

Ocl :

```
context Etat
inv contrainte6:
if self.Holon_Complexe -> size()=1
      then forall self.Valeur_attribut -> forall(va|
      va.attribut.Partie_Physique.holon =
      self.Holon_Complexe.oclAsType(Holon))
else   if self.Holon_Elementaire ->size()=1
      then forall self Valeur_attribut -> forall(va|
      va.attribut.Partie_Physique.holon =
      self.Holon_Elementaire.oclAsType(Holon))
end if
```

Annexe B

Dans cette annexe, nous présentons la définition standard des propriétés ACID dans le domaine des transactions :

- *Atomicité* **-** Une transaction peut être faite ou défaite totalement, et sans ambiguïté. Dans le cas d'une défaillance ou d'un échec, toutes les opérations constituant la transaction doivent être défaites. Les données sont alors remises dans leur état initial avant l'enclenchement de la transaction.
- *Cohérence* - Une transaction doit préserver la cohérence des propriétés invariantes relatives aux données. A la fin d'une transaction se terminant avec succès, les données sont toujours dans un état cohérent. Ainsi, l'ensemble du système passe d'état stable en état stable
- *Isolation* **–** Chaque transaction doit pouvoir s'exécuter en parallèle avec d'autres transactions dans le même environnement: L'exécution en parallèle d'un certains nombre de transactions, doit avoir le même effet que leur exécution en série. Ceci nécessite deux axiomes principaux :
    - Durant l'exécution d'une transaction, aucun état intermédiaire des données ne doit être accessible ou exposé aux autres transactions.
    - Deux transactions concourantes ne peuvent modifier simultanément les mêmes données.
- *Durabilité* - Les effets d'une transaction validée sont persistants à travers le temps.

Annexe C

Afin d'assurer la cohérence sémantique des modèles réalisés en utilisant l'implémentation ***MEGA*** du meta-modèle holonique présenté dans le chapitre 2, nous avons utilisé la fonctionnalité de l'environnement *MEGA* permettant de définir des règles de modélisation pour contraindre l'utilisation des modèles *MEGA* et assurer ainsi leur sémantique. Nous présentons dans ce qui suit quelques unes des règles de modélisation implémentées dans le nouvel environnement de *MEGA :*

- Règle 1 : cette première règle exprime le fait qu'un flux doit obligatoirement avoir un émetteur et un récepteur. Pour un flux dans notre implémentation, l'émetteur et le récepteur peuvent être de 5 types différents: *Acteur*, *Activité*, *Processus*, *Procédure* et *Site*.

  Ci-dessous l'ensemble des Tests pour le Récepteur :

  Il faut donc vérifier que il existe au moins un lien de type « Réception » avec au moins un de ces cinq types d'objets. Ce qui se traduit en une expression « OU exclusif » entre 5 prédicats :

```
// Si le nombre de lien "Acteur Récepteur" est supérieur à 0
      ItemCount([Acteur Récepteur]) > 0
// Si le nombre de lien " Activité Réceptrice" est supérieur à 0
      ItemCount([Activité Réceptrice]) > 0
// Si le nombre de lien " Procédure Réceptrice" est supérieur à 0
      ItemCount([Procédure Réceptrice]) > 0
// Si le nombre de lien " Processus Récepteur" est supérieur à 0
      ItemCount([Processus Récepteur]) > 0
// Si le nombre de lien " Site Récepteur" est supérieur à 0
      ItemCount([Site Récepteur]) > 0
```

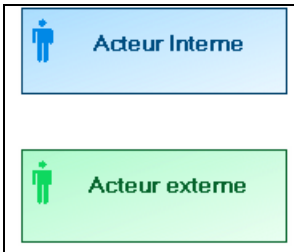
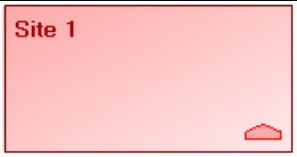
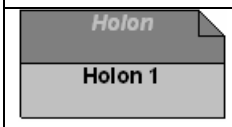
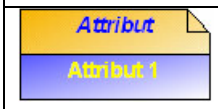
- Règle 2 : Etant donné que le meta-modèle *MEGA* ne permet pas de différencier les différents types de flux (holonique, informationnelle, physique) qu'à travers leur contenu, cette règle exprime le fait que le type d'un flux dépend du type des éléments qu'il contient.

