

THÈSE

pour obtenir le grade de
DOCTEUR DE L'ÉCOLE CENTRALE DE LILLE
Doctorat délivré conjointement par l'École Centrale de Paris
et par l'École Centrale de Lille
Discipline : GENIE INDUSTRIEL
présentée et soutenue publiquement le 31/Août/2006
par

Monsieur Hatem BEN STA

CONTRIBUTION DE LA MODELISATION CONCEPTUELLE A L'INGENIERIE DU KNOWLEDGE MANAGEMENT : APPLICATION DANS LE CADRE DE LA MEMOIRE DE PROJET

Directeurs de thèse :

Michel BIGAND, Maître de Conférences, HdR, à l'École Centrale de Lille
Jean-Pierre BOUREY, Professeur à l'École Centrale de Lille
Khaled GUEDIRA, Professeur à l'Institut Supérieur de Gestion de Tunis

JURY :

M. Salah MAOUCHE	Président	Professeur à l'Université de Lille 1
M. Faiez GARGOURI	Rapporteur	Maître de Conférences à l'Institut Supérieur d'Informatique et de Multimédia de Sfax
M. Emmanuel CAILLAUD	Rapporteur	Professeur à l'Université Louis Pasteur de Strasbourg
M. Moncef TAGINA	Examineur	Maître de Conférences à l'École Nationale des Sciences l'Informatique

REMERCIEMENTS

Ce travail a été réalisé dans le cadre d'une cotutelle entre l'Ecole Centrale de Lille et l'Ecole Nationale des Sciences de l'Informatique a Tunis au sein du Laboratoire Génie Industriel de Lille (LGI Lille) d'une part et de l'Unité de Recherche en Stratégies d'Optimisation des Informations et de la connaissance (SOIE) de l'autre, sous la direction des Professeurs Jean Pierre BOUREY (EC Lille), Khaled GHEDIRA (ENSI-ISG) et Michel BIGAND Maître de Conférences à (EC Lille) que je remercie vivement pour l'accueil, l'encadrement, le soutien et les précieux conseils dont j'ai bénéficié tout au long de ce travail.

Je suis très reconnaissant à Monsieur Emmanuel CAILLAUD Professeur à l'Université Louis Pasteur de Strasbourg et à Monsieur Faiez GARGOURI Maître de Conférences à l'Institut Supérieur d'Informatique et de Multimédia de Sfax pour l'honneur qu'ils me font en acceptant d'être les rapporteurs de cette thèse. Leurs remarques et conseils m'ont permis d'améliorer la qualité du mémoire.

Toute ma gratitude va ensuite aux membres du jury de ma thèse. Je remercie ainsi Monsieur Salah MAOUCHE Professeur à l'Université de Lille 1 autant que président de jury et Monsieur Moncef TAJINA Maître de Conférences à l'Ecole Nationale des Sciences de l'Informatique de l'Université la Manouba autant qu'examineur.

Je ne saurais oublier ici Monsieur Hédi AMARA, Professeur à l'Institut National des Sciences Appliquées et de Technologies de Tunis pour le soutien et pour la confiance qu'il m'a accordé durant les trois années de thèse.

Je tiens à remercier Monsieur Jean Claude BOCQUET directeur du Laboratoire de Génie Industriel à l'Ecole Centrale de Paris de m'avoir accepté pour faire cette thèse dans son équipe.

Je tiens à remercier Monsieur Mounib MEKHILEF Maître de Conférences à l'Université d'Orléans à l'origine de l'idée de cette thèse.

Je tiens à remercier Monsieur Etienne CRAYE directeur de l'Ecole Centrale de Lille de m'avoir encouragé à réaliser cette thèse.

Je tiens à remercier Monsieur Armand TOGUYENI directeur de l'Institut de Génie Informatique et Industriel.

Je souhaite finalement adresser mes sincères remerciements à chacun des membres de ma famille pour leur patience et leur compréhension d'avoir su me motiver et m'encourager durant certains moments difficiles.

L'ordre de mes remerciements n'a pas d'importance. Tous ceux que j'ai nommés m'ont apporté, à un moment ou un autre, un soutien décisif.

TABLE DES MATIERES

Liste des Figures.....	6
INTRODUCTION GENERALE.....	9
CHAPITRE I : CONTEXTE ET MOTIVATIONS	13
I.1. Introduction.....	13
I.2. La connaissance, un enjeu pour l'entreprise	14
I.2.1. Contexte et défis	14
I.2.2. Qu'est-ce que la gestion des connaissances ?	16
I.2.3. Typologie des Connaissances	19
I.2.4. Objectifs pour l'entreprise	21
I.3. Mémoires d'entreprise et capitalisation des connaissances.....	22
I.3.1. Définitions.....	22
I.3.2 Typologies de mémoire d'Entreprise.....	23
I.4. La gestion des connaissances en conception et en gestion de projets : cas du projet SEP ..	24
I.4.1. Introduction.....	24
I.4.2. Cas du Projet de la Société Européenne de Propulsion (SEP).....	25
I.5. Pourquoi une Mémoire de projet	26
I.6. Etat de l'art lié à l'activité de conception :	27
I.6.1. Introduction :	27
I.6.2. Définitions.....	27
I.6.3. Caractéristiques générales de l'activité de conception :	29
I.6.4. Les différents types de conception :	29
I.6.5. Les Approches de conception :	30
I.6.6. Synthèse sur la conception.....	32
I.6.7. Les Processus de conception.....	32
I.6.7.1. Définitions.....	32
I.6.7.2. Modélisation du processus de conception.....	33
I.6.7.3. Modélisation du processus d'innovation	34
I.7. Conclusion	34
CHAPITRE II : ETAT DE L'ART LIE A LA MEMOIRE DE PROJET	36
II.1. Introduction	36
II.2. Etat de l'art du Knowledge Management (KM).....	36
II.2.1. Le KM: Une branche pluridisciplinaire.....	36
II.2.2. Les dimensions du KM.....	38
II.2.3. Le KM : un ensemble de processus.....	39
II.2.4. Enjeux et Obstacles d'une démarche KM	40
II.2.4.1. Les enjeux.....	40
II.2.4.2. Les obstacles.....	40
II.3. Les Systèmes de Gestion de Connaissances (SKM)	41
II.3.1. Définition d'un SKM.....	42
II.3.2. Fonctionnalités d'un SKM.....	43
II.3.3. Exemples de SKM existants	44

II.3.3.1. Les SKM intégrés	45
II.3.3.2. Les SKM Collaboratifs.....	46
II.3.3.3. Les systèmes centrés sur la représentation de connaissances.....	48
II.3.3.4. Conclusion	49
II.4. Etat de l'Art des méthodes de formalisation des connaissances	50
II.4.1. Introduction	50
II.4.2. Méthodes de capitalisation des connaissances	51
II.4.2.1. La méthode REX	51
II.4.2.2. La méthode CYGMA	52
II.4.2.3. L'approche Componential Framework	53
II.4.2.4. L'approche EMMA	54
II.4.2.5. La méthode SAGACE	55
II.4.2.6. La méthode MKSM.....	57
II.4.2.7. La méthode CommonKADS	59
II.4.2.8. La méthode KOD.....	61
II.4.3. Les Approches de capitalisation de mémoire de projet.....	61
II.4.3.1. La méthode IBIS.....	61
II.4.3.2. L'approche QOC	64
II.4.3.3. Le système DRAMA	65
II.4.3.4. Le système DRCS.....	66
II.4.3.5. Le formalisme DIPA	67
II.4.4. Synthèse comparative des différentes méthodes de formalisation des connaissances	68
II.4.4.1. Méthodes et Outils.....	68
II.4.4.2. Types de Méthodes.....	68
II.4.4.3. Recueil et sources de Connaissances.....	69
II.4.4.4. Types de Connaissances manipulées.....	70
II.4.4.5. Représentation des Connaissances	70
II.4.4.6. Choix d'un outil pour la formalisation de la mémoire de projet	71
II.5. Du KM à la Mémoire de projet	72
II.5.1. Besoins en Mémoire de projet.....	72
II.5.1.1. Introduction :	72
II.5.1.2. Quelques définitions :	72
II.5.1.3. La mémoire de projet : un outil de gestion des projet ou des entreprises.....	73
II.5.1.4. Les fonctionnalités que doivent assurer une mémoire de projet	74
II.5.1.5. Les bénéfices attendus de la mémoire de projet	75
II.5.1.6. Etat de l'art de la mémoire de projet	77
II.5.1.7. Conclusion	79
II.5.2. Les Systèmes de Gestion des Données Techniques	80
II.5.2.1. Son origine.....	80
II.5.2.2. Fonctionnalités du SGDT	81
II.5.2.3. Les Avantages d'utilisation des SGDT	84
II.5.2.4. Les limites d'un SGDT.....	85
II.5.2.5. Conclusion	86
II.5.3. Traçabilité.....	87
II.5.3.1. Quelques Définitions	87
II.5.3.2. La traçabilité dans le domaine de conception.....	87
II.5.3.3. Les Domaines concernés par la traçabilité	87

II.5.3.4. Conclusion	91
II.6. Conclusion	92
CHAPITRE III : MODELES ET STRUCTURES PROPOSES POUR LA MODELISATION DE LA MEMOIRE DE PROJET	93
III.1. Introduction	93
III.2. Architecture Générale du Système d'Information	93
III.2.1. Niveaux de détail	93
III.2.2. Les paquetages	94
III.3. Présentation des Modèles	97
III.3.1. Le Patron de Conception	97
III.3.2. Modèle d'Organisation	98
III.3.3. Modèle de Processus	102
III.3.4. Le modèle de Produit	103
III.3.5. Modèle de Fonctions du Produit	105
III.3.6. Les Documents	107
III.3.7. Lien entre les différents modèles	111
III.3.8. Modèle de Paramètres	113
III.3.9. Modèle de Contraintes	115
III.3.10. Modèles de Variantes vs Versions	117
III.3.11. Le Modèle des Relations «Evoluées» entre les différents éléments de la mémoire de projet.....	121
III.3.12. Le Modèle des Ressources	134
III.3.13. Le Modèle de Justification	137
III.3.14. Le Modèle des points de Vue	141
III.3.15. Le Modèle des Objectifs	144
III.3.16. Le Modèle de Projet	146
III.4. Conclusion.....	148
CHAPITRE IV : VERS UN OUTIL DE DEFINITION ET DE REUTILISATION DE LA MEMOIRE DE PROJET : P2M2.....	149
IV.1. Introduction.....	149
IV.2. Outils utilisés pour le développement de P2M2 :	150
IV.2.1. Les choix technologiques.....	150
IV.2.2. Choix des modèles implémentés.....	151
IV.2.3. Règles de transformation des modèles : des modèles de classes UML en objets persistants Oracle10g	151
IV.2.4. Architecture de P2M2	162
IV.2.5. P2M2 : Fonctionnalités et interfaces.....	165
IV.3. Validation des modèles auprès d'une entreprise d'ingénierie : Cas de la STUDI.....	172
IV.3.1. Présentation de la société STUDI	172
IV.3.2. Démarche pour la mise en place de la mémoire de projet chez STUDI	173
IV.3.2.1. Notion de Projet-Processus chez STUDI	173
IV.3.2.2. Notion de Produit chez STUDI.....	178
IV.3.2.3. Notion d'Organisation chez STUDI	180

IV.3.2.4. Bilan d'adaptation des modèles chez STUDI	181
IV.4. Conclusion	182
CONCLUSION	183
REFERENCES BIBLIOGRAPHIQUES	185

Liste des Figures

Figure 1 : Pôles majeurs que l'entreprise doit prendre en compte.....	15
Figure 2 : Quelques définitions.....	16
Figure 3 : Le Knowledge Management	18
Figure 4 : Activité de conception : processus de résolution de problèmes	28
Figure 5 : Technologies et Approches influençant la construction d'un SKM	42
Figure 6 : Classification des Systèmes de Gestion des Connaissances	45
Figure 7 : Un exemple d'un élément d'expérience dans le domaine de l'analyse d'accident	52
Figure 8 : Les trois perspectives : tâche, méthode et information.....	53
Figure 9 : Les trois visions définies dans SAGACE.....	55
Figure 10 : Trois éléments qui représentent les données dans les visions.....	56
Figure 11 : Le triangle sémiotique.....	57
Figure 12 : les six modèles CommonKADS.....	60
Figure 13 : Les principaux éléments de la méthode IBIS	62
Figure 14 : L'interface principale de gIBIS.....	63
Figure 15 : Représentation de la prise de décision selon QOC.....	64
Figure 16 : Arbre de solution.....	65
Figure 17 : Modèle d'Argumentation	66
Figure 18 : Méthodes et Outils de recueil des Connaissances.....	68
Figure 19 : Domaines d'application des méthodes de Capitalisation.....	68
Figure 20 : Techniques de Recueil des Connaissances.....	69
Figure 21 : Types de Connaissances manipulées.....	70
Figure 22 : Types de représentation des Connaissances.....	70
Figure 23 : Illustration des fonctionnalités d'une mémoire de projet par un diagramme de cas d'utilisation.....	74
Figure 24 : Illustration des fonctionnalités d'un Système de Gestion des Données Techniques...83	83
Figure 25 : Les domaines concernés par la traçabilité	88
Figure 26 : Architecture Globale du Système d'Information.....	95
Figure 27 : Patron décrivant une décomposition arborescente.....	98
Figure 28 : Diagramme de classes de description de l'organisation.....	99
Figure 29 : Diagramme d'objets de décomposition d'une organisation fonctionnelle hiérarchique (département-sous-département).....	100
Figure 30 : Diagramme d'objets d'une organisation matricielle (département – projet).....	101
Figure 31 : Diagramme de classes de description de Processus.....	102
Figure 32 : Diagramme d'objets d'un Processus.....	103
Figure 33 : Diagramme de classes de description de Produit.....	104
Figure 34 : Diagramme d'objets d'un Produit.....	105
Figure 35 : Diagramme de classes de description des Fonctions de Produit.....	106
Figure 36 : Diagramme d'objets de description des Fonctions de Produit.....	107
Figure 37 : Diagramme de classes de description de Document.....	108
Figure 38 : Diagramme d'objets de description de Document.....	109
Figure 39 : Diagramme de classes faisant le lien entre Acteur et Document.....	110
Figure 40 : Diagramme de d'objets de lien entre Acteur et Document.....	110
Figure 41 : Diagramme de classes de description de liens entre les différents modèles.....	111
Figure 42 : Diagramme d'objets de description de lien Produit-Processus.....	112
Figure 43 : Diagramme d'objets de description de lien Processus-Organisation.....	113

Figure 44 : Diagramme de classes de description de Paramètres.	114
Figure 45 : Diagramme d'objets de description de Paramètres.	115
Figure 46 : Diagramme de classes de description de Contraintes.	116
Figure 47 : Diagramme d'objets de description de Contraintes.	117
Figure 48 : Diagramme de classes de description des Variantes.	118
Figure 49 : Diagramme d'objets de description de Variantes.	119
Figure 50 : Diagramme d'objets de description de Variante.	120
Figure 51 : Diagramme de classes de description de Relations.	121
Figure 52 : Diagramme de classes de description des différents types de relations.	122
Figure 53 : Lien entre deux éléments n'appartenant au même paquetage sans historique	124
Figure 54 : Lien entre deux éléments du même paquetage sans historique.	124
Figure 55 : Lien entre deux éléments du même paquetage avec historique.	124
Figure 56 : Lien entre éléments qui ne sont pas du même paquetage avec historique.	125
Figure 57 : Diagramme d'objets UML de description de la relation du type Succession.	126
Figure 58 : Diagramme d'objets de description de la relation du type Composition.	127
Figure 59 : Diagramme d'objets de description de la relation du type Généralisation.	128
Figure 60 : Diagramme d'objets de description de la relation du type Composition-Variante. ...	129
Figure 61 : Diagramme d'objets de description de la relation du type Composition-Variante. ...	130
Figure 62 : Diagramme d'objets de description de la relation du type Composition-Evolution. ...	131
Figure 63 : Diagramme de classes de description de type Variante-Evolution.	131
Figure 64 : Diagramme d'objets de description de type Composition-Xor.	132
Figure 65 : Diagramme d'objets de description de type Composition-And.	133
Figure 66 : Diagramme de classes de Ressource.	135
Figure 67 : Diagramme d'objets de description des ressources.	136
Figure 68 : Diagramme de classes de la Justification.	138
Figure 69 : Diagramme d'objets d'évolution de deux éléments du type Variante.	138
Figure 70 : Diagramme d'objets de description de la Variante V1	139
Figure 71 : Diagramme d'objets de description de la Variante V2.	140
Figure 72 : Traçabilité au niveau des justifications.	140
Figure 73 : Diagramme d'objets décrivant l'ensemble des relations d'évolution.	140
Figure 74 : Diagramme de classes de description de Point de Vue (Version 1).	141
Figure 75 : Diagramme de classes de description de Point de Vue (Version 2).	142
Figure 76 : Diagramme de classes de description de Point de Vue (Version 3-1).	142
Figure 77 : Diagramme de classes de description de Point de Vue (Version 3-2).	143
Figure 78 : Diagramme d'objets de description de Point de Vue (Version 3).	144
Figure 79 : Diagramme de classes de description des objectifs.	145
Figure 80 : Diagramme d'objets de description des Objectifs.	146
Figure 81 : Diagramme de classes de Projets.	147
Figure 82 : Diagramme d'objets partiel de description d'un Projet.	147
Figure 83 : Exemple d'un diagramme de classes utilisant l'association du type 1..*	152
Figure 84 : Schéma de transformation d'un diagramme de classe utilisant l'association du type 1..*	153
Figure 85 : Diagramme de classes utilisant un lien de composition monovaluée.	154
Figure 86 : Schéma de transformation d'un diagramme de classes utilisant l'association du type composition monovaluée.	154
Figure 87 : Diagramme de classes utilisant un lien de composition multivaluée.	155

Figure 88 : Schéma de transformation d'un diagramme de classe utilisant l'association du type composition multivaluée.	156
Figure 89 : Diagramme de classes utilisant une association réflexive.	157
Figure 90 : Schéma de transformation d'un diagramme de classes utilisant l'association réflexive.	157
Figure 91 : Diagramme de classes utilisant une association réflexive.....	158
Figure 92 : Schéma de transformation d'un diagramme de classes utilisant l'association réflexive (Composition multivaluée).....	159
Figure 93 : Diagramme de classes utilisant un lien d'héritage.	160
Figure 94 : Schéma de transformation d'un diagramme de classe utilisant l'héritage	160
Figure 95 : Comparaison entre le modèle horizontal et le modèle plat.....	162
Figure 96 : Modèle d'architecture multi-niveaux d'application J2EE de la mémoire de projets	164
Figure 97 : Interface de description de la liste des produits.....	166
Figure 98 : Interface de description de l'arborescence du produit « Turbopompe ».	166
Figure 99 : Interface de description de l'arborescence d'un document.	167
Figure 100 : Interface de gestion de l'arborescence d'un document.....	168
Figure 101 : Interface de description des variantes.	169
Figure 102 : Interface d'affichage de la variante V1012.	170
Figure 103 : Interface d'Affichage de la Variante V1013.	171
Figure 104 : Interface d'affichage de la liste des évolutions existant dans la mémoire de projet avec une description de chacune des évolutions.	172
Figure 105 : Diagramme d'activités d'un Projet chez STUDI (Etape phase d'étude).	174
Figure 106 : Diagramme de description des notions Projet-Processus chez STUDI.	177
Figure 107 : Diagramme objet du concept Projet chez STUDI.	178
Figure 108 : Diagramme de description du concept Produit chez STUDI.....	179
Figure 109 : Diagramme objet du concept Projet chez STUDI.	180
Figure 110 : Diagramme de description du concept Organisation chez STUDI.....	180
Figure 111 : Diagramme objet du concept Projet chez STUDI.	181

INTRODUCTION GENERALE

Au cours des dernières décennies, l'intérêt porté aux problématiques du management des connaissances et des savoir-faire s'est accru, les entreprises prenant progressivement conscience du capital que représente la matière grise de ses collaborateurs. Si le développement des technologies de l'information et de la communication a ouvert de nouvelles perspectives, les dimensions organisationnelle et managériale restent essentielles pour appréhender une véritable culture de partage de la connaissance. L'idée principale sous-jacente au concept de management de connaissances, plus connu sous le terme anglo-saxon de « Knowledge Management » (KM), est que « la connaissance constitue une ressource critique et un facteur de compétitivité déterminant pour un pays, une industrie, une entreprise » [Drucker, 1993]. En conséquence, les pratiques de KM s'avèrent primordiales : comment acquérir, créer, partager, distribuer et accéder aux connaissances à travers les processus organisationnels, autant de questions auxquelles le KM doit répondre.

Le KM (capitalisation des connaissances, gestion des connaissances) présente une double dimension. D'une part, il présente un enjeu pour le management des ressources humaines, les pratiques organisationnelles et managériales et la culture d'entreprise et d'autre part les technologies d'information et de communication jouent un rôle crucial pour plusieurs aspects du KM ; une démarche de KM est donc par essence pluridisciplinaire.

Les technologies supportant le KM sont généralement construites autour de ce qu'on appelle les mémoires d'entreprise.

Plusieurs typologies de connaissances dans l'entreprise ont été proposées dans la littérature. Elles peuvent être utiles pour déterminer les connaissances essentielles à capitaliser par l'entreprise. Les travaux de Nonaka et Takeuchi [Nonaka et al., 1995] sont d'une grande contribution dans ce domaine. Ils distinguent les connaissances tacites et les connaissances explicites : les connaissances tacites ont un aspect personnel qui les rend difficiles à formaliser et à communiquer alors que les connaissances explicites sont les connaissances transmissibles dans un langage formel.

Dans [Van Heijst et al., 1996] la mémoire d'entreprise est définie comme la « représentation explicite, persistante, et désincarnée, des connaissances et des informations dans une organisation ». Elle peut inclure par exemple, les connaissances sur les produits, les procédés de production, les clients, les stratégies de vente, les résultats financiers, les plans et buts stratégiques, des bases de données, des documents électroniques, des rapports, des spécifications de produit, la logique de conception, etc.

La construction d'une mémoire d'entreprise permet de « préserver afin de les réutiliser plus tard ou le plus rapidement possible, les raisonnements, les comportements, les connaissances, même en leurs contradictions et dans toute leur variété » [Pomian, 1996].

Il existe des méthodes de gestion des connaissances permettant la définition d'une mémoire d'entreprise. Elles sont classées en deux grandes catégories : les méthodes de capitalisation des connaissances et les méthodes d'extraction des connaissances :

- Les méthodes de capitalisation des connaissances : ces méthodes d'ingénierie des connaissances sont issues de 20 ans de pratique, d'abord pour l'élaboration de systèmes experts (systèmes informatiques reproduisant le raisonnement d'experts), puis elles ont été transposées pour rédiger des livres de connaissances capitalisant le savoir-faire critique des collaborateurs, puis complétées afin de réaliser des bases de connaissances vivantes constituant la mémoire collective commune du savoir-faire d'une organisation.

- Les méthodes d'extraction des connaissances : ce sont des méthodes qui visent à extraire les connaissances de l'activité de l'organisation. Il s'agit par exemple des méthodes de fouille de données, de fouille de texte, de traçabilité de communication (email, forum de discussion, etc.) et de la logique de conception.

L'ingénierie des connaissances (IC) est définie comme un processus de modélisation des connaissances, individuelles ou collectives, explicites ou implicites, stabilisées ou évolutives, expertes ou techniques [Charlet et al., 1999]. Elle permet de rendre ces connaissances accessibles sous une forme définie en fonction du contexte, opérationnel ou non. Pour ce faire, plusieurs méthodes telles que CommonKADS, MASK, CYGMA [Dieng et al., 2000] ont été définies. Ces méthodes permettent de construire des mémoires d'entreprise modélisant des connaissances relatives à des activités au sein d'une organisation.

De nos jours des défaillances liées à des pertes de savoir-faire (départs en retraite, turn-over) sont observées dans les entreprises mettant en évidence la connaissance au centre du processus de conception comme un élément conditionnant la compétitivité des entreprises. S'il existe des outils permettant de capitaliser des expertises particulières, très peu sont capables d'assurer la capitalisation des connaissances dans les projets menés au sein des organisations. Face à ce manque, **l'objectif principal de nos travaux est de spécifier et de modéliser un outil de gestion des connaissances pour des projets de conception.** Il s'agit de proposer un modèle d'architecture de domaine permettant, par instanciation et adaptation à un contexte particulier, de réaliser une mémoire de projet.

Un des objectifs majeur de la construction d'une mémoire de projet est de permettre la réutilisation de solutions développées dans des projets passés. De telles mémoires de projet devraient répondre au moins à deux besoins identifiés des concepteurs en situation de réutilisation : accéder aux résultats des projets passés et à la logique de conception qui a conduit à la conception de ce produit. D'après [Matta et al., 2000], cette logique de conception constitue l'ensemble des leçons et expériences vécues lors de la réalisation de projet. Il s'agit de remonter dans l'histoire afin de tracer et comprendre les différents choix et arguments acceptés et les solutions rejetées par les concepteurs lors des étapes de conception. Cette traçabilité devient par ailleurs une exigence des démarches qualité, permettant par exemple d'établir les responsabilités en cas de défaillance du produit en phase d'exploitation.

Il existe plusieurs méthodes de formulation de la logique de conception dans la littérature exemple la méthode QOC [McLean et al., 1991], la méthode IBIS [Buckingham Shum, 1997], etc. Ces méthodes permettent de garder une trace de résolution collective de problèmes, extraite spécialement des réunions de prise de décision. L'inconvénient de ces méthodes est qu'elles ne couvrent pas tous les aspects d'un projet à savoir son contexte et sa logique de conception.

Il est à noter qu'il existe très peu de travaux dans la littérature à propos de la mémoire de projet.

Notre apport consiste à proposer un ensemble de modèles qui prennent en compte à la fois le contexte et la logique de conception. L'étude du contexte de projet nous a conduit à élaborer des modèles par exemple d'organisation, des ressources, des acteurs, des produits, processus etc. Quant à la logique de conception, elle nous a conduits à élaborer des modèles de justification, idées, arguments.

Le standard UML [OMG, 2005] a été utilisé comme langage de modélisation. Nous avons d'abord établi les différents diagrammes de classes et d'objets relatifs aux éléments constituant la mémoire de projet. Nous les avons regroupés en paquetages afin de structurer le système d'information dédié à la mémoire de projet.

Nous avons élaboré l'architecture de domaine afin de représenter l'ensemble des composants de la mémoire de projet et leurs relations. L'approche orientée objet utilisant UML que nous avons

retenue est bien adaptée pour modéliser ces éléments du fait de la relative lisibilité des modèles pour les utilisateurs non avertis, de la capacité de ces modèles à représenter assez fidèlement les éléments du monde réel, et des méthodes issues du génie logiciel permettant de passer efficacement des modèles à une implémentation informatique.

Notre architecture de domaine est conçue pour être une base d'intégration des différents modèles : l'ajout d'un nouveau composant peut s'effectuer sans impact sur les modèles existants ; il suffit d'exploiter les relations existantes pour le lier au reste des modèles.

Le système que nous proposons est basé sur une architecture à plusieurs niveaux, et tire profit du niveau des données d'un système de gestion des bases de données objet-relationnel, permettant ainsi d'avoir une structure de mise en œuvre flexible qui peut être facilement maintenable. De telles architectures permettent aussi d'accéder à la mémoire de projet selon des points de vue différents.

Pour appliquer notre méthode, nous avons développé l'outil « P2M2 » (Project Product Memory Management system) permettant l'archivage et la réutilisation des éléments de la mémoire de projet. Cet outil offre des interfaces permettant l'accès aux données selon le besoin des différents utilisateurs.

Notre méthode de développement, ainsi que l'outil développé, ont été appliqués à une expérience pratique afin de pouvoir adapter les modèles conçus dans le cadre d'une entreprise d'ingénierie. Ceci nous a permis de tester nos modèles et de mesurer le degré de facilité d'adaptation.

Le document est structuré de la façon suivante :

Nous nous intéressons dans le chapitre 1 de cette thèse au contexte ainsi qu'à la problématique liée à la gestion des connaissances et la mémoire d'entreprise d'une part et l'activité de conception d'autre part. Nous présentons un ensemble de définitions de la notion de connaissance avant d'aborder la gestion des connaissances en conception et en gestion de projets. Nous décrivons un exemple type, le projet de la société européenne de propulsion (SEP), révélateur du besoin d'une mémoire de projet en entreprise. Un état de l'art de l'activité de conception est présenté afin de cerner l'existant dans le domaine de l'activité de conception qui nous intéresse et de positionner nos travaux.

Le chapitre 2 présente un état de l'art sur la gestion des connaissances (KM) en précisant les enjeux et obstacles que peut avoir une démarche KM au sein de l'entreprise. Nous poursuivons avec la présentation des Systèmes du Knowledge Management (SKM). Nous dressons par la suite un état de l'art des méthodes de capitalisation des connaissances, puis des mémoires de projet. Une étude comparative du point de vue des fonctionnalités entre les Systèmes de Gestion des Données Techniques (SGDT) et la mémoire de projet est également présentée.

Dans le chapitre 3, nous présentons l'architecture générale de notre système d'information puis les patrons de conception, sur lesquels s'appuie en grande partie la construction des différents modèles de la mémoire de projet. Nous présentons ensuite comment nous gérons l'historique des éléments d'un projet.

Dans le chapitre 4, l'architecture de déploiement des modèles est présentée afin de décrire l'outil-prototype dédié que nous proposons pour la mémoire de projet. Nous présentons la structure du système d'information après avoir précisé notre choix quant au type de système de gestion de base de données adopté. Nous précisons les limitations des AGL (Ateliers de Génie Logiciel) actuelles afin d'aider à la transformation des modèles dans un contexte objet-relational et une approche de déploiement des modèles est proposée. Nous avons mené une expérience pratique au sein d'une société d'ingénierie la STUDI (Société Tunisienne de Développement et d'Ingénierie); la démarche d'adaptation des modèles et les difficultés rencontrées sont présentées. En regard de l'architecture MDA (Model Driven Architecture) de l'OMG, nos modèles (paquetages, classes, objets) se situent au niveau M1. La définition d'un méta modèle dédié à la mémoire de projet au niveau M2 de l'architecture MDA pourra éventuellement mieux généraliser nos modèles conçus afin de les adapter pour n'importe quel domaine d'application. Nous concluons en synthétisant l'apport de notre travail et en proposant quelques perspectives de recherche.

CHAPITRE I : CONTEXTE ET MOTIVATIONS

I.1. Introduction

L'ensemble des connaissances d'une entreprise forme un capital intellectuel très volatile. Pour Jean-Louis Ermine, la *gestion des connaissances* « est revenue d'outre-atlantique, sous le vocable de Knowledge Management (KM), avec une vigueur et une force qui en font un des mots-clés des entreprises actuellement » [Ermine, 2003]. C'est dire que pour lui il s'agit aussi d'un mot « made in USA » pour décrire des approches, démarches, méthodes, outils, etc.

Le KM englobe la gestion du savoir et savoir-faire également le savoir-faire développé par le personnel, de manière à créer un système interactif de formation continue qui débouche sur une meilleure qualité des produits et services, ainsi que sur une plus grande compétitivité de l'entreprise.

Le KM permet de sauvegarder ce savoir et/ou savoir-faire. En favorisant la diffusion des informations, il favorise la transversalité aux dépens de la hiérarchisation. Il accroît la compétitivité de l'entreprise.

Tout salarié quitte un jour son entreprise : Passage à la concurrence, réduction des effectifs, départ en retraite, etc. Pour l'entreprise, la perte est minorée si elle a su recueillir les connaissances, les expériences de l'intéressé. En archivant sa propre mémoire, l'entreprise évite aussi la répétition d'erreurs du passé ; le partage de la connaissance permet à chaque acteur d'améliorer ses performances et donc celles de l'entreprise. Ce concept du KM n'est pas limité à un secteur de l'entreprise ni même à un type d'entreprise.

Une grande partie de l'information « capital de l'entreprise », n'est pas intégrée dans la partie informatisée de son système d'information. Elle se trouve sur des documents papiers ou dans les cerveaux des experts ; dans ce dernier cas, elle n'est donc pas exploitable par les autres.

La mise en place d'un programme de gestion des connaissances demande du temps. Il faut recueillir, et de façon permanente, la matière première. Les connaissances tacites doivent être formalisées pour pouvoir jouer leur rôle de données pour les processus (informatisés ou à informatiser) de l'entreprise.

La retranscription des informations ne peut être exhaustive mais doit être suffisamment fine pour rester utilisable. Le volume d'informations s'accroissant, il faut maîtriser les outils permettant de la traiter, de la synthétiser, d'en extraire les éléments nécessaires à son utilisation, d'effectuer les recherches pertinentes.

Il est donc important de mettre en place une organisation apprenante comprenant notamment une infrastructure informatique capable de travailler dès l'amont du processus, et donc de traiter l'ensemble de la chaîne de production et de traitement de la connaissance.

La sensibilisation des salariés est primordiale, pour qu'ils mettent à disposition leur savoir comme pour aller chercher ce que d'autres ont fourni. Certaines entreprises choisissent même de motiver les salariés en les rémunérant en fonction de leur participation à l'enrichissement de la mémoire de l'entreprise.

L'ensemble des informations recueillies devient rapidement très volumineux. La diffusion par intranet et sur abonnement est une façon de responsabiliser les acteurs, de les pousser à faire la démarche de rechercher l'information.

Plus l'information est partagée par l'entreprise, plus elle l'aide à progresser. A l'opposé de cette façon de penser on peut rencontrer l'attitude qui consiste à croire que l'information est synonyme de pouvoir. Trop d'entreprises très hiérarchisées sont sclérosées par des niveaux intermédiaires qui bloquent l'information. Cette rétention leur apparaît comme le meilleur moyen d'affirmer leur pouvoir et de justifier leur poste. Au contraire, la gestion et la diffusion des connaissances accompagne naturellement l'avènement des nouvelles organisations : en réseau, par projets, par des équipes pluridisciplinaires.

Le partage et la capitalisation transversale de la connaissance se concrétisent rapidement par des résultats tangibles. Notamment, pour ce qui concerne la conception de produit, l'entreprise voit augmenter sa capacité d'innovation, car elle sait créer le bon produit au bon moment. Elle anticipe mieux les changements, améliore sa réactivité, se montre capable de modifier son organisation et de faire évoluer ses produits. Les informations recueillies par une veille efficace sur les concurrents sont automatiquement intégrées. La formalisation et la numérisation du capital connaissance débouchent pratiquement sur une gestion en temps réel. Forte de tous ces progrès l'entreprise augmente ainsi sa compétitivité sur son marché.

I.2. La connaissance, un enjeu pour l'entreprise

I.2.1. Contexte et défis

Sous la pression d'une concurrence internationale de plus en plus vive, l'entreprise cherche à améliorer sans relâche le triptyque Qualité Coûts Délais, ce qui l'amène à optimiser son organisation et à modifier en profondeur la culture de ses différents acteurs. Si elle veut se développer, l'entreprise doit tenir compte d'au moins trois pôles majeurs dont les intérêts divergent (Figure 1).

En premier lieu, l'entreprise propose un produit et/ou un service à des clients réels et potentiels ; elle se doit de leur faire une offre correspondant à un besoin, ceci à un prix compétitif. Dans ce contexte, on voit par exemple les constructeurs automobiles proposer à leurs clients des gammes de véhicules renouvelées de plus en plus rapidement, possédant des fonctions nouvelles, associées à des offres de service très riches (financement, entretien, assurance...).

En second lieu, pour prospérer l'entreprise doit dégager un profit, faute de quoi elle disparaît. Bien qu'il s'agisse d'une évidence, il ne faut pas oublier la véritable mutation culturelle (passage à une logique concurrentielle) que doivent opérer certaines entreprises.

Le troisième pôle enfin est celui du personnel, les collaboratrices et collaborateurs de l'entreprise, qui doivent s'épanouir dans leur travail, progresser, se sentir véritablement acteurs des progrès de l'entreprise ; la mise en pratique de tels principes se heurte encore aujourd'hui à de nombreux obstacles.

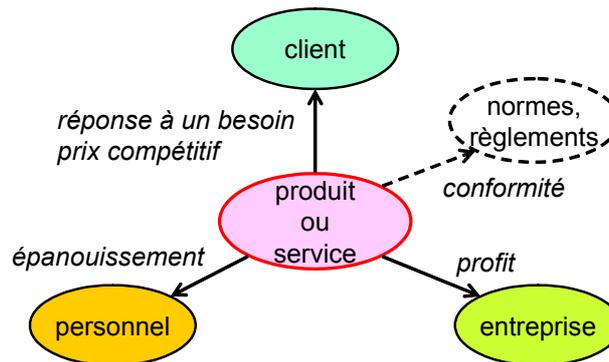


Figure 1 : Pôles majeurs que l'entreprise doit prendre en compte.

Ces exigences, auxquelles il convient d'ajouter le respect de normes précises en termes de qualité, sécurité et de protection de l'environnement, obligent les entreprises à prendre en considération de nombreux paramètres internes et externes. La maîtrise de ces paramètres demande l'intervention de disciplines nombreuses et variées issues des sciences économiques, des sciences humaines et sociales, des sciences de l'ingénieur. Pour améliorer son efficacité, l'entreprise est souvent amenée à remplacer une organisation vue principalement sous l'angle de structures « métiers » verticales au profit de processus transversaux orientés vers le client [Midler, 2002].

Il devient donc nécessaire pour l'entreprise d'assurer la coordination de tous ses moyens (méthodes, outils, modèles) et activités hétéroclites pour les mettre en synergie, ce qui constitue un véritable défi compte tenu de la complexité croissante des systèmes industriels.

Les produits évoluent progressivement vers les services, les métiers s'ouvrent à de nouveaux domaines ; à cela s'ajoute l'explosion de l'information : denrée rare et mal répartie il y a peu, elle est aujourd'hui distribuée partout dans le monde et nous submerge.

Ainsi, dans le cadre de la conception de produits et de services, au moins trois défis majeurs se doivent d'être relevés par l'entreprise : i) la maîtrise de la connaissance, ii) l'innovation et iii) la pro-activité.

- La maîtrise de la connaissance possède au moins deux facettes : la veille technologique et le management des connaissances internes.
 - La veille technologique consiste pour l'entreprise à se tenir informée de manière constante et organisée des activités de ses concurrents, connaître en temps réel les nouveaux brevets et les nouvelles normes pouvant impacter son action, être à l'affût des nouveaux composants, des articles de la presse spécialisée... et être en mesure d'exploiter ces informations après les avoir conceptualisées. Comme on le voit, il s'agit typiquement d'un enjeu lié à la connaissance.

- Le management des connaissances internes est un processus de création, d'enrichissement, de capitalisation et de diffusion des connaissances impliquant tous les acteurs d'une entreprise, en tant que consommateurs et producteurs, qui débouche sur une meilleure qualité des produits et services, ainsi que sur une plus grande compétitivité de l'entreprise.
- L'innovation constitue un avantage concurrentiel majeur, à condition d'en maîtriser les risques ; l'entreprise doit se doter d'une véritable ingénierie lui permettant de mettre en œuvre de manière systématique la recherche de solutions innovantes, ainsi que la sélection des meilleures d'entre-elles. Si une part de l'innovation tient du « trait de génie », il s'avère que ses fondements sont souvent conditionnés par de solides connaissances du domaine.
- La pro-activité s'applique à la maîtrise des connaissances et à l'innovation ; il semble indispensable, pour préserver un avantage concurrentiel, non seulement de réagir aux événements, mais surtout de les anticiper [Prat et al., 2003].

Dans ce contexte, il devient nécessaire pour l'entreprise de se doter de nouveaux outils et modes d'organisation lui permettant de capter les informations opportunes (les rechercher, les filtrer), capitaliser les connaissances de l'entreprise, et les partager pour en générer de nouvelles.

I.2.2. Qu'est-ce que la gestion des connaissances ?

Avant de donner une définition de la gestion des connaissances, nous proposons dans la Figure 2 quelques définitions des termes : donnée, information, connaissance et compétence, issues de [Schreiber et al., 1999]. On pourra se référer à [Mekhilef et al., 2003] pour une terminologie très complète réalisée au plan européen¹.

Terme	Exemple	Définition
Donnée	Fait élémentaire, discret et objectif résultant d'une acquisition, d'une mesure effectuée par un instrument naturel ou construit par l'homme.
Information	SOS	Ensemble de données sémantiquement cohérentes pouvant être interprété pour lui donner un sens, dans une intention particulière.
Connaissance (syn. savoir)	Alerte : lancer les opérations de secours	Ensemble d'informations construites permettant de comprendre, et d'agir. La connaissance implique l'Homme porteur : elle est interprétée, appropriée, activable dans un but, contribue à une recherche de vérité.
Compétence (syn. savoir-faire)	Capacité à porter secours	Ensemble de connaissances, de capacités d'action et de comportements structurés en fonction d'un but et dans un type de situation donnée.

Figure 2 : Quelques définitions.

¹ The European KM Community ; <http://www.knowledgeboard.com>

Proposer une terminologie commune au KM est une tâche vraiment ambitieuse. En effet, ce concept émergent n'a pas encore connu de définition stable. Le KM répond à plusieurs définitions plus au moins étendues selon le domaine des intervenants qui l'utilisent. Certaines définitions peuvent parfois être réductrices et refléter les lacunes et erreurs du passé et du présent, en particulier quand elles sont véhiculées par certains éditeurs de logiciels, voulant profiter de la nouvelle vague.

De plus, l'hétérogénéité des discours marketing des différents acteurs souhaitant se positionner sur le marché prometteur du KM (éditeurs de logiciels, cabinets de conseils etc.) complique la situation en s'appropriant le concept de telle sorte qu'il reflète parfaitement leurs offres.

Nous avons sélectionné, dans cette section des définitions de KM proposées dans la littérature du KM. Notre objectif est de relever les principales questions qui résultent de tentatives typiques de définitions de KM.

[Skyrme, 1999] avance une définition qu'il veut générique du Knowledge Management : « il s'agit du management explicite et systématique de connaissances vitales et des processus qui y sont associés incluant la création, la collecte, l'organisation, la diffusion, l'utilisation et l'exploitation des connaissances. Le KM exige le passage des connaissances personnelles à des connaissances collectives pouvant être partagées largement dans une organisation ».

« La gestion des connaissances est l'objectif de formaliser les connaissances tacites afin de les rendre mobilisables et opérationnelles au niveau de l'organisation entière » [Ermine, 2000].

Pour Davenport (1994) «le Knowledge Management est un processus de capture, de diffusion, et d'utilisation efficace de la connaissance » [Firestone, 2001].

Pour Jean-Louis Ermine, la *gestion des connaissances* « est revenue d'outre-atlantique, sous le vocable de Knowledge Management, avec une vigueur et une force qui en font un des mots-clés des entreprises actuellement » [Ermine, 2003]. C'est dire que pour lui il s'agit aussi d'un mot « made in USA » pour décrire des approches, démarches, méthodes, outils, etc.

Karl-Erik Sveiby définit le Knowledge Management comme une discipline ayant deux volets :

- Knowledge Management = *Management des informations.*

Les praticiens et les chercheurs impliqués dans cette perspective préconisent l'utilisation des solutions technologiques pour gérer la connaissance systèmes d'information, Intelligence Artificielle etc. La connaissance est perçue comme un objet ou une entité statique qui peut être manipulée par un système d'information. Cette nouvelle perspective est en pleine expansion grâce aux développements fulgurants en matière de nouvelles technologies de l'information et de la Communication (NTIC)

- Knowledge Management = *Management des ressources humaines*

Les chercheurs et les praticiens de ce domaine sont essentiellement des philosophes, des psychologues, des sociologues ou des managers. Ils s'intéressent à l'évaluation et à l'amélioration des compétences, des habilités ainsi que du comportement des divers acteurs de l'organisation. La connaissance est perçue comme un processus, un ensemble complexe d'habilités, de savoir-faire, d'expertises, de compétences qui évoluent constamment.

La proposition de Karl Erik Sveiby est illustrée par la Figure suivante [Sveiby, 2000].

	Knowledge Management	
Pistes d'Activités	Management de l'information Connaissance = Objet	Management des Ressources Humaines Connaissances = Processus
Niveau Organisation	« Réingénierie »	« Théoriciens de l'organisation »
Niveau Individu	« Spécialiste Intelligence Artificielle »	« Psychologues »

Figure 3 : Le Knowledge Management [Sveiby, 2000].

Selon [Wiig, 1999], qui fut l'un parmi les auteurs les plus en vue dans le champ de KM le définit comme suit² : « le Knowledge Management concerne la compréhension, le management systématique, la construction explicite et délibérée de connaissances et son application. Il s'agit d'un management efficace des processus centrés connaissances ».

Il souligne que le KM doit être perçu selon quatre perspectives :

- la perspective technologique (IM & IT Focus) qui s'adresse à la capture, la manipulation et la localisation des connaissances ;
- la perspective organisationnelle (Enterprise Effectiveness Focus) qui concerne l'utilisation optimale des connaissances pour améliorer l'efficacité des processus opérationnels de l'organisation ;
- la perspective capitale intellectuelle (Intellectual Asset Focus) qui met en œuvre les méthodologies d'exploitation du capital intellectuel pour optimiser la valeur économique d'une organisation ;
- la perspective Humaine (People Focus) s'attache à créer un environnement de partage de connaissances pour les employés de l'organisation.

Swan définit le KM comme étant un « ensemble de démarches, méthodes et outils de création, collecte, formalisation, capitalisation, partage, et utilisation de connaissances. Sa finalité est de favoriser les processus collectifs d'apprentissage et d'innovation » [Swan et al., 1999].

Dave Snowden définit le Knowledge Management comme « l'identification, l'optimisation et le management actif du capital intellectuel, qu'il soit sous forme de connaissances explicites contenues dans des artefacts, ou sous forme de connaissances tacites possédées par des individus ou des communautés. L'optimisation du management des connaissances explicites est réalisée à travers la consolidation et l'amélioration de l'accès aux artefacts. L'optimisation du management des connaissances tacites est réalisée à travers la création de communautés partageant et cultivant ces connaissances. Le Knowledge Management, concerne la création de processus de management et d'infrastructures pour réunir les artefacts et les communautés dans une écologie

² KM is to understand, focus on, and manage systematic, explicit and deliberate Knowledge building, renewal and application-that is, manage effective Knowledge process (EKP).

commune qui va maintenir la création, l'utilisation et la rétention des connaissances » [Snowden, 2000].