- Un "Flux" de Type Holonique doit contenir un " Holon "
- Un "Flux" de Type Informationnel doit contenir une " Partie Informationnelle "
- Un "Flux" de Type Physique doit contenir une " Partie Physique "
- Un "Flux" non typé ou possédant un type en dehors du modèle Holonique ne doit pas être vérifié

- Règle 3 : Un holon doit obligatoirement être contenu dans un flux au moins, de même une partie physique doit faire partie d'un flux, ou bien d'un holon qui lui-même est compris dans un flux, même chose pour la partie informationnelle.
- Règle 4 : Cette règle concerne le fait qu'un *attribut holonique* doit obligatoirement être relié à un *holon*, en effet, il est impossible de créer un *attribut holonique* sans l'assigner à un *holon*.
- Règle 5 : De même, une *propriété holonique* doit forcément être rattachée à un *holon*.

Annexe D

Légende et définition des différents concepts apparaissant dans les modèles holoniques

	Un *acteur* représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise.
	Un *processus* est une chaîne de valeur fournissant un bien ou un service à un client interne ou externe à l'entreprise. Cette chaîne de valeur est décrite par une séquence d'activités de transformation. Elle est mise en œuvre par des procédures.
	Un *Site* est un lieu géographique où est implantée l'entreprise. Les sites peuvent être des sites-types tels que le siège, l'agence, l'usine, ou des lieux géographiques précis comme l'agence de Marseille, l'usine de Poissy, etc.
	Un *flux* représente un échange d'information, de matière ou des deux en même temps entre différents processusde l'environnement de production. Les holons, les parties physiques et les parties informationnelles, peuvent ainsi être véhiculés à travers différents types de flux. Le type d'un flux étant contraint par la nature de ce qu'il contient.
	Un *holon* est une représentation des objets véhiculés dans les flux exprimant l'aggrégation d'une partie physique et une partie informationnelle décrite par des attributs et des propriétés.
	Un *attribut* est un élément informationnel décrivant un holon et représentant une caractéristique intrinsèque à l'objet physique lié à cet l'holon.
	Une *propriété* est un élément informationnel représentant une caractéristique non intrinsèque à l'objet physique de par l'holon

Résumé

Les travaux de la thèse présentent une approche pour l'interopérabilité entre les différents modèles de produit, nous appellerons cette approche « l'interopérabilité orientée produit ». Nous proposons ainsi un meta-modèle dont les instances jouent le rôle de passerelle de communication entre différentes applications d'entreprise pour assurer l'interopérabilité des parties de systèmes concernant le produit.

Nous nous sommes intéressés à formaliser un meta-modèle pour la définition du concept de produit comme l'agrégation d'une partie physique représentant les éléments physiques du produit et une partie informationnelle reprenant les éléments informationnels décrivant le produit.

L'outillage et le prototypage du concept de produit holonique lors de la modélisation du processus de fabrication dans l'entreprise ainsi que la prise en charge des mapping pour l'interopérabilité s'appuient sur l'intégration du meta-modèle holonique dans un environnement de modélisation d'entreprise particulier.

La phase de validation a été réalisée en deux parties représentées chacune par une application industrielle. La première application se situe dans le cadre d'une collaboration avec le département meunerie dans un grand groupe industriel, pour une application en environnement réel de la modélisation holonique. L'objectif de cette application est de concevoir un système de traçabilité pour les différents produits par les moulins du groupe. Notre participation dans ce projet, a consisté en l'application de l'approche de modélisation holonique pour la spécification, a priori, de l'ensemble des données et informations relatives aux produits devant être prises en compte dans le système de traçabilité, et ainsi de générer de manière automatique le schéma de données préliminaire du système. La seconde application concerne la mise en œuvre de l'approche holonique pour une solution d'interopérabilité orienté produit dans le cadre du pôle AIP Lorrain (AIPL).