Selon [Malhotra, 2000a], [Malhotra, 2000b] « le Knowledge Management est une discipline qui doit bénéficier de la synergie des capacités humaines en matière de créativité et d'innovation et de la puissance des technologies de l'information pour satisfaire les organisations quant à leurs besoins d'adaptation, de survie et de compétence dans un environnement marqué par une complexité et une concurrence de plus en plus accentuées ».

D'après [Ramon, 2001] « Le terme Knowledge Management est le processus à travers lequel une entreprise utilise son intelligence collective pour accomplir ses objectifs stratégiques ».
Cette définition conduit à souligner quatre dimensions importantes du KM :

- Le KM est un processus : ainsi il comprend des phases et des composants, il existe plusieurs approches de structuration de ce processus. D'une part, il s'agit d'un management des processus centrés sur la connaissance (Knowledge Processes) plutôt qu'une manipulation de connaissances en tant qu'objet, d'autre part, il s'agit de considérer toutes les phases de ce processus, et ne pas l'aborder en partie.
- Le terme utilise : implique que cette connaissance collective doit amener à l'action, ce qui implique qu'il existe des outputs et des performances à mesurer.
- Intelligence collective : cet aspect souligne l'importance du rôle du partage de connaissances au sein de communautés, car la connaissance résultante est bien plus riche que les connaissances individuelles.
- Le KM n'est pas une fin en soi, il est fortement lié aux objectifs stratégiques de l'organisation qui le pratique.

L'ensemble des définitions citées ci-dessus se focalisent principalement sur les concepts suivants d'une manière explicite ou implicite : processus, management, méthodes et, connaissance.

Notre contribution : le KM est un ensemble de processus d'organisation, d'outils technologiques et de méthodes permettant de gérer la connaissance à savoir sa création, sa collection, son organisation, son stockage, sa transformation, sa diffusion, son utilisation et sa diffusion au sein de l'entreprise.

I.2.3. Typologie des Connaissances

Connaissances explicites – Connaissances implicites

La plupart des auteurs présentent les connaissances selon une approche dichotomique : les connaissances tacites et les connaissances explicites. Cette classification a été popularisée par les deux chercheurs Ikujiro Nonaka et Hirotaka Takeuchi en 1995 dans leurs théories de création de connaissances: la spirale de connaissances. [Nonaka et al., 1995] et par suite Michael Polanyi (1996) [Polanyi, 1996]. Ils distinguent deux types de connaissances :

- Les connaissances explicites : « les connaissances explicites se réfèrent à la connaissance qui peut être exprimée sous forme de mots, de dessins, d'autres moyens « articulés » notamment les métaphores » [Polanyi, 1996].
- Les connaissances tacites : « les connaissances tacites sont les connaissances qui sont difficilement exprimables quelle que soit la forme du langage » [Polanyi, 1996].

Autres typologies

Il existe bien d'autres classifications des connaissances. Nous en énumérerons brièvement certaines, en remarquant que la distinction tacite/explicite demeure la plus citée.

Karl Popper distingue trois mondes de connaissances [Karl, 1972] :

- Premier Monde de la connaissance : il s'agit de l'ensemble des structures encodées dans les systèmes physiques (par exemple : le codage génétique dans l'ADN) cela permet à ces objets de s'adapter à leur environnement.
- Second monde de la connaissance : croyances validées dans les esprits humains, concernant le monde, le beau, le vrai.
- Troisième monde de la connaissance : formulations linguistiques validées concernant le monde, le beau, le vrai.

Michel Saloff-Coste dans son petit livre blanc du management des connaissances, distingue trois niveaux dans l'intégration d'une connaissance : savoir, savoir-faire, savoir-être [Saloff, 2000].

- Le savoir : une conceptualisation fondée sur l'abstraction et qui prétend à des énoncés universels. On la rencontre en philosophie et en science par exemple. Le savoir est clairement formulé et contextualisé dans l'ensemble des connaissances implicites nécessaires à sa bonne compréhension.
- Le savoir-faire : Le savoir-faire se développe parallèlement à l'expérience d'un collaborateur. Il résulte de la mise des connaissances acquises par ce dernier dans le cadre d'une activité particulière.
- Le savoir-être : au-delà du savoir-faire, quand on analyse finement les raisons de l'excellence dans une matière on découvre un véritable « savoir-être ». Ce savoir-être est souvent complètement implicite et donc non-dit. Il fait référence à un certain nombre de valeurs morales, éthiques ou esthétiques que la personne véhicule sans les avoir analysées. Ce sont des représentations du monde, des « paradigmes » qui sont transmis de manière informelle, le plus souvent à travers les relations sociales et culturelles, ou qui sont élaborées plus ou moins inconsciemment par l'individu ou l'entreprise dans un désir de se dépasser.

Dans le contexte des études organisationnelles contemporaines, Blackler [Blackler, 1995], fournit cinq types de connaissances:

- « Embrained Knowledge » : Ce sont des connaissances abstraites fortement dépendantes des compétences conceptuelles et des habilités cognitives d'un individu.
- « Embodied Knowledge » : Il s'agit des connaissances personnifiées. Cette catégorie de connaissances est orientée vers l'action. Elle dépend de la présence physique d'un individu, de sa sensibilité et de sa personnalité. Ce type de connaissances est acquis par l'apprentissage dans des contextes spécifiques.

- « Encultured Knowledge » : Elles se réfèrent au processus par lequel les membres d'un groupe atteignent le niveau d'une compréhension commune et partagée. C'est un processus fortement dépendant des valeurs culturelles et sociales des acteurs qui y participent. Il peut être supporté par des formulations explicites d'idées, par des métaphores, par des discussions de groupes etc.
- « Embedded Knowledge » : Il s'agit des connaissances imbriquées. Elles résident dans des routines organisationnelles systématiques. Elles sont analysées par l'exploration des relations existantes, par exemple, entre les technologies, les rôles des acteurs, les processus, les procédures formelles et les routines organisationnelles émergentes. Au final, ce sont des connaissances émergentes des interactions de l'intégralité des éléments organisationnels.
- « Encoded Knowledge » : Il s'agit des connaissances codées. Ce sont des informations acheminées sous forme de symboles, elles prennent forme dans des documents, des manuels, des livres, des rapports etc. et désormais sous forme électronique.

Dans ses études sur le management stratégique, Spender [Spender, 1998] suggère une vue pluraliste des connaissances organisationnelles, il en catégorise quatre:

- « Conscious Knowledge » : Ce sont les connaissances conscientes. C'est la connaissance explicite détenue par l'individu.
- « Objectified Knowledge » : Ce sont les connaissances objectivées c.à.d les connaissances explicites détenues par l'organisation.
- « Automatic Knowledge » : Ce sont les connaissances automatiques. C'est celles que l'individu possède de manière inconsciente.
- « Collective Knowledge » : Ce sont les connaissances collectives. Elles sont fortement dépendantes de son contexte de création et souvent manifestées sous forme de pratiques organisationnelles.

Quant aux auteurs Nahapiet et Ghoshal [Nahapiet et al., 1998], ils distinguent deux catégories de connaissances : les connaissances pratiques, basées sur l'expérience et les connaissances théoriques dérivées à partir de réflexions et d'abstractions faites sur ces expériences.

Cette typologie rejoint la classification de connaissances proposée par Ryle [Ryle, 1949], qui différencie les connaissances procédurales des connaissances déclaratives.

Les connaissances procédurales concernent les compétences, les routines organisationnelles et le savoir-faire. Les connaissances déclaratives concernent le développement de faits et de propositions.

Ces différentes manières de classer et typer les connaissances, ne sont évidemment pas les seules, il suffit de se reporter par exemple à [Cook et al., 1999] [Tsoukas, 2002] pour découvrir d'autres typologies. Elles correspondent souvent à de légers changements de points de vues sur l'objet, mais la distinction explicite/tacite demeure toujours perceptible.

I.2.4. Objectifs pour l'entreprise

Nous avons souligné la difficulté de définir le champ couvert par le management des connaissances et la classification des différents types de connaissances. Pour l'entreprise, les objectifs dans ce domaine peuvent être divers et de portée très variable.

Il peut s'agir de partager une expertise, lorsque le spécialiste n'est pas disponible en permanence ou que le risque lié à son départ devient trop élevé pour l'entreprise.

Dans d'autres cas, l'entreprise peut souhaiter définir de meilleures pratiques dans le cadre de sa production ou de ses services, ou encore améliorer ses activités de conception et développer sa capacité à innover.

Elle peut également s'attacher à constituer une mémoire de ses projets afin d'en raccourcir les délais tout en augmentant la qualité du livrable.

Plus globalement, elle peut développer une véritable organisation apprenante en engageant une évolution culturelle profonde de ses collaborateurs (trices).

Si l'on applique l'outil QQQQCP au management des connaissances dans cette dernière optique, on peut répondre aux questions comme suit :

- Qui ? Le KM concerne potentiellement tous les acteurs de l'entreprise ;
- Quoi ? On s'intéresse exclusivement à la connaissance utile au développement de l'entreprise, en lien avec ses objectifs stratégiques ;
- Où ? Là où la connaissance est requise, ce qui nécessite d'établir une cartographie des compétences de l'entreprise et des connaissances sur lesquelles elles s'appuient ;
- Quand ? Dans la continuité, ce type de démarche s'apparentant davantage aux approches qualité (donc dans la durée) qu'au re-engineering (modification brutale et profonde) ;
- Comment ? Il s'agit de mettre en œuvre des démarches adaptées au contexte ;
- Pourquoi ? Parce qu'il faut préserver le capital intellectuel de l'entreprise.

La connaissance constitue un levier stratégique pour l'entreprise, car c'est à la fois un capital économique, une ressource stratégique, un facteur de stabilité et un avantage concurrentiel décisif.

Déjà, un certain nombre de fonctionnalités du KM commencent à être intégrées dans des environnements tels que les ERP (Entreprise Resource Planning), les EPM (Entreprise Performance Management) ou les outils de SCM (Supply Chain Management) [Tiwana, 2001].

I.3. Mémoires d'entreprise et capitalisation des connaissances

I.3.1. Définitions

Une mémoire d'entreprise a pour objectif de recueillir, afin de les préserver et de les transmettre, les savoirs acquis par une organisation, savoirs qui comprennent à la fois les connaissances détenues par les acteurs et les documents produits ou utilisés par l'entreprise.

La mémoire d'entreprise est donc un recueil des connaissances, « un ensemble structuré de connaissances attaché à l'expérience de l'entreprise dans un domaine donné » [Simon, 1996].

[Euzenat, 1996] Considère lui aussi la mémoire d'entreprise comme « une base de connaissances et du savoir-faire d'un ensemble d'individus travaillant dans une entreprise spécialisée ».

De la même manière, [Van Heijst et al., 1996] définit ainsi la mémoire d'entreprise: « Une mémoire d'entreprise est la représentation de connaissances explicites, désincarnées et persistante dans une organisation. (...). N'importe quelle partie de la connaissance ou de l'information qui contribuent à la performance de l'organisation peut (et peut être doit) être stockée dans la mémoire d'entreprise. Ceci inclut les connaissances à propos des produits, des processus de production, clients, stratégies du marché, résultats financiers, plans stratégiques et les objectifs etc. ».

Les connaissances d'une société, et sa capacité à les gérer, sont considérées comme des atouts majeurs. L'objectif est donc bien d'améliorer la compétitivité de cette entreprise, en évitant par exemple la répétition d'erreurs déjà commises : « Une mémoire d'entreprise intègre des données du produit avec de la connaissance (à traduire : product legacy) afin d'amener les expériences accumulées auparavant à peser sur les nouvelles tâches. De cette façon une répétition des erreurs peut être évitée, le savoir-faire est améliorée systématiquement » [Kühn et al., 1997], ou en permettant à l'entreprise de s'adapter dans de bonnes conditions au contexte: « Pour améliorer la puissance compétitive en continuant l'adaptation de l'organisation à l'environnement externe (le marché, le climat, les préférences clientes sociaux et politiques » [Van Heijst et al., 1996].

D'après [Pomian, 1996] la mémoire d'une entreprise inclut non seulement « une mémoire technique » obtenue par capitalisation du savoir-faire de ses employés mais également « une mémoire organisationnelle » (ou « mémoire managériale ») liée aux structures organisationnelles passées et présentes de l'entreprise (ressources humaines, management, etc.) et des « mémoires de projet » pour capitaliser les leçons et l'expérience de certains projets [Pomian, 1996].

1.3.2 Typologies de mémoire d'Entreprise

Comme pour les connaissances, il existe une multitude de typologies différentes de mémoire d'entreprise :

Ainsi [Tourtier, 1995] cité dans [Dieng et al., 1998] distingue quatre types de mémoire :

- la **mémoire de la profession**, composée des références, documents, outils, méthodes employés dans une profession donnée ;
- la **mémoire de la société**, liée à l'organisation, aux activités, produits, et acteurs comme les clients, sous-traitants, fournisseurs ;
- la **mémoire individuelle**, contenant le statut, les compétences, savoir-faire et activités d'un salarié donné ;
- la **mémoire de projet**, constituée de la définition d'un projet, les activités attenantes, l'historique et les résultats.

Pour [Pomian, 1996], la mémoire d'entreprise se décline en trois sous mémoires, la mémoire de projet, la mémoire organisationnelle et la mémoire technique :

- la **mémoire de projet** est directement liée à une mission particulière réalisée au sein de l'entreprise. Un groupe de personnes est formé pour un temps donné pour réaliser un projet bien défini. La mémoire de projet rassemble les connaissances, savoir-faire, compétences, documents qui ont été nécessaires à l'accomplissement de ce projet.
- la **mémoire organisationnelle** rassemble les connaissances pertinentes pour les activités de l'organisation à tous les niveaux. Elle peut ainsi inclure des informations sur les structures organisationnelles présentes et passées, sur les ressources humaines, etc. Par ailleurs, [Girod, 1996] y distingue les connaissances sur les liens actions résultats et les conditions dans lesquelles une action donnée conduit à un résultat donné, les connaissances sur la technologie, sur l'environnement interne et l'environnement externe.
- la **mémoire technique** se rattache quant à elle à un métier. Elle s'intéresse à l'aspect opérationnel de l'entreprise, c'est-à-dire à l'expérience acquise liée au travail et permettant à l'entreprise de vivre. Elle est constituée des connaissances liées à un métier et nécessaires à l'exécution des tâches des individus de l'entreprise en vue d'une activité particulière. Ces connaissances rassemblent les connaissances tacites inscrites dans le cerveau des hommes, leurs savoir-faire, leurs compétences, et les connaissances explicites, contenues dans les bases de données, manuels techniques, plans de fabrication, etc.

Le fonctionnement de ces sous mémoires repose sur un principe commun : un mécanisme d'acquisition, de rétention ou stockage et de restauration. Chaque sous-système utilise cependant des modes différents d'acquisition, de stockage ou de restauration selon ses spécificités et ses fonctions. C'est ainsi que l'on distingue généralement la **mémoire de court terme** qui stocke temporairement des informations et la **mémoire de long terme** qui stocke les informations de façon permanente et possède donc une capacité de stockage beaucoup plus importante que la mémoire de court terme.

Un exemple concret de la mémoire de projet est celui des travaux de recherche qui ont été réalisés par [Cantzler, 1997] sous l'égide de la Société Européenne de Propulsion (SEP) et du Centre National d'Etudes Spatiales (CNES) dans le cadre du programme de recherche et technologies afin de soutenir des compétences industrielles et l'efficacité des projets de développement dans le domaine spatial. Ce cas de projet sera présenté plus en détail dans le paragraphe qui suit. Les résultats de ce projet sont l'un des éléments clés à l'origine de ce travail de mémoire de recherche.

I.4. La gestion des connaissances en conception et en gestion de projets : cas du projet de la Société Européenne de Propulsion

I.4.1. Introduction

Dans le milieu industriel, l'activité de conception au sein des entreprises part rarement de « rien » : on doit faire évoluer une solution existante, ou tenter de reprendre une solution ancienne pour l'adapter à une demande nouvelle. En outre, des contraintes de temps, d'efficacité et de recherche d'une fiabilité maximale renforcent actuellement cette tendance à la réutilisation de solutions anciennes. La question qui se pose est de savoir si tous les moyens sont réunis dans les entreprises pour favoriser la réutilisation.

D'après [Tollenaere, 1996] la capitalisation des connaissances en conception contribue à une meilleure qualité des produits et une diminution des délais de conception.

La gestion de projet consiste en l'organisation méthodologique de mise en œuvre pour faire en sorte que l'ouvrage réalisé par le maître d'œuvre réponde aux attentes du maître d'ouvrage et qu'il soit livré dans les conditions de coût et de délai prévus initialement, indépendamment de sa fabrication.

La gestion de projet a pour objectif d'assurer la coordination des acteurs et des tâches dans un souci d'efficacité et de rentabilité.

Ainsi la phase de capitalisation consiste à faire un bilan du projet en archivant l'expérience métier (savoir-faire) qui découle du projet afin d'améliorer l'efficacité de la conduite pour les projets futurs. Il est donc indispensable à la fin d'un projet d'organiser la mise en mémoire de ces informations et de permettre leur restitution au plus grand nombre.

I.4.2. Cas du Projet de la Société Européenne de Propulsion (SEP)

La SEP et le Centre National d'Etude Spatiales ont souhaité introduire dans leurs bureaux d'études les méthodes permettant de capitaliser et d'enrichir les connaissances générées dans leurs projets de développement. Pour ce faire, un projet a été lancé dans ce cadre afin de mettre en place un outil qui permet de constituer une base de connaissances dans le domaine des avant-projets de turbopompes appelé ELACT³.

Les activités principales d'un avant-projet de TURBOPOMPES sont la vérification de la cohérence et de la complétude du cahier de charges, l'élaboration d'architecture viable, la sélection d'une ou plusieurs architectures de référence à étudier fonctionnellement et les études d'aménagement.

L'outil ELACT n'est pas dédié uniquement au domaine des turbopompes spatiales. Plus généralement, la mémoire technique (Base de connaissances) doit permettre la « traçabilité » des « historiques de conception » de systèmes technologiques complexes en phase de conception préliminaire. L'historique de conception est une structure combinée d'informations, qui associe la description de l'objet de conception, la description des activités qui le modifient, et la description de l'organisation réalisant ces activités. L'historique de conception est représentée à la fois par la trace des évolutions successives du produit et par la logique de cette progression.

Les connaissances qui sont capitalisées dans ELACT, sont des connaissances de « projet ». ELACT propose une structuration de ces connaissances et permet leur accès par des moyens de navigation. Ainsi, cette structuration permet-elle d'accéder aux principaux points clés d'un projet de conception, tels que les configurations étudiées, les choix technologiques, les justifications, les dates les ressources utilisées, les blocages, les goulets d'étranglement, et les défauts d'organisation. La masse d'information que représente la liste exhaustive de tous les apports et modifications à la turbopompe, est difficilement exploitable et trop coûteuse à capitaliser. Les utilisateurs d'ELACT ne capitalisent que le déroulé « macroscopique » du processus, appelé aussi le processus « global » de conception. Les fonctions générales d'ELACT se résument en quatre actions :

- Réutiliser l'expérience que constituent les choix technologiques et leurs justifications ainsi que les dysfonctionnements organisationnels du passé ;
- Suivre et planifier les évolutions du produit et les activités d'un projet en cours, en se basant sur l'expérience du passé

³ Environnement Logiciel d'Aide à la Conception de Turbopompes

- Evaluer les solutions proposées en fonction des contraintes organisationnelles et budgétaires ayant conditionnées un projet ;
- Comprendre les raisonnements qui ont débouché sur une solution et qui la justifient globalement depuis le besoin, et en tenant compte des apports de chacun des acteurs.

D'après [Cantzler, 1997] ELACT doit être une plate-forme répondant aux deux besoins de capitalisation et de réutilisation. L'architecture conceptuelle de la mémoire technique ELACT est composée de modèles de produit, d'activités, d'organisation et d'une couche dite cognitive.

I.5. Pourquoi une Mémoire de projet

Le souhait des concepteurs est de pouvoir réutiliser les projets passés, et notamment pouvoir retrouver leur logique de conception [Karsenty, 1996].

Pour favoriser la réutilisation, il faut aider les concepteurs à accéder au contexte des projets réalisés dans le passé. Il faut créer des mémoires de projet qui soient autre chose qu'un archivage des solutions réalisées dans chaque projet : ces solutions doivent être conservées, mais les raisons des choix ayant conduit à ces solutions aussi. Ces raisons doivent faire apparaître les connaissances expertes, les représentations du problème traité, les faits appris pendant le projet et toutes les informations contextuelles déterminantes dans les prises de décision des concepteurs.

Principalement l'objectif premier poursuivi par la mémoire de projet est l'amélioration continue de la façon dont on conduit les projets.

Quatre autres points doivent être soulignés pour dresser un tableau complet des retombées qu'une entreprise peut attendre de la mémoire de projet.

D'abord, elle ne se limiterait pas à améliorer la gestion de projets futurs. Si elle est réalisée tout au long d'un projet donné (plutôt que seulement à la fin), elle permettrait également de faire le point à différentes étapes de ce projet en train de se dérouler et de réajuster le tir, au besoin [Meredith et al., 1989].

Ensuite, le référentiel constitué pourrait servir à faire évoluer les « connaissances centralisées » de l'organisation, c'est-à-dire ses règles, ses procédures et ses normes formelles [Pomian, 1996]. Par ailleurs, la mémoire de projet ne comporterait pas que des avantages mais également des risques. Celui évoqué le plus souvent par les auteurs concerne la mise au jour d'« éléments noirs » que l'histoire officielle n'aurait pas retenus autrement [Pomian, 1996].

Enfin, si elle faisait partie d'une démarche de mémoire d'entreprise, elle contribuerait à un ensemble plus large de bénéfices. En particulier, elle serait un moyen d'améliorer la gestion d'ensemble de l'organisation (par exemple, en favorisant la mise sur pied de structures souples), notamment ses pratiques de communication et de gestion des ressources humaines : évaluation du rendement, mobilité interne, évaluation des besoins de formation, gestion prévisionnelle de la main-d'œuvre, valorisation des compétences, etc. [Pomian, 1996] [Davenport et al., 1998].

I.6. Etat de l'art lié à l'activité de conception :

I.6.1. Introduction :

Il s'agit d'une activité intellectuelle de création, fondamentalement liée à l'activité humaine.

De nos jours, les technologies de l'information permettent de réduire le processus de conception. Pour arriver à concevoir des produits dans les meilleures conditions de délais, de coûts et de qualité, il faut maîtriser tous les facteurs qui permettent à un concepteur de créer des solutions techniquement réalisables et répondant au mieux aux contraintes et objectifs exprimés dans le cahier des charges, donc à satisfaire les désirs d'un client.

Ainsi un problème de conception est un problème ouvert qui n'admet ni formulation, ni solution définitive [Rittel, 1972] ; [Bazjanac, 1974]. En effet, on ne peut jamais, ni exprimer intégralement le besoin que l'objet à concevoir a pour objectif de satisfaire, ni établir une liste exhaustive des moyens qui peuvent servir à construire un modèle de cet objet [Latombe, 1977]. Pour un tel problème, on doit se contenter d'une solution acceptable (au sens de critères définis en fonction du problème posé) obtenue avec des ressources limitées [Simon, 1969].

Ainsi, le processus de conception est un processus au cours duquel le problème de conception est formulé et résolu plusieurs fois [Bazjanac, 1974].

L'évolution et les besoins du marché actuel se traduisent par la recherche d'une qualité sans cesse croissante. Afin de pouvoir mieux s'adapter à cette évolution, les entreprises s'orientent de plus en plus vers la fabrication de produits dont le cycle de vie est plus réduit que par le passé. De ce fait, les entreprises se trouvent engagées plus fréquemment dans le processus de conception de produits nouveaux ou de re-conception de produits existants. De plus, l'émergence d'outils de CAO a fortement contribué à l'adoption d'une telle pratique, et ce à l'inverse de ce qui fut par le passé où l'activité de conception était une action plus rare et sur laquelle on revenait très peu. Cette démarche fréquente et même quasi quotidienne, dans le cas de gros secteurs industriels tels que l'aéronautique ou l'automobile, pose le problème de la capitalisation et de la réutilisation des expériences de conceptions passées. Cette capitalisation vise une éventuelle automatisation du processus de conception plus particulièrement dans la conception de produits similaires ou de re-conceptions de produits existants.

Ainsi par exemple et dans un contexte industriel, la conception a pour but de spécifier un produit répondant aux attentes supposées mais non toujours clairement et complètement identifiées d'un client. D'après [Stephanopoulos, 1989], cette activité s'opère suivant une dialectique entre des buts et des possibilités. D'après [Darses, 1997], cette activité est soumise à des conditions nombreuses et variées (lois du domaine, contraintes de fabrication, critères fonctionnels, etc.). [Bucciarelli, 1988] ajoute que cette activité revêt également un caractère social, dans la mesure où elle fait intervenir de nombreux acteurs ayant des langages, des règles, des objectifs et des contraintes différents.

I.6.2. Définitions

Dans la littérature il existe un grand nombre de définitions relatifs à l'activité de conception. Parmi elle et celle qui s'approche de nos besoins, à savoir la conception de produits est celle donnée par [Bonnardel, 1992] : Le terme « conception » désigne également le résultat des opérations précitées. Cette activité est « une activité essentiellement mentale,... orientée vers la réalisation d'objectifs,... focalisée sur les aspects pertinents du produit... ce qui laisse présager le

rôle important de l'expertise... et la nécessité pour le concepteur de se construire (progressivement) une représentation précise du but de la conception ».

D'autres définitions existent telles que celles données par [Jones et al. 1963], NF L 00-007⁴, l'AQAP⁵ et [Guéna, 1992].

Dans le cas où le concepteur ne dispose d'aucune procédure directement applicable pour atteindre le but de conception [Malhotra et al., 1980], cette situation amène le concepteur à adopter des stratégies visant à élaborer une « réponse adaptative » [Reuchlin, 1981].

Certaines situations de conception industrielle, en particulier quand le produit est déjà connu, peuvent mettre à la disposition du concepteur, des procédures prédéfinies, par exemple, dans le cadre d'une politique d'Assurance Qualité en Conception (AFNOR, normes ISO 9000). Cependant, l'activité de conception étant une activité complexe, les situations de conception sont généralement considérées comme des situations de résolution de problèmes [Calvez, 1990], [Bonnardel, 1992].

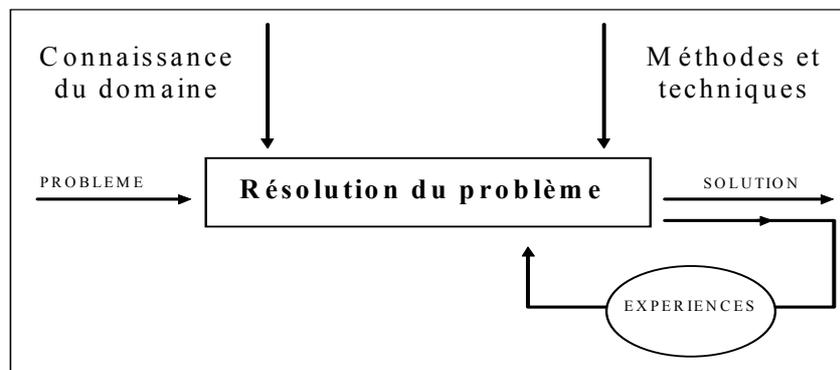


Figure 4 : Activité de conception : processus de résolution de problèmes [Bonnardel, 1992].

Un des problèmes majeurs dans le développement de systèmes d'aide à la conception est la représentation de la connaissance liée à la conception. Cette connaissance doit comprendre la représentation des objets à concevoir ainsi que la représentation des processus de conception de ces objets.

Selon [Chandrasekaran, 1990], la résolution d'un problème de conception dépend de la disponibilité de certaines connaissances en fonction desquelles on doit pouvoir choisir les méthodes de résolution. D'autre part [Chandrasekaran, 1990] définit la solution d'un problème de conception comme étant la spécification complète d'un ensemble de composants et de leurs relations, le tout décrivant un objet satisfaisant des contraintes et remplissant des fonctions.

⁴ NF L 00-007 : Norme française - industrie aéronautique et spatiale - vocabulaire- termes généraux - Paris, AFNOR, mars 1987.

⁵ AQAP : Allied Quality Assurance Publication: publication interalliée (OTAN) (cf THOMSON CSF "Vocabulaire de termes normalisés usuels de la Qualité".)

I.6.3. Caractéristiques générales de l'activité de conception :

D'après [Lanzara, 1986] concevoir c'est définir le problème. Une part importante de l'activité de conception est consacrée généralement à la structuration du problème, et le reste, à la recherche de la solution [Simon, 1973].

Cette participation du concepteur à la structuration du problème et à sa reformulation, provient du fait que les données de base du problème sont rarement complètes, suffisantes, et pertinentes.

Les réponses à un problème de conception sont souvent multiples. Les concepteurs travaillant en groupe ont alors à faire des choix de solutions alternatives qu'ils devront évaluer en fonction de critères validés par le demandeur du produit, et du poids de ces critères. Ce processus d'évaluation intervient tout au long du cycle de conception d'un produit [Bonnardel, 1992]. Parmi ces techniques d'évaluation pondérée des solutions par rapport aux besoins du client, on peut citer par exemple, le Quality Function Deployment (Q.F.D.). Le déploiement des fonctions Qualité, se compose d'un ensemble de programmes de planification et de communication. Cette méthode est née dans les entreprises japonaises au début des années 70. Le déploiement de la Fonction Qualité se compose de trois éléments indissociables :

- Une équipe pluridisciplinaire ayant en charge le projet du début à la fin.
- L'écoute du marché visant à témoigner des besoins des « clients-futurs utilisateurs », ainsi que des exigences du type réglementations, standards, législation, d'une part, et options stratégiques, contraintes d'entreprise, d'autre part.
- De supports graphiques, appelés « Maison de la Qualité » s'enchaînant les uns aux autres. Ces supports graphiques permettent de mettre en correspondance deux séries de données devant répondre les unes aux autres et d'évaluer la qualité de cette réponse.

Son utilisation facilite la communication de l'information, au sein d'une équipe chargée d'un projet et dans l'entreprise. Sa mise en œuvre exige cependant de gros efforts et une grande disponibilité des esprits, car la masse, la nouveauté et la précision des informations requises pour alimenter efficacement la « première Maison » est importante.

I.6.4. Les différents types de conception :

D'après [Durruvu et al., 1989], la conception est vue comme étant une activité de résolution de problème. Il distingue 4 catégories de conception :

- conception routinière : un plan de conception existe a priori. Pour tout nouveau problème, l'ensemble des sous problèmes et leurs solutions candidates sont connues à l'avance. La conception routinière consiste à déterminer, pour chaque sous-problème les alternatives adéquates.
- re-conception : une conception existante est modifiée pour satisfaire de nouvelles exigences fonctionnelles.
- conception innovante : la décomposition du problème est connue, mais il n'existe pas d'alternative connue pour tous les sous-problèmes. La conception innovante fait en général appel à la créativité, quoique le résultat de la conception puisse s'avérer être une nouvelle combinaison de solutions existantes.

- conception créative : elle consiste en une décomposition abstraite d'un problème en un ensemble de niveaux de sous-problèmes. Il n'existe pas de plan de solution a priori. L'élément clé de ce type de conception est la transformation de l'inconscient en conscient.

Il existe d'autres types de conception tel que ceux de [Sriram, 1989], [Iffenecker, 1992] cités par [Harani, 1997] ou l'on distingue trois différents types de conception : La conception prédéfinie ; la conception innovatrice ; la conception créative.

D'après [Brun, 1991], où sont rappelées les grandes approches méthodologiques de conception :

- conception à partir de rien (ce qui correspondrait à la conception créative, présentée auparavant),
- conception du type LEGO/MECCANO (ce qui correspondrait à assembler ou agencer des modules prédéfinis pour créer un produit nouveau, exemple : circuits électroniques),
- conception par variantes (ce qui correspondrait à modifier un existant),
- conception d'une famille de produits (ce qui correspondrait à la conception innovatrice) et
- conception d'une famille d'architecture de produits bien connue (ce qui correspondrait à la conception prédéfinie).

On retrouve aussi ces différentes typologies de conception dans [Anderl, 1995], typologies développées et utilisées suivant le contexte de la conception :

- conception nouvelle ou innovatrice (elle requiert un développement et une conception complètement nouveaux d'un produit, ce qui présume de passer par toutes les phases de conception), ou bien,
- conception adaptée (elle utilise une solution de conception existante, généralement le résultat de la conception préliminaire et de la définition de l'architecture, et adapte cette solution suivant les besoins spécifiques) et
- conception par variantes « *variant design* » (elle est basée sur une définition structurée des produits et des procédures prédéfinies pour modifier une conception existante).

I.6.5. Les Approches de conception :

D'après [Iffenecker, 1992] cités dans [Harani, 1997] il existe cinq stratégies ou approches de conception les plus fréquemment proposés :

- **La conception par décomposition de problèmes**

Un des principes de base en conception est le principe de décomposition largement utilisé dans l'analyse des systèmes complexes [Courtois, 1985]. Il consiste à décomposer un module en cours de conception en modules plus élémentaires. Il est admis que ce principe permet un fort degré de cohésion entre modules du système et un faible couplage entre modules. Ce principe a été étudié dans le cadre de la conception par [Brown et al., 1985] et [Chandrasekaran, 1989]. Ils présentent une approche de raffinement de la conception où celle ci est considérée comme un système hiérarchique. Le problème est résolu par une approche descendante de type *Top-Down* qui raffine le problème à chaque niveau de la hiérarchie. Cette approche a été reprise par [Kusiak et al., 1993] qui proposent une approche de décomposition d'une tâche globale de conception en sous-tâches ayant des interdépendances minimales entre sous-tâches de manière à augmenter le parallélisme dans le processus de conception (et donc de réduire le temps de conception).

- **La conception par prototype**

Cette modélisation suppose que l'on est capable de modéliser la conception du produit à partir d'un modèle général représenté par un prototype. La notion de prototype offre une structure d'accueil permettant de se rapprocher d'une conception passée déjà faite [Gero, 1987].

Si la conception du nouveau produit se fait par instanciation du prototype, c'est l'idéal. Ceci n'est efficace que dans-le cadre de conception à faible innovation.

- **La conception à base de cas**

A partir d'un nouveau problème de conception, on essaye de trouver les solutions passées pouvant être réutilisées. Ce modèle utilise directement l'expérience des concepteurs sous forme de bibliothèques de cas et non sous forme de généralisation (voir des modèles existants). Cette modélisation ascendante ou Bottom-Up de l'activité de conception est proche des techniques employées par les concepteurs. Toutefois, son automatisation dans un système est difficile car elle suppose la gestion d'un nombre important de cas. On arrive difficilement à savoir comment les cas passés doivent être transformés pour concorder avec les solutions désirées.

- **Les principes de spécialisation, de décomposition et de connexion**

Ces deux principes sont décrits par [Klein, 1993]. La spécialisation consiste à raffiner la définition d'un module en cours de conception en un module plus complet (par exemple : par le calcul d'un de ses paramètres ou en complétant sa géométrie). Le principe de décomposition est celui cité plus haut. Le principe de connexion consiste à relier deux modules entre eux et à préciser le type de connexion. Pour Klein, les principes de spécialisation, de décomposition et de connexion sont suffisants pour décrire l'activité cognitive du concepteur.

- **L'approche fonctionnelle par opposition à l'approche comportementale**

Il est essentiel de distinguer l'activité du concepteur (toutes les alternatives qui se sont présentées à lui, les choix qu'il a effectués lors de la conception) de la description du fonctionnement de l'objet conçu (artefact). Pour ce faire, [Chandrasekaran et al., 1993] propose un langage de représentation du fonctionnement de l'objet (langage FR) basé sur une approche fonctionnelle par opposition à l'approche comportementale. Dans l'approche fonctionnelle, on adopte une approche descendante pour représenter le mécanisme d'un objet dans le sens où la fonction globale est décrite en premier et le comportement de chaque composant est décrit dans le contexte de la fonction. Le langage FR fournit ainsi une structure nommée (*STRUCTURE*) pour décrire le mécanisme (*DEVICE*) et ses composants (*COMPONENTS*). Chaque composant est sujet à des fonctions (*FUNCTIONS*) et peut être relié à d'autres composants par des relations (*RELATIONS*). On peut ensuite caractériser les états du mécanisme par un ensemble de variables d'états constituées par des valeurs des variables du mécanisme. La description d'un processus causal faisant intervenir les fonctions pour passer d'un état à un autre permet de spécifier le comportement du système (ou mécanisme).

[Deneux, 2002] a distingué essentiellement deux approches de la conception :

- approche axiomatique : les principales approches axiomatiques ont été développées par [Yoshikawa, 1989], [Suh, 1990], [Sohlenius, 1992].

- approche algorithmique : les principales approches algorithmiques ont été développées par [Pahl et al., 1984], [Dinsdale, 1991], [Griethuysen, 1992].

D'après [Deneux, 2002], l'approche algorithmique est plus adaptée à guider le concepteur dans son travail car elle détermine plus précisément les opérations nécessaires au passage du domaine fonctionnel vers le domaine physique. De plus, les opérations préconisées dans l'approche algorithmique sont spécifiques à chaque tâche de conception contrairement à l'approche axiomatique où les opérations sont génériques et donc identiques pour chaque « mapping ». Néanmoins, un des intérêts de l'approche axiomatique est l'axiome d'indépendance car il permet de décomposer un problème complexe en sous-problèmes indépendants de complexité acceptable.

Par rapport au processus de conception, les deux approches sont assez similaires dans le sens où elles sont toutes deux séquentielles monotones.

I.6.6. Synthèse sur la conception

L'activité de conception est un ensemble de tâches assez complexe. L'artefact (objet de la conception) et le processus d'élaboration de l'artefact (ou processus de conception) sont deux concepts différents. Ils sont fortement interdépendants. En effet, d'une part, c'est à travers les différentes étapes du processus de conception que la spécification du produit est complétée. De même à partir des tâches composant le processus de conception que les différents paramètres caractérisant le produit sont calculés et déterminés. L'approche du type « conception par décomposition de problèmes » est une approche qui couvre nos besoins dans le présent travail de recherche. Ainsi et en se basant sur cette approche de conception, notre mémoire de projet sera décomposée en un ensemble de modules plus élémentaires à savoir la conception du produit, du processus, de l'organisation, des justifications etc.

Dans l'approche de décomposition du type de conception présentée ci-dessus par [Durravu et al., 1989], notre mémoire de projet est du type « conception innovante ». La décomposition de la mémoire de projet est connue à l'avance (produit, processus, organisation etc.). Le résultat de la conception est la combinaison des différents modèles construits.

En résumé, l'activité de conception est un système complexe composé d'éléments en interaction et en évolution [Delattre, 1985], [Le Moigne, 1994]. Supporter l'activité de conception revient à spécifier les moyens de conduire certains processus opérants de ce système ou mieux, d'assister les décisions qui interviennent dans ses processus de pilotage. Pour pouvoir supporter l'activité de conception, il est donc indispensable d'étudier et de formaliser ses processus.

L'objet du prochain paragraphe sera d'étudier les approches de modélisation des processus de conception.

I.6.7. Les Processus de conception

I.6.7.1. Définitions

De nombreuses définitions existent dans la littérature afin de caractériser le processus de conception, telles que :

[Castellani, 1993] a défini le processus de conception comme étant une ou plusieurs démarches qui permettent d'analyser et de concevoir, de retro concevoir ou de refondre les systèmes

d'objets. Une démarche est un ensemble de tâches ordonnées avec un déroulement séquentiel ou parallèle.

[Finkelstein et al., 1995] a défini le processus de conception comme étant un processus de création qui débute par le recensement d'un besoin et s'achève par la définition d'un système (physique ou abstrait) qui répond au besoin, ainsi que la méthode de sa réalisation.

[Nadler, 1995] : le processus de conception est un re-bouclage continu sur des séquences analytiques, synthétiques et d'évaluation jusqu'à la détermination d'une solution adéquate.

[Simon, 1995] : Le processus de conception est une activité de synthèse dans le sens de création des objets, des processus (le processus est récursif) et des idées en vue d'atteindre des objectifs. Elle débute par des primitives et des composants et consiste en une suite de combinaisons entre ces primitives et composants.

[Sobiesh, 1995] : Le processus de conception est un large enchaînement de changements d'états et de création d'une structure dynamique.

[Gauss et al., 1995] : Il s'agit d'un processus de résolution des problèmes et/ou de création des opportunités. Il est caractérisé par l'examen de la perception des objets tels qu'ils sont et des objets tels qu'ils doivent être.

D'après [Deneux, 2002], deux tendances sont à dégager de ces différentes définitions à savoir :

- La première regroupe celles qui considèrent que la nature du processus de conception est plutôt séquentiel [Pahl et al., 1984], [Nadler 1995], [Gauss et al., 1995], [Finkelstein et al., 1995] et organisé en une séquence de tâches ordonnées chronologiquement. Ainsi et dans le cadre du présent travail de thèse on se place dans cette catégorie de groupe de définition.
- La seconde concerne ceux qui considèrent que le processus de conception est mixte (séquentiel-parallèle) [Suh, 1990], [Castellani, 1993]. Le parallélisme exprime la simultanéité des tâches.

De plus, il est nécessaire de distinguer le processus de conception du processus d'innovation. Ce dernier correspond au passage de l'invention à l'innovation. Il mobilise de nombreuses activités tels que le marketing, la recherche et développement, la production etc. Pour [Cooper, 1979] « le processus d'innovation débute par une idée, développée par des activités techniques et de marketing ». Pour [Xuereb, 1991], le processus d'innovation comprend « l'ensemble des activités mises en œuvre pour transformer une idée de produit nouveau en une réalisation effective ».

I.6.7.2. Modélisation du processus de conception

D'après [Deneux, 2002], il existe deux catégories d'approches de modélisation des processus de conception :

- l'approche fonctionnelle, qui se base sur l'analyse structurée et exploite des modèles de type IDEF0 [Mayer et al. 1995]. Elle fait l'objet de réflexions sur l'interaction, les dépendances et les recouvrements entre les tâches.

- l'approche « modèle de comportement » qui exploite les réseaux de Petri synchronisés. L'objectif est d'évaluer la durée du cycle de développement en présence d'infaisabilités. Il existe plusieurs modèles spécifiques d'activités de conception de produits tels que décrits dans [Vargas, 1996], [Harani, 1997], [Ouazzani et al., 1997]. Ces derniers s'inspirent et se basent sur les caractéristiques des outils classiques tels qu'IDEF0 et IDEF3 [Mayer et al. 1995], le modèle de comportement continu et discontinu de Petri et le modèle GRAI [Doumeingts, 1984]. Dans [Perrin, 1999] cité par [Deneux, 2002], une étude de la diversité des représentations du processus de conception, met en évidence la pluralité des modes de pilotage de ces processus. On y rencontre différentes approches de modélisation du processus de conception, vu respectivement comme une succession hiérarchique de phases, une itération d'un « cycle élémentaire de conception », une production mobilisant des ressources, ou encore la mobilisation et la création de ressources spécifiques. Selon [Eynard, 1999], la conduite des processus de conception de produits consiste à en coordonner les différentes activités. Elle se définit comme la synchronisation de la gestion de l'évolution de la connaissance du produit et de la gestion de l'affectation des ressources. D'autres travaux sur la modélisation des processus de conception [Wilson, 1980], [Bahrami et al., 1993], [Breuker et al., 1993], [Vargas, 1996], [Leinen, 1997],[Ouazzani et al., 1997], [Sénéchal et al., 1998], [Serré, 2000], etc. existent.

I.6.7.3. Modélisation du processus d'innovation

Il existe plusieurs structures ou modèles du processus d'innovation. Le modèle le plus connu en matière d'innovation est le modèle linéaire et séquentiel, qui se présente sous la forme d'une succession de phases de recherche, de développement et de déploiement. Ce modèle est à l'innovation ce que le modèle algorithmique est à la conception.

Un modèle séquentiel, s'inspirant des techniques de gestion de projet, a été formalisé dans le cadre du projet AIT-IMPLANT, [Eschenbacher et al., 1999]. Ce modèle, qui supporte la méthode générique de « gestion du changement » comprend cinq activités principales, séparées par des décisions :

- « Context for change : contexte de changement » : modéliser l'état de l'entreprise avant l'innovation,
- « Assessment of change : estimation du changement » : formaliser une vision du futur prenant en compte l'innovation,
- « Preparation for change : préparation pour le changement » : prévoir le processus opérationnel permettant l'intégration de l'innovation,
- « Achieving the change : finalisation du changement » : intégration progressive de l'innovation dans l'entreprise et
- « Auditing the change : auditer le changement » : mesure périodique de l'écart entre ce qui était prévu et le résultat constaté.

I.7. Conclusion

Dans ce premier chapitre nous avons présenté un état de l'art de quelques concepts liés au Knowledge Management, les difficultés et l'intérêt d'avoir une politique de KM dans une entreprise. On a vu que aussi que le KM puise ses racines du savoir et du savoir-faire de l'être

humain. Cependant, le passage de la théorie à la pratique en matière de KM est certainement un exercice qui requiert un nombre important de compétences afin d'être réellement traité. Dans un but d'avoir une bonne politique de KM dans une entreprise il serait préférable de :

- Miser sur l'être humain,
- Adopter une structure organisationnelle favorisant l'échange de connaissances,
- Utiliser une infrastructure technique intégrant les fonctionnalités d'intégration et de collaboration et
- Instaurer une culture de partage et de confiance

D'autre part et dans une entreprise, pour réussir la capitalisation des connaissances il faut tenir compte de tous les aspects organisationnels, humains, technique etc. et de ne pas réduire le problème de la capitalisation à un problème humain par exemple (ou technique). Pour ce faire la mémoire d'entreprise à mettre en place requiert une approche multidisciplinaire afin de mieux capitaliser la connaissance.

C'est cette connaissance capitalisée que les approches d'ingénierie des connaissances formalisent. Ces techniques d'ingénierie des connaissances seront d'une utilité pour formaliser l'activité coopérative qui dans notre cas n'est autre que l'activité de conception. Dans ce type d'activité plusieurs dimensions sont à prendre en compte comme la négociation la gestion de conflits, la coordination des tâches et la prise de décision coopérative. Le contexte de production de cette connaissance doit être également pris en compte, puisqu'il permet de fournir un cadre d'interprétation pour la restitution de la connaissance.

L'origine du présent travail de mémoire est le projet mené par [Cantzler, 1997] au sein de la SEP et le CNES. Ce dernier, dans son travail, a présenté l'architecture conceptuelle d'une mémoire technique [Pomian, 1996] pour la capitalisation de connaissances en conception préliminaire de turbopompes de moteurs de fusées. La spécification conceptuelle d'un tel système d'information intègre tous les aspects fonctionnels, informationnels, méthodologiques et graphiques, ainsi que leurs interactions dans une perspective globale.

L'univers de discours de [Cantzler, 1997] été limité au produit, processus, organisation et aux éléments cognitifs. Dans le présent travail on se propose de développer et de mettre en œuvre une mémoire de projet constituée de modèles de produit, de processus, d'organisation et d'étendre le domaine de connaissances à d'autres concepts tels que la justification des choix de décision lors des phases de conception, le versionnement des produits, la traçabilité etc. La grande caractéristique des modèles est la généralité. Cette mémoire de projet n'est autre que la mémoire de cumul de mémoire technique ou de métier.

Dans un but d'atteindre notre objectif, à savoir, la modélisation et la mise en œuvre d'une mémoire de projet, un état de l'art sur les méthodes de formalisation des connaissances sera présenté dans le prochain chapitre afin de retenir la méthode la plus adéquate et qui répond au mieux à nos besoins de modélisation. Une réponse à nos besoins en mémoire de projet, à savoir, les Systèmes de Gestion des Données Techniques (SGDT) solution qu'on démontrera insuffisante de points de vue objectifs de la mémoire de projet.

CHAPITRE II : ETAT DE L'ART LIE A LA MEMOIRE DE PROJET

II.1. Introduction

De nos jours les concepteurs dans beaucoup d'entreprises se réfèrent à des mémoires de projet afin de faire évoluer une solution existante ou d'essayer de reprendre une solution ancienne pour l'ajuster à un nouveau besoin. Le facteur réutilisation dans les mémoires de projets est très important, il a un effet direct sur les contraintes temps, coûts et efficacité. Pendant les phases de projets des connaissances émergent. Il existe un certain nombre de méthodes qui permettent de capitaliser ces connaissances. Ces méthodes aident à garder trace des expériences du passé.

Dans le présent chapitre nous allons présenter un état de l'art du KM et les systèmes de gestion des connaissances. Nous présentons un état de l'art de différentes méthodes de capitalisation des connaissances avec une synthèse comparative de différentes méthodes de formalisation des connaissances. Nous présentons un état de l'art sur la mémoire de projet. Nous présentons une étude comparative des fonctionnalités d'une mémoire de projet et d'un SGDT.

II.2. Etat de l'art du Knowledge Management (KM)

II.2.1. Le KM: Une branche pluridisciplinaire

Aujourd'hui, alors que le KM est largement connu et pratiqué dans de nombreuses organisations, peut être serait-il utile de regarder un peu en arrière et de le mettre en perspective par rapport aux différents courants qui ont contribué à son émergence. Nous reprenons ci-après quelques éléments significatifs de l'étude de Larry Prusak⁶ sur les origines du KM [Prusak, 1999] d'une part et celle d'Alexandre Pachulski [Pachulski et al., 2000] d'autre part qui ont révélé quelques éléments significatifs sur la naissance du concept KM. Ces courants sont :

- **Courant sociologique** : l'intérêt très fort de la recherche sociologique pour les structures complexes des réseaux et des communautés internes a un rapport très évident avec le KM. En effet, ces études considèrent que les réseaux communautaires constituent les unités les plus productives pour les tâches liées à la connaissance. Emilie Durkheim a souligné l'importance des «faits sociaux» : les comportements réels et observables sur lesquels devrait s'appuyer la pensée sociologique [Durkheim, 1982]. Ainsi le KM, plutôt que de partir de la théorie, regarde ce que font réellement les gens, les circonstances dans lesquelles ils partagent ou non leurs savoirs, leur manière d'utiliser, de modifier ou d'ignorer ce qu'ils apprennent d'autrui. Ces faits sociaux devraient guider le développement des approches et des techniques de KM.
- **Courant philosophique** : depuis ses débuts le KM, a exploré les différences existant entre les connaissances tacites et explicites, entre le savoir comment et le savoir pourquoi. Cette étude épistémologique des connaissances a été à la base du KM [Ryle, 1984] [Polanyi, 1996].

⁶ Larry Prusak est le maître à penser Américain d'IBM Global Services pour la gestion de connaissances. Cette étude a été publiée sous le titre « where did knowledge management come from? » 1999, est une version mise à jour d'un article publié initialement dans Knowledge Direction, publication de l'Institute for Knowledge management, Fall 1999.

- **Courant psychologique** : la psychologie cognitive est elle aussi concernée par les différents types de savoirs, tout autant que du pourquoi et du comment les gens apprennent, ignorent, agissent. Elle étudie les processus cognitifs naturels et soulève des questions de volonté et de motivation qui rendent impossible de considérer la connaissance en termes de transmission mécanique entre un donneur et un receveur [Prusak, 1999].
- **Courant économique et managérial** : ce courant a fortement participé à l'émergence du concept de management de connaissances. En effet, nous avons assisté à un changement de paradigme de la stratégie d'entreprise passant de « l'approche basée sur les ressources », à une approche basée sur le savoir. En conférant à la connaissance une valeur économique, au même titre que toute autre ressource matérielle faisant partie du capital, ces recherches ont ouvert la voie à une nouvelle théorie économique qui place le savoir au centre du processus de création de la richesse. Sur le plan managérial, nous assistons à des changements organisationnels prenant en charge la problématique de management des connaissances de l'entreprise [Drucker, 1993] [Polanyi, 1996]. Concrètement, l'entreprise doit apprendre à établir des relations de confiance entre ses membres, c'est-à-dire mettre en relation des personnes dont la coopération peut être génératrice de connaissances nouvelles et utiles tant pour elles-mêmes que pour l'entreprise.
- **Courant intelligence artificielle et ingénierie des connaissances** : l'intelligence artificielle a introduit la notion de connaissance dans l'univers informatique où il n'était question que des données et de leur traitement, les connaissances déterminant à la fois « le comportement, la configuration et la portée des programmes d'intelligence artificielle » [Ganascia, 1990]. Ainsi en introduisant la connaissance comme matière première de l'informatique, l'intelligence artificielle a produit une véritable révolution: « La généralisation des techniques de résolution de problèmes induit un nouveau mode de programmation pour lequel les connaissances du domaine sont assimilables à un programme. Le pas est franchi, on est passé d'une programmation procédurale classique à la construction d'une base de connaissances, c'est-à-dire d'une succession d'instructions, exécutables selon un ordre rigoureusement établi, à une simple description structurelle des objets de l'univers et de leurs propriétés » [Ganascia, 1990]. De là sont nés les domaines de l'apprentissage, de la résolution de problèmes, des modes de raisonnement et plus tard de l'ingénierie des connaissances.
- **Courant Ingénierie des systèmes d'information** : la gestion de l'information et le KM sont tous deux importants et nécessaires pour les entreprises d'aujourd'hui. Ces deux disciplines ont pour objectif commun l'amélioration de processus et de leurs produits ; toutefois elles présentent des différences fondamentales.

Alors que la gestion de l'information va se concentrer sur les technologies et l'organisation, le KM va s'attacher à considérer un autre facteur : le facteur humain puisque seul l'être humain possède la connaissance, peut la créer et décider de la partager ou non. De plus, ces deux disciplines se préoccupent de deux ressources différentes : information versus connaissance. En définitif, un système d'information a pour rôle de supporter le KM en fournissant une infrastructure technique facilitant la circulation des connaissances. Un système de KM est un système plus général qu'un système d'information intégrant l'aspect humain [Balmisse, 2002].

II.2.2. Les dimensions du KM

Un projet de KM est un projet transversal qui touche toutes les composantes de la stratégie de l'entreprise, ainsi toute stratégie efficace de KM doit adopter une vue aussi complète de ce champ. Pour illustrer les dimensions à prendre en considération dans une démarche de KM, nous nous sommes basés sur le modèle de Karl Wiig [Studer, 1999].

En adhérant à cette vision, nous considérons que le KM est basé sur quatre piliers complémentaires fortement imbriqués par leurs liens et leurs interactions : l'organisation, le capital humain, les technologies et la culture. Ces piliers sont fondés sur une culture organisationnelle appropriée : une culture de partage. Pour réussir la mise en œuvre d'une démarche de KM, une approche intégrée est exigée en vue d'examiner ces quatre éléments entièrement et mettre en valeur les barrières potentielles sous-jacentes à ces dimensions.

- **Dimension organisationnelle** : dont l'objet est l'organisation : ses valeurs, sa structure (hiérarchique, plate, etc.), son mode de pilotage et de fonctionnement. Comme nous l'avons expliqué précédemment, la connaissance est caractérisée par une dimension collective. Le processus central pour créer de nouvelles connaissances se déroule intensément au sein de groupes ou communautés d'intérêts. Ce type d'interaction intense, a difficilement lieu dans une structure hiérarchique. Ainsi, il est primordial d'adapter la structure de l'organisation pour être en mesure de supporter la formation de communautés d'intérêts auto-organisées.
- **Dimension humaine** : dont l'objet est le comportement des groupes et des individus acteurs du KM : leurs besoins, leurs pouvoirs, leurs zones d'autonomie, leurs responsabilités, leurs modes de rémunération, leurs structures, leur éthique, et leurs valeurs. Il s'agit là de dépasser la conception selon laquelle la connaissance constitue un bien personnel, encore plus une façon d'exercer le pouvoir et s'orienter plutôt vers une culture de partage où le tout est supérieur à la somme des parties. Cette dimension est la phase la plus délicate dans une démarche de KM.
- **Dimension technique** : les technologies vont aider à la capitalisation et à la diffusion des connaissances dans l'organisation. Elles supportent la capture, le stockage, la structuration et la diffusion des connaissances. Elles permettent notamment de s'affranchir des problèmes de temps et d'espace en fournissant un support facilitant les communications entre individus. Les technologies sont donc indispensables au déploiement des solutions de KM. « Le Knowledge Management met les hommes au centre de son projet de création de valeur, l'organisation et les outils informatiques formant l'épine dorsale du système » [Balmisse, 2002]. La majorité des solutions technologiques de KM traite essentiellement, grâce à des moteurs d'indexation et à des outils de classification, des connaissances explicites. Néanmoins, l'émergence de solutions telles que collecticiels, localisation d'expertise ou encore messagerie instantanée ont permis au KM de prendre en compte partiellement les aspects tacites de la connaissance.
- **Dimension culturelle** : le terme « culture » a de multiples sens. Pour comprendre la portée de ce terme dans le cadre du KM, Joseph Firestone [Firestone, 2001] identifie deux dimensions de la culture : subjective et objective.

- « la culture subjective d'un groupe est constituée de l'ensemble de ses prédispositions émergentes pour percevoir son environnement. Elle inclut les valeurs et les attitudes à l'échelle organisationnelle, ainsi que leurs relations ».
- « la culture objective d'un groupe est relative à l'ensemble des problèmes émergents, modèles, théories, créations artistiques, langages, programmes, histoires etc. qui ont été reflétés dans les documents, livres, systèmes d'information, dictionnaires et autres conteneurs ».

En tenant compte de ces deux dimensions, la plupart des auteurs affirment qu'une démarche de KM doit être accompagnée de l'instauration d'une culture de partage. Le passage à une culture de partage soulève des questions essentielles comme : Pourquoi partager ? Quoi partager ? Comment partager ? etc.

Ainsi avant de partager, il est essentiel de convaincre les collaborateurs d'une organisation que le partage apporte plus que la rétention. En effet, contrairement aux actifs corporels, les connaissances progressent fortement lorsqu'elles sont partagées.

[Prax, 2002] a instauré une culture de partage est donc un élément de succès fondamental dans une démarche de KM.

Ayant analysé, ces dimensions nous tournons notre attention vers les différents processus caractérisant une approche de KM. Ces processus seront discutés dans la section suivante.

II.2.3. Le KM : un ensemble de processus

[Grundstein et al., 1999] affirme que le KM est un méta processus cognitif de l'organisation qui part de la construction et la consolidation de ses « fondations » pour un développement exponentiel. Pour ce faire, une démarche de KM met en œuvre un certain nombre de processus depuis la création de connaissances jusqu'à son utilisation au sein de l'organisation. Cet enchaînement est qualifié par ce qu'on appelle le cycle de vie de connaissances. Dans ce qui suit nous présentons les processus clés qui contribuent à une démarche de KM.

- **Processus de création de connaissance** : la création de connaissances est étroitement liée au degré d'interaction et de collaboration dans une communauté de pratiques [Wenger, 1999]. C'est la capacité d'une organisation dans son ensemble de créer de nouvelles connaissances, de les disséminer au sein de l'organisation, de les faire prendre corps dans des produits et des services. La création de connaissances est principalement un processus social auto organisé, complexe, donnant naissance à des propriétés (connaissances) émergentes. Ces connaissances se créent par des interactions initialement aléatoires d'agents en relation [McElroy, 2002]. Ceci demande de la part de l'organisation un changement important de culture pour stimuler des interactions créatrices et innovantes de ses collaborateurs. Ce processus peut inclure l'exploitation de techniques de veille stratégique et d'outils de Data Mining pour fournir des informations cruciales, enrichissantes aux collaborateurs de l'organisation.
- **Processus de capture des connaissances** : il concerne les problèmes liés au repérage de connaissances cruciales nécessaires aux processus de décision et au bon déroulement des processus essentiels qui constituent le cœur des activités de l'organisation: il faut les identifier, les localiser, les caractériser, en faire des cartographies, les hiérarchiser etc.

- **Processus de capitalisation des connaissances** : le but est de « *localiser et rendre visibles les connaissances de l'entreprise, être capable de les conserver, y accéder et les actualiser, savoir comment les diffuser et mieux les utiliser, les mettre en synergie et les valoriser* » [Wenger, 1999]. Il s'agit principalement de modéliser, formaliser et conserver les connaissances pour permettre leurs traitements par des systèmes d'information. Plusieurs méthodologies ont été développées pour capitaliser les connaissances (REX, CommonKADS etc.)

- **Processus de partage et d'utilisation de connaissances** : c'est le principal objectif du KM. Sa réussite conditionne l'utilité du système établi. Le savoir partagé, issu de la synergie des esprits est une source d'innovation durable pour l'organisation. Cette phase a lieu quand les collaborateurs d'une entreprise qui se sont approprié la connaissance, l'ont intégrée, commentée ou échangée. Le partage des connaissances est plus efficace quand il se fait par le biais d'échanges « face to face » car la communication entre individus est constituée par un ensemble complexe de signaux (gestuels, postures) ; il peut aussi être supporté par des technologies de communication tels que les collecticiels qui visent à faciliter la coopération et la communication au sein et entre divers groupes.

II.2.4. Enjeux et Obstacles d'une démarche KM

II.2.4.1. Les enjeux

Jean Yves Prax [Prax, 2002] a identifié quatre grands enjeux du KM segments :

- optimiser les processus, la productivité, l'efficacité collective, réduire les délais, améliorer la qualité et réduire les coûts, en réutilisant mieux la connaissance déjà existante : capitalisation des bonnes pratiques, réduction des erreurs répétitives, etc.
- améliorer les décisions par l'échange multidisciplinaire, être à l'écoute des collaborateurs, anticiper leurs besoins.
- mieux valoriser le capital de compétences, comprendre l'entreprise non pas comme un système de production mais comme un réseau de compétences, dont chaque acteur fait partie intégrante.
- Innover, être capable de créer des idées nouvelles, les intégrer dès leur émergence, puis les valider, et les transformer en projet, en processus et en artefacts. Le Knowledge Management est un levier d'innovation et de productivité tout en étant un garant de la mémoire de l'organisation.

II.2.4.2. Les obstacles

Nous avons vu que le KM repose sur trois facteurs essentiels : Humains, organisationnels et technologiques. Chacun de ces facteurs soulève des problèmes qui peuvent ralentir la mise en œuvre efficace du KM.

- **Freins liés à l'organisation** : nous avons expliqué que la mise en place d'une démarche de KM doit s'accompagner d'une organisation pour faciliter l'échange des connaissances entre leurs détenteurs. Malheureusement, la structure hiérarchique est tout de même très présente et il est difficile de basculer rapidement vers un autre type d'organisation favorisant le décloisonnement des unités organisationnelles.

- **Freins liés aux individus** : la mise en place de dispositifs de KM se heurte fréquemment à un manque de motivation de la part des utilisateurs potentiels, soit parce que ces derniers ne voient pas ce que le dispositif peut leur apporter, soit car ils estiment que les outils sont mal adaptés à leurs besoins ou à leurs compétences. Un autre obstacle majeur lié à cette dimension tient à la difficulté d'adopter les principes de partage et d'échange dans les organisations au sein desquelles les logiques de cloisonnement sont encore très pesantes.
- **Freins liés à la technologie** : Certains éditeurs de logiciels contribuent à rendre le marché de solutions technologiques de KM très confus, en proposant des solutions intégrées inadaptées à la problématique de l'organisation. Un autre point important à souligner, est que les outils habituels de KM (portails, moteurs de recherche, collecticiels etc.) ne sont que des leviers qui accélèrent la médiation des connaissances. Mais ils ne participent pas aux processus de transformation et d'intégration des connaissances.

II.3. Les Systèmes de Gestion de Connaissances (SKM)

Le KM ne se résume guère à une problématique technique. Pour autant, sa réalité opérationnelle dans les organisations reste indissociable des TIC et de leurs évolutions. La liste des technologies pouvant être impliquées dans une approche KM peut être longue, nous nous limitons aux :

Intranets : des plateformes de gestion intégrée, utilisées à des fins de communication interne (messagerie électronique, téléconférence) d'une organisation. Elles sont considérées comme des moteurs simples, peu coûteux permettant de sauvegarder, échanger, et rechercher de l'information.

Systèmes de gestion de contenu : Un contenu est le résultat d'une rédaction d'une ou de plusieurs personnes dans le but de transmettre des informations à un public ciblé. Un système de gestion de contenu aide les organisations à gérer ce contenu suivant un « cycle de publication », puisqu'il facilite la création, le stockage et le partage du contenu. Ces systèmes rejoignent le même principe que celui de la gestion électronique de documents (GED).

Systèmes de gestion de workflow : il s'agit de systèmes qui automatisent, supportent et animent l'exécution des processus organisationnels formalisés, structurés et définis avec tous les éléments clés d'un processus : les actions, leur ordonnancement, les intervenants et leurs rôles.

Technologies d'intelligence artificielle : Techniques d'analyse de contenu, des systèmes de recherche d'information et de classification incluant les techniques de traitement de langage naturel, les techniques de datamining et de textmining, les techniques de traduction automatique jusqu'aux techniques de gestion profiling,

Groupware : Ensemble d'applications informatiques, plus ou moins intégrées, qui visent à faciliter le travail en groupe et mettent l'accent sur l'aspect de communication (messagerie électronique, liste de diffusion, forums de discussions, chat), l'aspect coopération (agendas électroniques partagés, outils d'édition partagés) et l'aspect coordination.

E-learning : Un ensemble de méthodes et d'outils permettant un apprentissage à distance grâce aux technologies Internet, le e-learning est fondé sur une approche collaborative de l'enseignement, les collaborateurs peuvent suivre différentes formations à n'importe quel moment, tout en étant suivi par un formateur.

II.3.1. Définition d'un SKM

D'une manière générale, un SKM doit assister tout individu impliqué dans les processus de compréhension, d'évaluation et de (re)-organisation des différentes activités d'une organisation. Plus précisément, un SKM doit offrir les moyens aux individus de créer, organiser, partager et échanger leurs idées, compétences et expertises ; il doit permettre aux organisations de localiser les individus et les communautés sources d'innovation, de les inciter à créer et échanger des connaissances et améliorer leurs capacités d'apprentissage. Enfin, un SKM permet d'identifier et d'appréhender les relations existantes entre individus, processus et connaissances.

La Figure 5 illustre une métaphore pour montrer les technologies et les concepts pouvant influencer la conception et l'implémentation des SKM. Dans cette figure, le concept de SKM joue un rôle central. Un tel système est le résultat d'une combinaison synergique de plusieurs technologies tout en se basant sur les approches théoriques du KM (partie droite de la figure) et en adoptant une démarche ou une stratégie (partie gauche de la Figure 5) indiquant les objectifs du système ce qui donne lieu à des systèmes, des outils, des plates-formes se focalisant sur le support des concepts et les théories du KM.

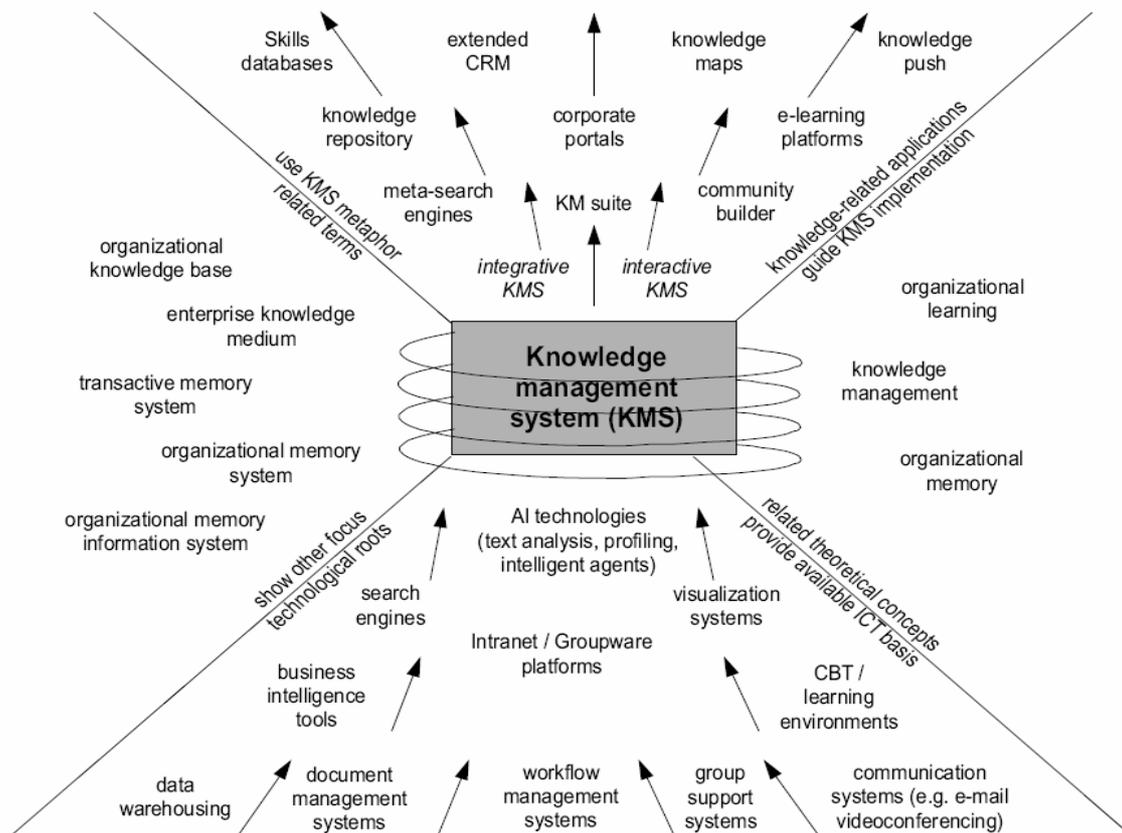


Figure 5 : Technologies et Approches influençant la construction d'un SKM [Maier, 2002].

II.3.2. Fonctionnalités d'un SKM

[Maier, 2002] distingue trois catégories de fonctions, à savoir : les fonctions d'intégration, les fonctions d'interaction et les fonctions de liaison. [Hansen et al., 1999] propose une classification rejoignant celle de [Maier, 2002]. La distinction de [Hansen et al., 1999] concerne les stratégies de KM en stratégie de personnalisation et stratégie de codification. Nous trouvons dans la littérature une autre classification, qui rejoint les deux premières et qui distingue les approches de KM orientées processus et des approches orientées produit.

Selon [Maier, 2002], les fonctions d'intégration d'un SKM sont celles qui supportent les stratégies de codification et adoptent une approche de SKM orientée produit ; l'objectif de ces fonctions est de se focaliser sur les technologies manipulant les connaissances explicites et plus précisément textuelles, ainsi celles qui concernent leurs création, stockage et (re)-utilisation dans des entrepôts documentaires (des mémoires organisationnelles).

Quant aux fonctions d'interaction d'un SKM, elles supportent les stratégies de personnalisation en adoptant une approche de KM orientée processus. Cette approche perçoit le KM comme un processus social auquel tous les individus d'une organisation participent, communiquent et coopèrent. Les fonctions d'interaction sont supposées faciliter ce processus social de communication. D'après [Maier, 2002], un certain nombre de fonctions ne peut être classé dans l'une ou l'autre classe de fonctions. Ainsi, il a introduit une troisième classe de fonctions qui sont les fonctions de liaison ayant pour objectif de réduire l'écart qui existe entre les fonctions d'interaction et les fonctions d'intégration.

- **Les fonctions d'intégration**

[Mayer, 2002] recense 28 fonctions d'intégration qui mettent l'accent sur la gestion des connaissances explicites (documents) d'une organisation. Ces fonctions peuvent être classées en trois sous catégories :

- recherche et présentation de connaissances,
- acquisition, publication et organisation de connaissances et
- administration du système.

- **Les fonctions d'interaction**

[Mayer, 2002] propose 20 fonctions d'interaction. Ces fonctions sophistiquées se focalisent sur les aspects de partage de connaissances, de coopération et de collaboration entre les communautés d'experts en leur fournissant des espaces virtuels de travail coopératif. Elles peuvent être classées en trois catégories :

- Communication et coopération de connaissances,
- e-learning et
- administration du système.

- **Les fonctions de liaison**

[Mayer, 2002] distingue 14 fonctions de liaison. Les fonctions d'intégration et d'interaction sont combinées pour aboutir à des SKM fortement intégrés, reliant ceux qui produisent la connaissance à ceux qui la consomment et enrichissent le contexte de recherche et présentation de connaissances en prenant en considération les unités organisationnelles. Ces fonctions peuvent être classées en trois catégories :

- recherche et présentation de documents,

- acquisition, publication et organisation de connaissance et
- administration

Dans la partie suivante, nous présentons des SKM développés en industrie et dans la recherche.

II.3.3. Exemples de SKM existants

Dans cette section, nous présentons les SKM, jugés à notre avis, les plus intéressants. Nous évoquerons avec un peu plus de détails les caractéristiques de quelques uns. Un peu plus haut, nous avons évoqué les approches du KM orientées processus et celles orientées produit pour différencier les fonctionnalités qu'un SKM est supposé assumer. En effet, un tel système doit au minimum supporter l'une de ces deux approches de KM. [Cobos et al., 2002] établi qu'un SKM doit vérifier l'une des deux caractéristiques techniques suivantes :

- Faciliter le travail coopératif entre les individus impliqués dans les processus de KM et
- Fournir une structure robuste pour les connaissances organisationnelles gérées.

Ces deux aspects techniques permettent de définir une classification des SKM existants. Ainsi, nous distinguons des systèmes qui se focalisent sur l'aspect de travail coopératif en vue de générer des connaissances organisationnelles collectives. D'autres systèmes s'intéressent aux représentations et aux structures de connaissances organisationnelles. Mais, il existe un certain nombre de systèmes que cités par [Cobos et al., 2002] et intitulés les SKM intégrés (Knowledge management integrated systems) qui fusionnent en un seul système les techniques de travail coopératif et les techniques de structuration et d'organisation de connaissances dans des mémoires organisationnelles communes (Figure 6).

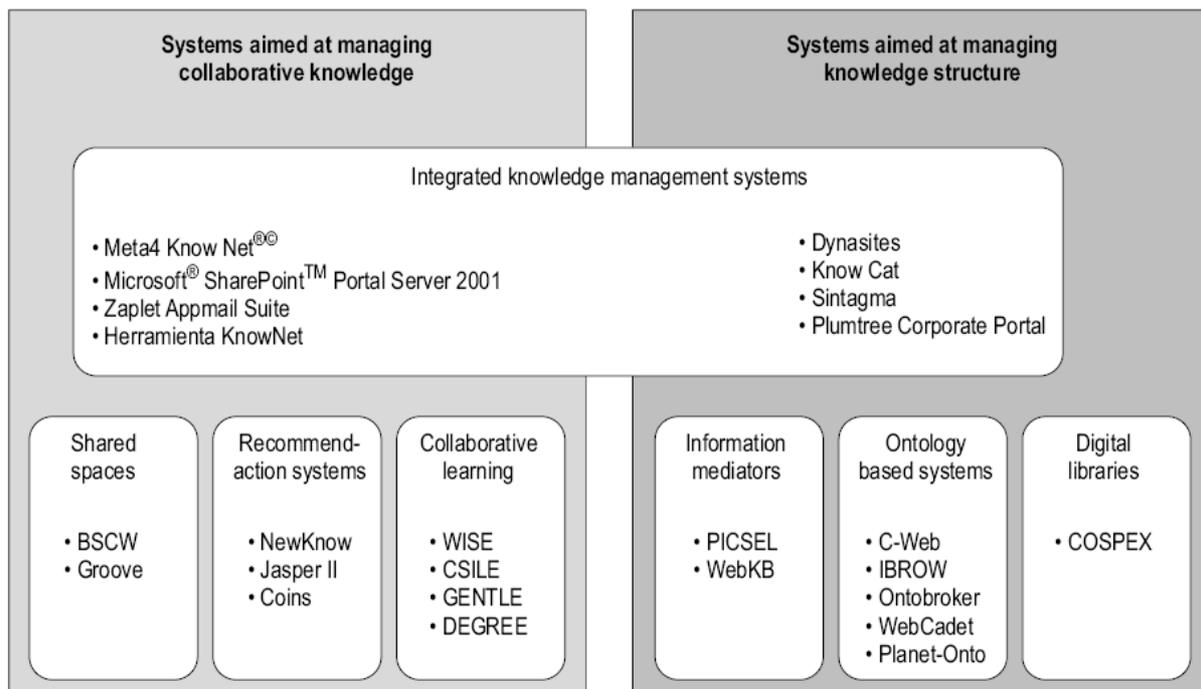


Figure 6 : Classification des Systèmes de Gestion des Connaissances [Cobos et al., 2002].

II.3.3.1. Les SKM intégrés

La caractéristique commune à tous les SKM intégrés est qu'ils capitalisent et organisent les connaissances collectives du type organisationnel, humaines, techniques, culturelles, dans des espaces communs aux utilisateurs. Ceci donne naissance à des mémoires organisationnelles. Dans ces systèmes, les unités de connaissances prennent, généralement, la forme de documents de différents formats allant des pages Web libres aux documents structurés.

Le premier aspect à analyser, dans de tels systèmes, serait la structure de la représentation formelle attribuée aux unités de connaissance manipulées. Dans la plupart des cas, la structure la plus utilisée est celle de la hiérarchisation de thèmes, intitulée aussi arbre de connaissances. Cependant, dans un certain nombre de systèmes, les connaissances peuvent être organisées en fonction de la manière dont les individus les utilisent et les partagent entre eux, dans ce cas, ce sont les groupes qui décident de la structure des connaissances.

Des exemples de systèmes organisant les connaissances sous forme de hiérarchie de thèmes sont Meta4 KnowNet[®], développé par la société Meta4⁷, Microsoft[®] SharePoint[™] Portal Server 2001⁸ de la société Microsoft, KnowCat⁹ (Knowledge Catalyser) qui est un système développé par Universidad Autónoma de Madrid¹⁰, KnowNet tool, qui est un système développé dans le cadre du projet européen ESPRIT et enfin Sintagma qui est un outil développé par la société Carrot informática y Comunicaciones¹¹.

Parmi les systèmes qui organisent les connaissances en tenant compte des critères des utilisateurs, citons par exemple Dynasites (Dynamic, Extensible and Integrated Information Spaces) développé dans l'université de Colorado¹². De manière similaire, le produit **Plumtree Corporate Portal**, développé par la société Plumtree Software¹³ organise les connaissances dans des espaces d'utilisateurs communs nommés MyPages, dans le cas où elles sont partagées par plusieurs utilisateurs, elles sont converties en OurPages. Aussi, Le système **Zaplet Appmail Suite** de la société Zaplet¹⁴ organise les connaissances qui sont en réalité des documents appelés appmail en assemblant des blocs de documents créés collaborativement par des utilisateurs intéressés par cet appmail.

Dans quelques uns de ces outils, l'utilisateur peut intervenir pour proposer son opinion concernant des connaissances du système et même de la structure des connaissances. Microsoft[®] SharePoint[™], Portal Server 2001, Meta4 KnowNet[®], KnowCat, Zaplet appmail Suite et Dynasites sont des exemples de tels systèmes.

Ces outils possèdent différents types d'utilisateurs : le lecteur ou le consommateur de connaissances, l'éditeur ou le producteur de connaissances, le coordinateur, qui a pour rôle de superviser les contributions des producteurs et enfin l'expert, ayant pour rôle d'évaluer le contenu

⁷ www.meta4.com

⁸ www.microsoft.com/sharepoint/

⁹ www.ii.uam.es/~rcobos/investigacion/knowcat/eng/fKC.htm

¹⁰ www.ii.uam.es/~rcobos/investigacion/knowcat/esp/intro.htm

¹¹ www.e-carrot.net/

¹² seed.cs.colorado.edu/Dynasites.Documentation.fcgi

¹³ www.plumtree.com/products/

¹⁴ www.zaplet.com

des connaissances. Tous les outils déjà cités intègrent des mécanismes de recherche permettant aux utilisateurs d'accéder aux connaissances dont ils ont besoin, quelques uns intègrent des mécanismes de catégorisation appliqués aux résultats de recherche obtenus. C'est le cas de Microsoft® SharePoint™ Portal Server 2001 et KnowCat. Quelques outils, tels que Meta4 KnowNet®©, vont plus loin et proposent des recommandations aux utilisateurs concernant les documents pouvant être les plus intéressants pour eux.

II.3.3.2. Les SKM Collaboratifs

L'objectif principal des systèmes de KM collaboratifs est de faciliter et d'assister les individus impliqués dans les processus de KM dans leurs travaux au sein des groupes. Nous distinguons trois types de systèmes de KM collaboratifs : les espaces partagés, les systèmes de recommandation et les systèmes d'apprentissage collaboratif.

A. Les espaces partagés

Ces systèmes fournissent aux utilisateurs une interface d'un espace partagé dans lequel les membres d'un groupe interagissent et communiquent dans le but de partager leurs connaissances et de créer de nouvelles connaissances de manière collaborative. Les systèmes de ce type ont tendance à offrir des fonctionnalités similaires telles que les fonctionnalités de communication (messagerie, forums de discussion, chat), les fonctionnalités de partage de contenu afin de partager des fichiers, des contacts ou des liens et enfin des fonctionnalités de travail collectif tels que le web browsing collectif, les outils d'édition Multi-Utilisateurs, et les agendas de groupes. Parmi les systèmes développés de ce type on peut citer BSCW¹⁵ (Basic Support for Cooperative Work), un système développé par le GMD (German National Research for Information technology) et Groove¹⁶ développé par Groove Networks.

B. Les Systèmes de recommandation

Les systèmes de recommandation sont basés sur des techniques de filtrage d'informations collaboratives assurant aux utilisateurs d'atteindre les informations qui les intéressent le plus en donnant juste leurs préférences. L'objectif d'un tel système est de chercher les informations jugées intéressantes à l'égard d'autres utilisateurs ayant des profils similaires. Généralement, dans le contexte du commerce électronique, un système de recommandation est un système qui recommande à son utilisateur une Liste de produits et l'aide à évaluer ces produits [Schafer et al., 2002].

Le premier système de filtrage collaboratif fut **TAPESTRY** développé par ZeroX Parc [Goldberg et a., 1992].

Plus tard, plusieurs produits et systèmes de recommandation ont été développés ; citons par exemple GroupLens, Ringo, EachMovie¹⁷.

Comme exemple de systèmes orientés connaissances, citons **NEWKNOW**, qui a été développé par la société NewKnow Network¹⁸. Ce système classe les nouvelles connaissances dans des

¹⁵ bscw.gmd.of

¹⁶ www.groove.net

¹⁷ www.research.compac.com/SRC/eachmovie

catégories et peut créer des relations entre des documents en analysant les consultations des utilisateurs de ces documents. Autres outils similaires, citons **JASPER II**, développé par British Telecommunication, le but de ce système étant de faciliter l'échange des connaissances tacites et explicites au sein d'une communauté d'intérêt [Davies, 2001]. Il y a aussi **COINS**, un outil développé par le GMD¹⁹ qui recommande les pages Web qui ont été récemment fréquemment consultées par les utilisateurs.

C. Les Systèmes d'apprentissage collaboratif

Nous évoquons dans cette partie les systèmes où les outils dédiés à l'apprentissage collaboratif. L'apprentissage collaboratif est une activité sociale impliquant une communauté d'apprenant qui partagent des connaissances et acquièrent de nouvelles connaissances, c'est le processus de « la construction sociale de la connaissance » [Jonassen et al., 1992].

Nous avons accordé de l'intérêt à de tels systèmes parce qu'ils permettent aux utilisateurs d'apprendre à travers un processus d'intégration, administration et dissémination des connaissances aux utilisateurs, trois aspects caractéristiques des Systèmes de Knowledge management. Les systèmes d'apprentissage collaboratifs partagent les aspects suivants :

- Un espace pour la communauté d'apprenant dans lequel ils peuvent échanger leurs idées et connaissances, utilisant les outils de collaboration intégrés dans l'espace,
- Les connaissances sont généralement structurées par thème. Et les unités de connaissances ne sont pas essentiellement des documents mais aussi des exercices, des études, des questions-réponses, etc.

Le premier exemple de ce type de systèmes est **WISE**²⁰. C'est un système pour l'acquisition de connaissances basé sur le Web. Son objectif est de fournir à ses utilisateurs (les apprenants) un outil de travail collaboratif didactique, ce qui leur permet d'apprendre et de répondre à des controverses scientifiques en concevant et discutant des solutions. En plus, WISE supporte d'autres communautés d'intérêt autres que celles des apprenants, par exemple la communauté des professeurs leur offrant la possibilité de créer leur propre espace dans lequel ils pourront partager et échanger les idées et les références à propos d'un thème et de la manière dont il peut être structuré.

Dans le système **GENTLE**²¹ développé au sein de l'université de Graz en Autriche, les connaissances prennent la forme d'une librairie statique (publications et revues digitales) et d'une bibliothèque dynamique (sites web, bases de données d'expertises, forums de discussion, etc) et le tout peut constituer une collection de connaissances classifiées par thèmes, cours et termes [Dietinger et al., 1998].

D'autres systèmes tels que **DEGREE** (Distance éducation Environnement for GRoup-ExpenencEs) organisent les connaissances des apprenants exprimés sous forme d'idées. DEGREE a été développé par **UNED** (Universidad Nacional de Educación a Distancia : Université

¹⁸ www.newknow.com

¹⁹ orgwis.gmd.of/projects/coins

²⁰ wise.berkeley.edu/

²¹ wbt-2.iicn.edu/product

Nationale d'Enseignement à Distance en Espagne), et permet à ses utilisateurs d'échanger des idées et des contributions en vue de parvenir à un consensus [Barros et al., 2000]. Un autre exemple de systèmes est le système CSILE (Computer Supported Intentional Learning Environment), développé par Marlene Scardamalia et Carl Bereiter de l'Ontario Institute for Studies in Education à Toronto²² [Scardamalia et al., 1999].

II.3.3.3. Les systèmes centrés sur la représentation de connaissances

Nous distinguons dans cette catégorie trois types de systèmes: les systèmes médiateurs d'informations, les systèmes basés sur les ontologies et les bibliothèques digitales.

A. Les Systèmes médiateurs d'informations

L'objectif principal de ces systèmes est de fournir aux utilisateurs une interface à travers laquelle ils consulteront des ressources concernant un domaine particulier, probablement via le web; ces ressources pouvant être distribuées et hétérogènes, l'interface masque à l'utilisateur l'aspect distribution et hétérogénéité et lui offre une vue centralisée et homogène.

Dans notre analyse, nous avons inclus deux systèmes qui sont des projets de recherche PICSEL²³ et WebKB; le premier développé par le laboratoire LRI (Laboratoire de Recherche en Informatique-Université Paris XI) et le CNET (Centre National d'études des télécommunications), le second développé par The School of Information Technology de Griffith University en Australie.

Ces deux outils possèdent deux caractéristiques techniques :

- Les processus de capture et d'intégration de connaissances sont effectués a priori et ne nécessitent aucune interaction avec l'utilisateur final du système ;
- Ces systèmes incorporent des langages pour la description et l'indexation des sources de connaissances et de leurs contenus. Plus précisément, PICSEL utilise un langage qui combine les descriptions logiques et les règles (nommés Datalog) développé spécialement pour modéliser et relier les informations. Le système WebKB utilise le standard RDF (Ressource Description Framework) pour définir les relations entre les unités de connaissances obtenues à partir de sources variées.

Ce type de système est centré sur la création de structures virtuelles qui révèlent des relations dans les connaissances et les rend disponible de manière transparente, sans donner aucune importance à une possibilité d'interaction entre les utilisateurs.

B. Les systèmes basés sur les ontologies

Dans [Bensta, 2005a] nous avons présenté une définition de l'ontologie ainsi que les différents domaines d'application des ontologies. Les premières utilisations du concept d'ontologie fut dans le domaine des web sémantiques. Plus tard, les ontologies ont été utilisées comme base de nombreux systèmes technologiques. Les systèmes de KM basés sur les ontologies ont été utilisés dans plusieurs environnements. Exemples : dans le domaine du business, on trouve des systèmes comme WebCADET [CaldWell et al., 2000] qui est un système basé sur le web pour le support

²² www.ed.gov/pubs/EdreformStudies/EdTech/csile.html

²³ www.lri.fr/~picsel/

de la prise de décision en intégrant un moteur d'inférence. Planet-Onto est un autre exemple, c'est un système développé comme un administrateur intelligent de news pour des groupes de travail interinstitutionnels.

Il existe d'autres systèmes comme C-Web²⁴ et IBBROW [Benjamins, 2000], chacun de ces systèmes offre, à sa manière, des modèles conceptuels pour un KM distribué dans des espaces de travail dans lesquels les informations relatives à un domaine ont une structure connue a priori : par exemple les connaissances concernant les groupes académiques.

Finalement, les ontologies ont été utilisées pour supporter les systèmes de recherche d'informations dans le Web : Ontobroker²⁵ utilise un langage robuste pour générer des structures de connaissances du web et des moteurs de recherche sémantiques.

C. Les bibliothèques numériques

Les bibliothèques numériques sont des systèmes combinant les technologies de communication et celle du stockage des informations numériques pour reproduire, imiter et étendre les services que les bibliothèques conventionnelles offrent comme par exemple la collection, le catalogage, l'administration et la dissémination des informations bibliographiques.

Un exemple de tels systèmes est COSPEX²⁶ (COncceptual SPace EXplorer) qui capture les informations de sources éparpillées et permet à l'utilisateur de construire sa propre bibliothèque numérique.

II.3.3.4. Conclusion

Les SKM sont des systèmes technologiques extrêmement complexes et sophistiqués, leur développement et leur implémentation est un processus difficile pour plusieurs raisons :

- la complexité technique des fonctions « intelligentes » qui caractérisent les SKM des systèmes traditionnels, en plus et du volume énorme de données, de documents, de messages, de liens, de critères de personnalisation que les SKM sont supposés gérer.
- la complexité organisationnelle de la solution qui affecte d'une part les processus de connaissances (création, organisation, partage, transfert) et d'autres part le rôle et la responsabilité des unités organisationnelles.
- la complexité humaine, puisqu'un SKM est supposé supporter les activités cognitives et sociales des individus au sein des organisations.

Une réussite d'une démarche de Knowledge Management (KM), de gestion des connaissances, consiste à répertorier les savoirs mis en œuvre, à les rendre explicites lorsqu'ils sont tacites, à les organiser au sein d'un système d'information ouvert à tous. Une démarche KM peut donc avoir plusieurs objectifs : « offensif », lorsqu'il s'agit de mieux combiner les connaissances pour améliorer la qualité ou la faculté d'innovation ; « défensif » lorsqu'il s'agit de préserver, ou de transmettre à de nouveaux arrivants, le capital de connaissances d'une entreprise soumise à un changement ou au départ en retraite de ses personnels.

Les connaissances collectivisées, les expériences capitalisées d'une entreprise doivent être préservées dans une mémoire d'entreprise dans un but de réutilisation plus tard.

²⁴ cweb.inria.fr

²⁵ ontobroker.semanticweb.org

²⁶ <http://www.r.dl.itc.u-tokyo.ac.jp/~sugi/cospex/wm/index.htm>

Ces expériences passées, les projets réalisés et mémorisés dans une entreprise forment une partie intégrante d'un capital que les décideurs dans une entreprise investissent dans la réalisation de nouveaux projets. Ce capital doit être préservé dans une mémoire appelée mémoire de projet, mémoire qui doit être capable de « décrire l'historique d'un projet » [Pomian, 1996] et « L'expérience acquise pendant la réalisation de projet » [Tourtier, 1995].

Dans le prochain paragraphe seront présentées quelques définitions de la mémoire de projet, l'utilité d'une mémoire de projet, les fonctionnalités d'une mémoire de projet, les bénéfices attendus d'une mémoire de projet et un état de l'art lié à la mémoire de projet.

II.4. Etat de l'Art des méthodes de formalisation des connaissances

II.4.1. Introduction

L'utilisation d'une méthode a pour avantage de rationaliser un processus, notamment lorsque celui-ci est complexe, et d'éviter les omissions. La nature intangible de la connaissance rend l'utilisation d'une méthode très intéressante : une méthode de formalisation des connaissances propose une manière de faire pour rendre visible une connaissance qui était tacite ou implicite.

D'après Magalie BRIQUET²⁷ une méthode s'inspire souvent de l'expérience humaine et de ce qui ne fonctionne pas. De plus, une méthode répond, entre autres, aux problèmes suivants :

- La complexité d'un système de conception avancé
- L'efficacité requise en termes de coûts et de temps
- Une quantité énorme d'information à traiter et à assembler
- La connaissance disséminée entre plusieurs utilisateurs
- Le besoin d'intégration des aspects économique, humain et technique
- La nécessité de s'assurer la participation des utilisateurs.

Les méthodes de formalisation ont toutes pour objectif « d'assister le professionnel dans la production d'une connaissance explicite, diffusable, réutilisable, évolutive, venant enrichir une mémoire organisationnelle » [Prax, 2000]. Les méthodes présentées ci-après procèdent toutes par recueil des connaissances auprès de l'expert, ou recueil d'informations dans un système d'information du type data mining. Même si elles ont été conçues à d'autres fins, elles ont pour résultat, intermédiaire ou final, de formaliser, sous des formes différentes, les connaissances, les rendant ainsi « visibles ».

Le mécanisme de la formalisation comporte deux grandes étapes :

- Dans un premier temps, il y a acquisition, recueil des connaissances auprès du détenteur initial
- Dans un second temps, les connaissances acquises, sont véritablement formalisées.

La formalisation peut prendre la forme d'une modélisation, plus ou moins formelle, ou d'une transcription structurée, sous forme de fiches par exemple, ou encore d'une combinaison des deux.

²⁷ Magalie BRIQUET <http://www.crpht.lu>

Les méthodes présentées ci-dessous répondent toutes à la définition proposée par Jean-Yves PRAX [Prax, 2000].

Nous allons présenter dans un premier temps les méthodes de capitalisation classique puis nous nous focaliserons sur les méthodes de capitalisation des mémoires de projet.

D'après [Dieng et al, 2000] les méthodes identifiées ont été segmentées comme suit:

1. Les méthodes de capitalisation de connaissances : Rex, MKSM, CYGMA, Atelier FX, Merex, Componential Framework, CommonKADS, KOD.
2. Les approches de capitalisation de mémoire de projet : IBIS, QOC, DRCS, DRAMA, EMMA, SAGACE, DIPA

II.4.2. Méthodes de capitalisation des connaissances

II.4.2.1. La méthode REX

La méthode REX a été définie au départ dans le but de capitaliser les expériences de conception de réacteurs nucléaires au sein du CEA [Malvache et al., 1993]. La méthode a été ensuite utilisée dans divers types d'applications telle que la spécification de systèmes de contrôle dans le domaine électrique, la conception de générateurs électriques, la conception aéronautique, etc. [Dieng et al., 2000].

Le principe de base de la méthode consiste à constituer des « éléments d'expériences », extraits d'une activité quelconque et à restituer ces éléments pour qu'un utilisateur puisse les valoriser. Les éléments d'expérience ainsi définis sont stockés dans une mémoire d'expérience appelée (CEMem) avant d'être restitués.

Différents types de fiches de connaissances peuvent être gérés dans une application REX. Par exemple des fiches d'expérience (Figure 7) sont typiquement définies par :

- un contexte d'un problème ou d'une situation de résolution de problème,
- une description ou corps et
- une liste de références.

Le corps est décomposé lui-même en trois parties :

- une description neutre d'un fait,
- une opinion propre et des commentaires et
- des recommandations.

Les éléments d'expérience sont définis principalement à l'issue des entretiens auprès d'experts et à partir des documents relatant une activité (i.e. documents de synthèses, bases de données).

On remarque que dans la fiche des éléments d'expertise, on y présente des éléments relatifs à une description d'une expérience de résolution de problème et de prise de décision. Par contre, on n'y trouve pas des représentations des éléments du contexte au sens caractéristiques du déroulement de l'activité.

Entête
<p>Nom: Traversée d'un carrefour Origine: Expert1, Référence entretien N.3 Auteur: Cogniticien Date d'émission: Janvier 1996 Domaine: Psychologie Contexte: Analyse des stratégies adoptées par un conducteur pour traverser la route</p>
Corps
<p>Observation: Le choix de la stratégie (traverser en une seule fois versus traverser en plusieurs fois) dépend d'un certain nombre d'attentes liées au volume du trafic. Si le trafic est faible, il y a de très grandes chances que le conducteur n'ait pas besoin de s'arrêter au milieu. Si le trafic est dense, on va être obligé de s'arrêter au milieu. Donc, quelqu'un qui arrive et voit que le trafic est faible, pense n'avoir pas besoin de traverser au milieu et choisit la stratégie de traverser sans arrêt. Et comme il s'attend à pouvoir traverser sans arrêt, on peut faire l'hypothèse qu'il est préparé à ce qu'il n'y ait personne.</p> <p>Hypothèses: Trafic faible implique une stratégie de traversée sans arrêt au milieu. Trafic dense implique une stratégie de traversée avec arrêt au milieu.</p> <p>Commentaires: Les attentes du conducteur sont différentes selon la stratégie choisie:</p> <ul style="list-style-type: none"> - Stratégie de traverser en deux coups: le conducteur simplifie le problème en s'occupant d'abord du flux de gauche, puis du flux de droite. - Stratégie de traversée en un coup: la solution la plus difficile. La plus dure, parce qu'il faut évaluer le créneau, simultanément de chaque côté, avec des durées de validité d'informations qui se conditionnent mutuellement, et qui supposent que l'on va même faire vite pour traverser.

Figure 7 : Un exemple d'un élément d'expérience dans le domaine de l'analyse d'accident [Dieng et al, 2000].

Types de représentation

La méthode REX utilise des éléments d'expérience pour décrire des situations de prise de décision précises. Il s'agit donc de fiches de définition de problème de données et de description de sa résolution. La méthode ne permet pas de décrire des modèles génériques pour représenter les processus de résolution de problème.

Formes de présentation

Le mode de présentation est textuel. On le voit clairement dans les fiches d'éléments d'expérience notamment dans la partie *corps* où il y a les observations, les hypothèses et les commentaires.

II.4.2.2. La méthode CYGMA

CYGMA (CYcle de vie et Gestion des Métiers et des Applications) a été définie par la société KADE-TECH. Cette méthode a été appliquée dans les industries manufacturières et spécialement dans l'activité de conception (bureau d'études, de méthodes et d'industrialisation). CYGMA prévoit six catégories de connaissances industrielles [Dieng et al., 2000]:

Connaissances singulières, terminologiques, structurelles, comportementales, stratégiques et opératoires. La méthode permet, en se basant sur ces catégories, de définir des référentiels métiers appelés « Bréviaire de connaissances de filière métier » et de Bases de Connaissances, exploitables par des algorithmes de raisonnement déductif. Ces Bases de connaissances sont appelés AMI (Applications Métier Industrielles ou Assistants Métier de l'Ingénieur).

La méthode CYGMA préconise des entretiens avec les experts et une étude de la documentation de l'entreprise afin de définir un « bréviaire de connaissances ». Ce bréviaire sera ensuite validé avec les experts. Les connaissances dans ce bréviaire sont structurées en quatre documents: le glossaire métier, le livret sémantique, le cahier de règles et le manuel opératoire [Dieng et al, 2000].

Types de représentation : La méthode CYGMA représente la résolution de problème à travers un modèle de processus de résolution. On n'y trouve pas de représentation par situation et par cas. De même le contexte de l'activité est représenté à travers un nombre de modèles complémentaires.

Forme de présentation : C'est plutôt la présentation graphique qui domine dans la méthode CYGMA.

II.4.2.3. L'approche Componential Framework

La méthode « Componential Framework » a été définie dans le cadre de l'acquisition des connaissances pour développer des systèmes à base de connaissances. Cette méthode a été ensuite adaptée pour supporter la gestion des connaissances dans une entreprise. Dans cette méthode, une activité peut être définie selon trois perspectives [Steels, 1993] : tâche, information et méthode (Figure 8).

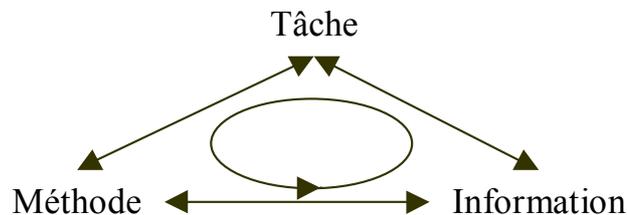


Figure 8 : Les trois perspectives : tâche, méthode et information.

Une tâche décrit les objectifs à atteindre, l'information met en avant les informations et les connaissances consultées et construites pour réaliser les tâches et la méthode met en évidence comment les informations ont été utilisées pour réaliser les tâches. La définition de ces trois perspectives forme un cycle dans lequel chaque perspective évoque des connaissances à définir dans une autre perspective.

On trouve dans l'approche « Componential Framework » des représentations de quelques éléments de l'environnement de l'activité. Par exemple, la décomposition des tâches est décrite graphiquement grâce à un arbre de tâches appelé "structure de tâches" et les différents types d'information sont organisés sous forme de deux sortes de modèles : modèle du domaine et

modèles de cas. La résolution de problème de son côté est prise en compte par l'approche à travers un « diagramme de contrôle » qui permet de décrire graphiquement un flot de contrôle sous forme d'un automate fini, auquel est attribué un état initial, un état de succès et un état d'échec. Aussi, on trouve dans le modèle de cas une spécialisation de certaines informations du modèle du domaine, évoquées dans la résolution d'un problème donné.

Types de représentation : L'approche « Componential Framework » permet de décrire des modèles de représentation de l'activité globale d'une expertise mais nous pouvons également, comme évoqué ci-dessus, trouver des modèles de cas qui correspondent à des descriptions de situations spécifiques de résolution de problèmes.

Forme de présentation : « Componential Framework » utilise des présentations mixtes : graphiques et textuelles. On note qu'un logiciel a été défini comme support de la méthode. Ce logiciel permet de représenter les différentes perspectives ainsi que les diagrammes de relations entre ces perspectives. Des liens hypertextes sont aussi définis. Ils permettent une navigation entre les descriptions graphiques et textuelles ainsi que les logiciels associés.

II.4.2.4. L'approche EMMA

L'approche EMMA (Evolution Memory Management Assistant) [McCullough et al, 1998] est une approche de représentation des connaissances utilisées dans un projet de conception de logiciels. Cette approche fournit un modèle de représentation appelé « MetaModel » qui permet de décrire le processus de conception d'un logiciel sous forme de buts à atteindre. Le principe de base de l'approche consiste à distinguer dans un processus de conception :

- Les buts à atteindre,
- Les plans définis pour atteindre ces buts,
- Le contexte des plans définis,
- Les changements apportés ainsi que l'évolution du processus et
- Le processus de conception est donc représenté comme une structure de solution qui met en association les buts avec les plans qui leur sont destinés.

Un plan est décomposé en sous-buts alternatifs. Un ensemble de primitives est fourni à cet effet. Notons parmi elles:

- But de la solution associée : Il est défini par son nom, sa description et des références.
- Spécification de la solution associée : Elle est définie par le type, la description et un ensemble de propriétés.
- Contexte. Il est défini par les termes utilisés, les hypothèses de solution et les ressources,
- Elaboration : Elle est définie par les plans alternatifs, un plan actif choisi et les justifications du choix. Ces justifications peuvent avoir des liens avec des documents hypertextes ou avec une structure d'arguments provenant d'autres types de représentation comme celles définies dans la méthode IBIS,
- Informations sur la collaboration : Elles mettent en évidence les propriétaires d'un but et les sous-traitants fournisseurs du plan qui permet d'atteindre le but et
- Informations sur l'évolution du but : Il s'agit d'une description des changements qui affectent le but.

Ces primitives sont associées à la description d'un but. De même un plan peut être défini par :

- **Elaboration** : Qui met en évidence le but du travail (une spécification du but père) et un ensemble de sous buts permettant d'atteindre le but principal,
- **Informations sur la collaboration** : Elles présentent les sous-traitants, auteurs du plan et
- **Informations sur l'évolution du plan**. Elles sont décrites par :
 - des événements de changement, émis par le plan ou souscrits par les buts père et
 - des réponses anticipées aux changements, des enregistrements des changements ou d'exceptions (ils sont décrits par le nom, la description, les circonstances et les actions).

L'outil EMMA est développé en Java. Il est basé sur des interfaces hypertextes. Il permet aux concepteurs et aux gestionnaires de logiciels complexes d'explorer, de choisir et de documenter de manière coopérative les solutions envisagées.

II.4.2.5. La méthode SAGACE

La méthode SAGACE [Penalva, 1990] et [Penalva, 1994], développée au sein du CEA (Commissariat à l'Energie Atomique) a comme principe de base la modélisation des connaissances statiques décrivant un système de production. Cette modélisation fournit une base de dialogue entre les différents acteurs dans un projet et une aide au choix des moyens d'actions. Elle peut être ainsi exploitée pour mémoriser la réalisation d'un projet de production d'un système. Le système produit peut être à « forte automatisation », à « forte composante humaine » ou à « forte identité » comme par exemple, un système à base de connaissances.

	<i>Activité</i>	<i>Fonctionnement</i>	<i>Evolution</i>	
<i>Vision Fonctionnelle</i>	PROCESSUS	PROGRAMME	SCENARIOS	<i>Ce que fait le Système</i>
<i>Vision Organique</i>	RESEAU OPERANT	RESEAU LOGISTIQUE	RESEAU AUXILIAIRE	<i>Ce qu'est le système</i>
<i>Vision Stratégique</i>	PILOTAGE	ADAPTATION	ANCIPIPATION	<i>Ce que décide le système</i>
	<i>Performance</i>	<i>Stabilité</i>	<i>Cohésion</i>	

Figure 9 : Les trois visions définies dans SAGACE.

Cette modélisation se base sur trois types de visions : fonctionnelle, organique et opérationnelle (Figure 9). Les visions sont décrites à l'aide de trois principaux éléments : processeur, flux et observateur (Figure 10).

Un Processeur décrit une activité qui réagit avec son environnement sous forme d'un Flux. Certaines caractéristiques du Flux sont observables. Elles sont appelées Observateurs.

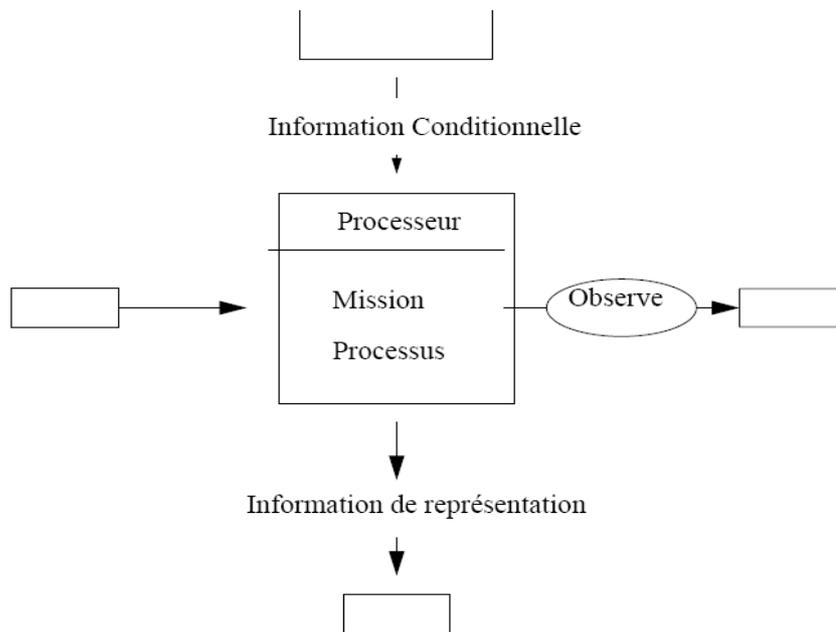


Figure 10 : Trois éléments qui représentent les données dans les visions.

Un processeur change de statut d'une vision à l'autre :

- Dans la vision fonctionnelle, un processeur décrit : une fonction, les contraintes de conception, les objectifs et les activités mises en œuvre,
- Dans la vision organique, un processeur décrit : les ressources et les moyens,
- Dans la vision opérationnelle, il décrit la tâche et les acteurs.

Par exemple, dans les systèmes technologiques, les trois visions représentent :

- les fonctions de service (vision fonctionnelle),
- les composants (vision organique),
- l'activité des utilisateurs (vision opérationnelle).

Dans une organisation, la vision organique décrit les acteurs humains et la vision opérationnelle décrit les conditions de leur coopération comme les règles instaurées et les objectifs sous-jacents. Dans un système à base de connaissances, la vision fonctionnelle représente la stratégie de résolution alors que la vision organique décrit les inférences. La vision opérationnelle contient les tâches contrôlant le raisonnement.

Un élément du Flux est représenté par son nom et son support ou sa forme. Ces éléments sont classés en quatre types :

- Entrée (axe horizontal et flèche de gauche),
- Sortie (axe horizontal et flèche de droite),
- Information conditionnelle (axe vertical et flèche d'en haut) ; ces informations représentent une condition d'exécution, une contrainte de pilotage ou une contrainte de sélection de moyens et
- Information de représentation (axe vertical et flèche vers le bas) ; ces informations peuvent décrire l'exécution de la mission, le déroulement du processus et une configuration du processeur.

Un observateur met en avant une caractéristique observable d'un élément du Flux. Il est représenté par une ellipse avec un nom reflétant la nature de la caractéristique, (par exemple, Débit, etc.). Il est présenté sur la flèche du Flux. Un observateur à des attributs comme défaut, alarme, tendance, etc.

L'ensemble des observateurs définis pour un système reflète le comportement du système, son évolution et sa réaction avec son environnement.

II.4.2.6. La méthode MKSM

MKSM (Methodology for Knowledge System Management) [Ermine et al, 1996] est une méthode qui a été développée au CEA avec un objectif de gestion des connaissances. La méthode se base principalement sur un triangle sémiotique (Figure 11) où trois dimensions sont prises en compte : la syntaxe, la sémantique et la pragmatique. L'analyse des connaissances dans une organisation suivant ces trois dimensions consiste à considérer l'information (syntaxe), la signification (sémantique) et le contexte (pragmatique). L'analyse de ces dimensions est guidée par l'étude du traitement des données, des tâches et de l'activité du domaine (Figure 11).

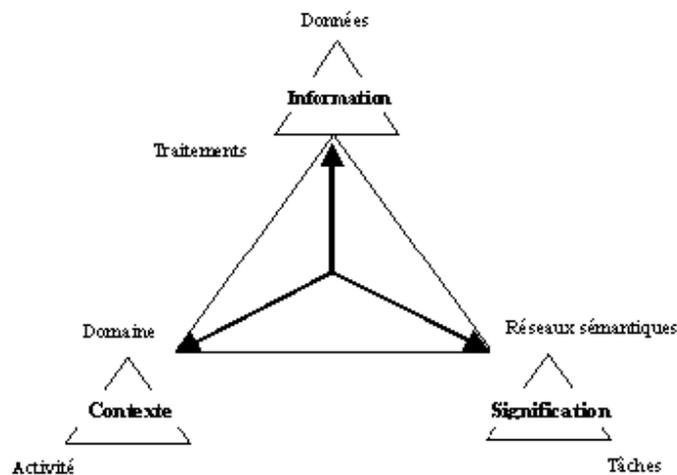


Figure 11 : Le triangle sémiotique.

Une modélisation du système des connaissances doit être aussi considérée. Cette modélisation met en évidence les flots de connaissances et d'informations, les acteurs (ou agents) producteurs et consommateurs de la connaissance. Chaque agent est défini par son rôle dans l'entreprise, les informations consommées, les informations produites, les connaissances consommées et les connaissances produites. Cette modélisation peut être représentée sous forme d'un diagramme SADT dans lequel le système opérant représente des informations sur les agents, le système de décision définit l'environnement et la capacité d'organisation et de structuration et le système d'information rassemble les documents et les bases de données de l'entreprise.

Le patrimoine de connaissances comporte deux éléments principaux : le livre de connaissances et le système opérationnel de gestion de connaissances.

Ce dernier peut être défini sous forme de systèmes de supervision, d'aide à la décision, d'information et de documentation, de formation, de veille technologique ou stratégique et de gestion de la qualité. Le livre de connaissances rassemble des modèles de connaissances en complément des documents de l'entreprise comme les fiches, les plans, les documents techniques, les images, les références, etc. La méthode MKSM préconise des techniques pour représenter des modèles de connaissances suivant le triangle sémiotique défini ci-dessus (Figure 11). Nous décrivons dans ce qui suit cette modélisation. La méthode MKSK se base sur deux modèles de connaissances :

a- Modélisation du contexte

Deux modèles sont définis pour représenter le contexte: un modèle du domaine et un modèle de l'activité.

- Le modèle du domaine: Deux types de connaissances sont mis en évidence dans le modèle du domaine: 1- le type du flot des connaissances (matériel, énergie, information, etc.) et 2- les sources et cibles des connaissances. Dans ce modèle, une source est définie par une action (peut être décomposée en plusieurs actions), qui crée le flux et par un ensemble d'événements qui activent le processus. De même, la cible est décrite par une action qui représente l'effet du flux.
- Le modèle de l'activité: Il représente le flot de données dans les activités. Il est défini sous une forme proche d'un actigramme SADT. Ce flux décrit les entrées, sorties, ressources et acteurs d'une activité.

Une activité peut être décomposée en plusieurs activités, représentées par plusieurs diagrammes.

b- Modélisation de la signification

La signification est modélisée avec deux types de modèles : un modèle de concepts et un modèle de tâches.

- Le modèle de concepts: Un concept représente une catégorie d'objets qui partagent les mêmes propriétés qui sont définies comme attributs du concept. Un concept peut avoir des instances. Une hiérarchie de concepts peut être définie en utilisant la relation de "spécialisation". Une autre relation "Valeur" permet de définir les valeurs d'un concept.

D'autres types de relations sont aussi définis entre les concepts comme les liens de cardinalité. Le réseau de concepts et relations est proche d'un réseau sémantique où une sémantique peut être attribuée à un lien.

- Le modèle de Tâches: Il s'agit d'une « représentation de la stratégie mise en œuvre pour résoudre les problèmes ». On y distingue deux aspects :

1- la résolution de problèmes, qui doit être modélisée en se posant les questions suivantes « Quel type de tâche doit-on résoudre ? » et « Comment résout-on généralement ce type de tâche ? »,

2- la manipulation de la connaissance statique qui doit être définie en évoquant la question suivante: « comment utiliser cette connaissance statique pour résoudre le ou les problèmes posés ? ». Un modèle de tâches est représenté sous forme d'une hiérarchie de tâches. Le type d'une tâche donnée exprime le contrôle exercé sur les sous-tâches qui la décomposent. Différents types de contrôle peuvent être exprimés comme: exécution séquentielle, alternative, en parallèle et répétitive. L'arbre de décomposition de tâches en sous tâches est appelé « flot de contrôle ».

Un outil support de la méthode permet de représenter les diagrammes et réaliser des recherches sur les connaissances. La méthode MKSM a été appliquée au sein du CEA dans différents types d'application allant de la biologie jusqu'aux technologies nucléaires. Elle a été également utilisée dans des applications dans le domaine de l'électricité et de la gestion bancaire.

II.4.2.7. La méthode CommonKADS

La méthodologie KADS (Knowledge Acquisition and Design System ou Knowledge Acquisition and Documentation Structuring), est née en 1985 dans le cadre du programme européen Esprit I. Ce projet est lancé par quatre chercheurs : Anne Brooking [Brooking, 1999] du KBSC, South Bank Polytechnic (Royaume-Uni), Joost Breuker et Bob Wielinga de l'Université d'Amsterdam et Mike Rogers du CEC, dans le cadre d'un projet ESPRIT, KADS. Le projet est reconduit en 1990, sous le nom de KADS II, dans le cadre du programme européen Esprit II. La méthodologie est améliorée dans le but d'en faire un standard commercial, notamment en Europe. On nomme désormais cette méthodologie **CommonKADS**. L'objectif premier de la méthode est d'*aider à la modélisation des connaissances d'un expert ou groupe d'experts dans le but de réaliser un système d'aide à la décision basé sur la connaissance (SBC) - ou système informatique à base de connaissances*.

La méthode CommonKADS permet de traiter tout le processus d'acquisition des connaissances, du recueil au développement d'un système complet. C'est une méthode issue des limites d'autres approches, telles que les méthodes dirigées par l'implémentation, et dirigée, à l'instar de MKSM, par des modèles.

CommonKADS propose une modélisation conceptuelle des connaissances en plusieurs étapes successives, soit six modèles, allant du général au particulier : organisation, tâche, agent, communication, connaissance et conception. Les quatre premiers modèles présentent un intérêt particulier pour l'analyse préalable à la capitalisation des connaissances.

- Le modèle organisationnel : décrit l'entreprise dans son ensemble avec ses grandes fonctions,
- Le modèle des tâches : décrit les tâches réalisant les fonctions identifiées dans le modèle d'organisation,
- Le modèle des agents : décrit les agents, humains ou informatiques, impliqués dans la réalisation des tâches,
- Le modèle de communication (ou modèle de collaboration) : rend compte de la communication homme-machine,
- Le modèle d'expertise (étape la plus importante de la modélisation CommonKADS), ou modèle de connaissance : permet de modéliser l'expertise nécessaire à la réalisation des tâches pour les agents. Il permet de clarifier la structure des tâches à base de connaissances. Les modèles de connaissances de CommonKADS sont plutôt destinés à modéliser une future application et non pas à rendre fidèlement compte de la connaissance humaine. Les fonctions sont décrites indépendamment de la structure des données, pour favoriser la réutilisabilité. Les fonctions s'expriment en termes de rôles orientés tâche, c'est-à-dire d'une ontologie des objets manipulés par la tâche, exprimée de manière indépendante du domaine d'application. Ainsi, en diagnostic, on parlera de panne, de symptôme observable, etc., sans faire référence à un domaine particulier et
- Le modèle de formalisation (ou modèle de conception) : traite spécifiquement de la conception d'un système à base de connaissances destiné à implémenter les connaissances modélisées.

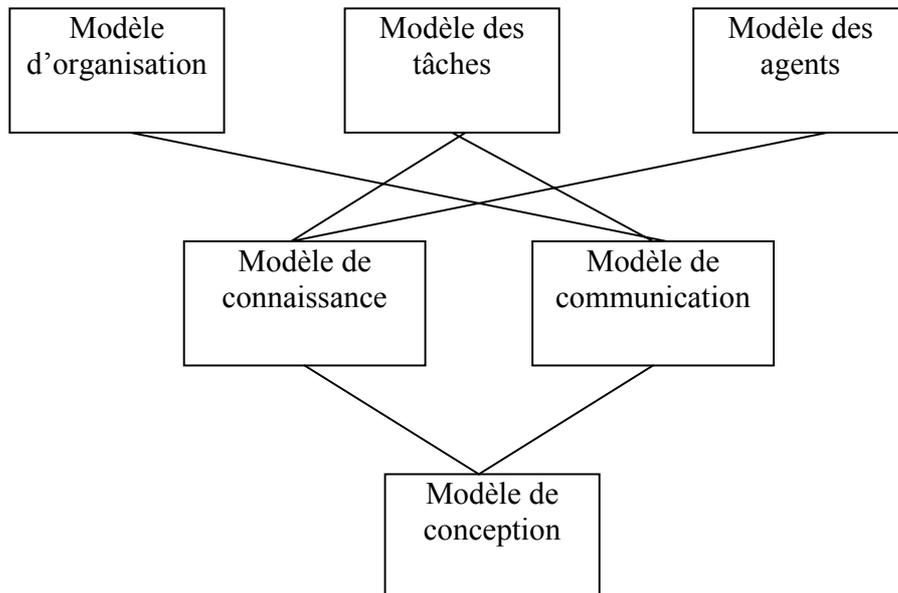


Figure 12 : les six modèles CommonKADS- [Dieng et al. 2000].

La méthode CommonKADS fait la distinction entre la connaissance du domaine, la connaissance d'inférence et la connaissance de la tâche dans la modélisation de la connaissance. La méthode est orientée méthode de résolution de la tâche où la connaissance de la tâche guide l'identification des connaissances du domaine.

Le modèle de connaissance structure la connaissance métier que le cogniticien a acquis durant sa collaboration avec l'expert, afin de réaliser une tâche. Il se décompose en trois grandes catégories de connaissance, qui sont :

- **la connaissance du domaine**, qui contient la connaissance spécifique à un domaine d'application. Par exemple, la connaissance du domaine d'une application de diagnostic médical pourrait contenir les définitions de « patient », « symptôme », et « test », aussi bien que les relations existantes entre ces types.
- **la connaissance d'inférence**, qui représente les outils de base pour utiliser la connaissance du domaine. Deux exemples d'inférences dans l'application du diagnostic médical pourraient être une inférence *supposée*, qui spécifie les symptômes d'un patient, et une inférence *vérifiée*, qui identifie les tests, utilisés pour certifier qu'un patient est victime d'un certain facteur qui a causé les symptômes observés.
- **la connaissance de la tâche**, détermine les objectifs de l'application (la tâche) et les moyens pour les réaliser (décomposition en sous-tâches, en méthodes et en inférences). Par exemple, une application de diagnostic médical pourrait avoir *diagnostic* comme tâche de plus haut niveau, qui pourrait être réalisée à travers une boucle d'invocations des inférences *supposer* et *vérifier*.

II.4.2.8. La méthode KOD

La méthode KOD a été développée en 1988 par Claude VOGEL [Vogel, 1988], un anthropologue, pour fournir un support à l'activité Intelligence Artificielle de CISI Ingénierie. Elle a été diffusée par une société de services (CISI) et a longtemps été concurrente de l'approche CommonKADS, bien qu'elle en soit très différente. La méthode KOD, fondée sur une approche anthropologique, utilise des techniques issues de l'ethnologie et de la linguistique, ce qui n'avait pas été fait auparavant, afin que le transfert des connaissances entre l'expert et l'interviewer soit optimal, dans la construction d'un système à base de connaissances.

Parmi les outils proposés pour le transfert des connaissances, KOD accorde une grande importance à la langue. En effet, la langue est le véhicule de l'information entre les différents intervenants d'un projet, en l'occurrence ici, le projet « recueil des connaissances » entre l'expert et l'interviewer.

« KOD propose d'effectuer ce procès de licitation et de réduction de la connaissance autour de trois modèles : pratique, cognitif et informatique, et de trois paradigmes : représentation, action et interprétation » [Dieng et al., 2000]:

KOD est une méthode qui se déroule en trois grandes phases :

1. Recueil des connaissances - très importante car elle garantit la qualité du résultat
 - l'originalité de la méthode repose sur l'étude systématique du texte
2. Exploitation des textes transcrits suite au transfert des connaissances
3. Construction d'un modèle d'expertise.

La méthode KOD se concentre sur l'expert, en donnant un rôle prépondérant à l'étude des entretiens. A partir du discours de l'expert, de sources documentaires, d'observations de l'analyse, la méthode KOD produit une spécification de l'expertise précisant :

- les domaines de compétence,
- les phases de mise en œuvre de l'expertise,
- les *Taxèmes* manipulés,
- les *Actèmes* actives et
- *Inférences* utilisées.

II.4.3. Les Approches de capitalisation de mémoire de projet

II.4.3.1. La méthode IBIS

Cette méthode a été définie dans les années 70 par Horst Rittel et al. [Buckingham Shum, 1997]. L'objectif de la méthode était à l'époque de fournir une structure pour le dialogue mené lors de la résolution de problèmes complexes de conception. Depuis, la méthode a évolué et plusieurs outils supports ont été développés. La méthode IBIS [Conklin et al. 1998] est notamment utilisée pour représenter la logique de conception (*Design Rationale*) dans un projet. La prise de décision comporte trois éléments: Questions (Issues), Positions et Arguments et chaque participant peut prendre une position, en proposant une réponse à la question posée. Il appuie sa position avec des arguments (Figure 13).

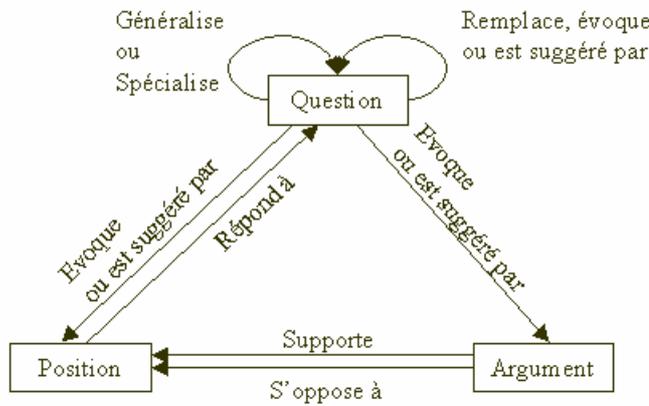


Figure 13 : Les principaux éléments de la méthode IBIS [Conklin et al. 1998].

La méthode IBIS représente exclusivement l'espace de la résolution de problème à travers des situations de prise de décision mettant en avant la question discutée, les positions prises par rapport à chaque question ainsi que les arguments émis pour appuyer telle ou telle position. Par ailleurs, les autres éléments de la logique de conception ainsi que ceux du contexte du projet de conception ne sont pas pris en compte. Dans le modèle IBIS, on ne voit pas, par exemple, les compétences et les rôles des participants. Les informations caractérisant le contexte du projet n'y sont également pas représentées.

La résolution de problème est représentée, dans IBIS, à travers des situations statiques de prises de décision. La méthode IBIS ne propose pas un modèle permettant de représenter l'évolution de la prise de décision pour la résolution de problèmes. Ce modèle permettrait de donner une vision plus globale sur la dynamique de la discussion et des prises de décision.

La méthode IBIS a été définie dans le but de fournir un moyen pour organiser d'une façon structurée les discussions des réunions de conception. Cependant, son application pendant les réunions a montré ses limites notamment pour l'identification des questions, des propositions et des arguments en temps réel.

Un outil support gIBIS (Graphical hypertext software tool for building IBIS network) est fourni avec la méthode. L'interface de l'outil gIBIS est formée de quatre fenêtres (Figure 14):

- Fenêtre Graphique, permettant un affichage des éléments de la méthode (Questions, Positions, Arguments,...) en nœuds et liens,
- Liste de nœuds ordonnés suivant la date de création des questions,
- Fenêtre de Commandes générales et
- Fenêtre de Description, permettant l'affichage de la description des nœuds et de leurs attributs.

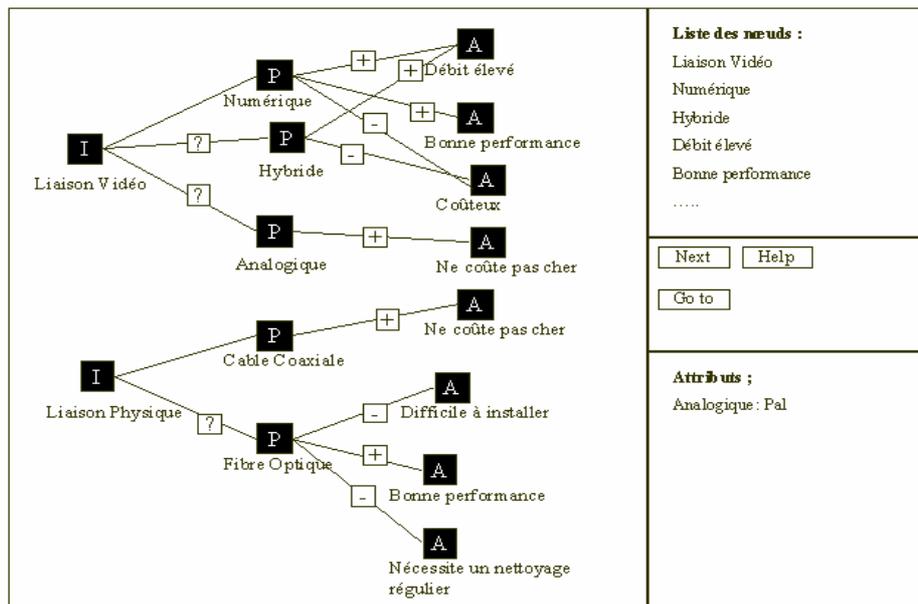


Figure 14 : L'interface principale de gIBIS.

Dans gIBIS L'accès aux informations se fait à travers des vues locales qui sont permises à travers le graphe ainsi qu'une vue globale de tout le processus. De même, des notations comme +, -, ?, etc. désignant des arguments de support, d'objection, des positions non choisies, etc. sont définies comme une marque sur les liens entre les nœuds. Cela peut offrir plus de précision quant à la prise de décision. Des possibilités de recherche d'un nœud sont aussi offertes dans l'outil. Une fenêtre de requête permet de déterminer certaines propriétés du nœud recherché comme : le type du nœud, la date de création, l'auteur, le label, le sujet, certains mots clés, etc. La recherche se fait suivant un simple filtrage entre les propriétés du nœud recherché et les nœuds du graphe. gIBIS est le modèle qui exige le moins d'effort pour formaliser la logique de conception. Cela est dû à deux de ses caractéristiques :

- il a pour vocation de conserver la trace de l'enchaînement des questions qui se sont posées au cours du projet de conception: on documente donc les problèmes traités dans l'ordre où ils l'ont été sans chercher à en faire une synthèse particulière;
- les arguments rapportés dans le document sont la transcription exacte des arguments donnés verbalement ou par écrit par les concepteurs.

Ces avantages se transforment toutefois en inconvénients dès qu'on envisage l'exploitation du document réalisé avec gIBIS. Comme le reconnaissent les auteurs faisant la promotion de ce modèle [Conklin et al., 1991], gIBIS n'est pas destiné à faciliter la réutilisation de la logique de conception, mais plutôt à favoriser une bonne coordination au sein d'un projet. L'enchaînement des questions telles qu'elles se présentent au cours de la conception peut en effet rendre difficile l'accès à certaines informations parce qu'elles conduisent à créer un contexte qui a peu de chance d'être celui qui caractérisera l'exploitation future du document. Par ailleurs, pour comprendre le choix d'une certaine option, il faut lire un ou plusieurs arguments rédigés sous forme de texte libre, ce qui devient vite fastidieux.

II.4.3.2. L'approche QOC

Comme dans IBIS, l'approche QOC «Questions, Options and Criteria» [MacLean et al. 1991] a pour objectif de représenter la logique de conception nommée aussi analyse de l'espace de conception. Elle représente (Figure 15) cette analyse à travers des :

- Questions: les questions et problèmes posés lors d'une conception.
- Options: les différentes réponses données à ces questions.
- Critères: les critères qui permettent de discriminer telle ou telle option.

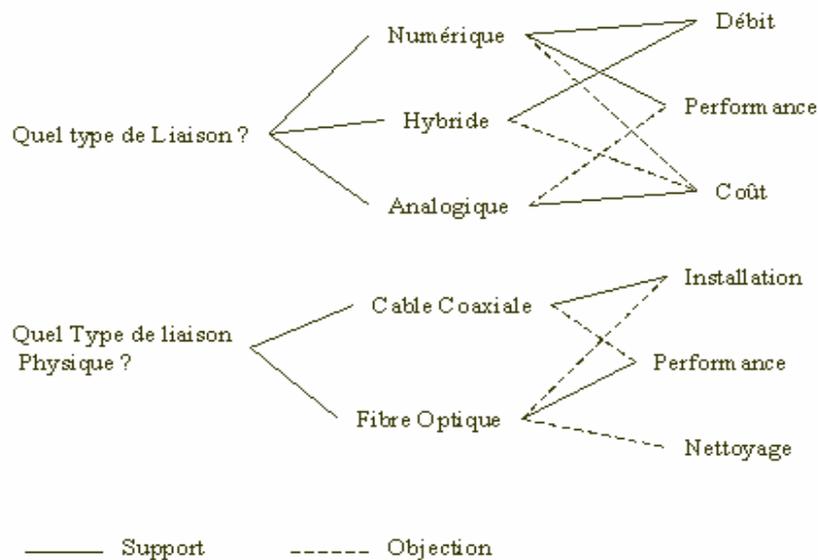


Figure 15 : Représentation de la prise de décision selon QOC.

L'approche QOC représente uniquement l'espace de la prise de décision sous une forme structurée. Par ailleurs, des éléments du contexte dans lequel se déroule la conception ainsi que d'autres éléments de l'espace de la résolution de problème ne sont pas représentés. Notons que dans la méthode QOC, on trouve une représentation des critères d'argumentation. En effet, les auteurs de l'approche considèrent que l'espace de conception peut être représenté par des choix de conception. Ces choix sont structurés comme réponse aux questions évoquées par les problèmes de conception et des critères sont utilisés dans le choix de certaines options comme solution et réponse à une question. Des arguments peuvent justifier les choix d'une option suivant un critère donné.

L'approche QOC ne fournit pas une représentation du processus de conception mais plutôt une analyse de la prise de décision dans la conception. De ce fait, on ne trouve pas dans QOC un modèle représentant la dynamique de la résolution de problèmes.

Comme la méthode IBIS, l'approche QOC a été définie pour aider à l'organisation des réunions de prise de décision. Elle a été, également, utilisée pour garder une trace de la logique de conception [Karsenty, 1994] en procédant par des entretiens, à posteriori, avec le chef de projet et en restituant le contexte des différentes réunions.

Dans QOC il n'y a pas d'accès guidés ou indexés au contenu de la mémoire de prise de décision, ce qui ne permet pas un accès direct et adapté au besoin du ré utilisateur.

Le formalisme QOC est destiné à faciliter non seulement la réutilisation d'une solution, mais aussi la réutilisation d'un raisonnement. La réutilisation d'un raisonnement est centrale dans les situations de formation. Ces deux activités peuvent susciter des questions en partie différentes de la part des concepteurs :

- pour réutiliser une solution, les questions typiques posées par les concepteurs sont : Pourquoi a-t-on besoin de X ? Pourquoi X et pas Y ? À quelles conditions Y pourrait être préféré à X ? Comment a-t-on pris en compte tel problème connu avec une solution de type X ?
- pour réutiliser un raisonnement, les questions qui peuvent être posées sont : Quels autres choix étaient possibles ? Quels sont les critères pertinents pour choisir entre X et Y ? Quelles sont les questions à se poser pour réaliser X ? Comment a-t-on validé tel choix ?

Un document QOC permet de répondre à l'ensemble de ces questions.

II.4.3.3. Le système DRAMA

DRAMA [Brice] est un logiciel défini sur PC. Il permet de garder une trace de la logique de conception d'un projet de conception. DRAMA permet de représenter les buts, les options de solutions ainsi que le choix d'options, suivant un arbre de solution. Une table de critères permet aussi d'illustrer le choix d'une option comme solution à un but donné.

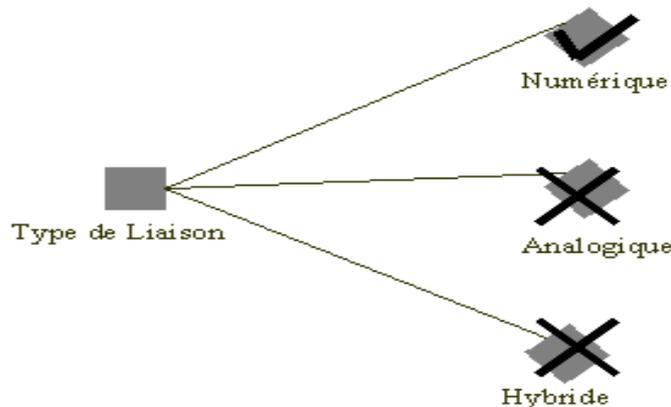


Figure 16 : Arbre de solution.

Comme IBIS et QOC, le système DRAMA représente uniquement l'espace de la prise de décision à travers des arbres de solutions (Figure 16).

DRAMA utilise des représentations statiques (les arbres de solutions). Par conséquent, la dynamique de la résolution des problèmes est omise dans le système.

L'applicabilité du système en temps réel n'est pas évidente car il est difficile de classer, en temps réel, les éléments importants d'une discussion de conception tels que les questions, les arguments et les critères pour les représenter dans des arbres de solutions.

Une interface graphique est fournie dans le logiciel DRAMA. Elle permet de définir aussi bien des arbres de solution que des tables de décision. Elle permet aussi d'afficher différents types de documents: textes, dessins, etc. Des fonctionnalités de recherche sont aussi présentes dans DRAMA.

II.4.3.4. Le système DRCS

DRCS (Design Rationale Capture System) [Klein, 1993] est un système qui permet de représenter la logique de conception dans un projet en ingénierie concurrente. Il est implanté en Common-Lisp et exploite un réseau de stations. Le système permet un travail collaboratif et utilise des liens hypertextes.

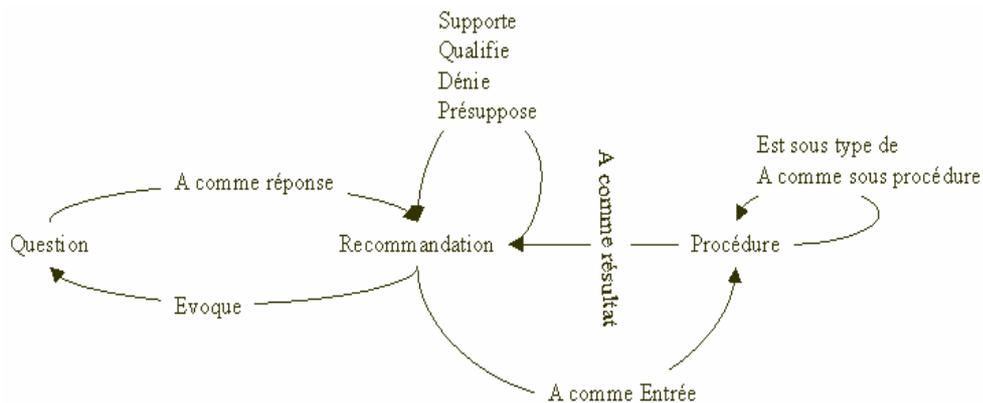


Figure 17 : Modèle d'Argumentation [Klein, 1993].

Le système DRCS offre une représentation plus globale de la logique de conception. En effet, certains éléments du contexte comme l'organisation de l'activité ainsi que l'artefact sont représentés. En outre, le système représente l'espace de la prise de décision à travers quatre modèles : modèle d'argumentation (Figure 17), modèle d'intention, modèle d'évaluation et modèle de versions. En outre, plusieurs éléments du contexte du projet de conception sont également représentés notamment dans le modèle de séquence de tâches, le modèle de synthèse (description d'une tâche) et le modèle de représentation de l'artefact. Le modèle d'évaluation quant à lui, permet de représenter comment les spécifications de l'artefact ont été atteintes.

On constate, ainsi, dans le système DRCS plus d'effort pour une représentation plus exhaustive du processus de conception. Ceci dit, le système ne permet, par contre, de représenter qu'une partie de ce contexte (l'organisation en tâches et la projection des décisions sur l'artefact). D'autres éléments importants tels que les compétences des participants, les outils et techniques utilisés ou les contraintes à considérer ne sont pas représentés.

Dans les modèles du système DRCS, on voit bien l'aspect dynamique de la résolution des problèmes. En effet, le modèle de versions représente la dynamique de la résolution de problème en permettant de montrer les alternatives de conception et celles qui peuvent résoudre un problème rencontré ou trouver une solution à un conflit donné.

Ce modèle permet de faciliter la compréhension de la dynamique de la résolution de problèmes. Par ailleurs, les modèles de DRCS sont plus orientés produit ou artefact et par conséquent on ne voit pas sur ces modèles la dimension collective de la conception à travers, notamment, les participants, les rôles et les compétences ce qui rend la représentation du processus de conception tout de même partielle.

DRCS est appliqué à posteriori, c'est-à-dire après la fin du processus de conception ce qui peut causer une perte de quelques informations importantes dans la compréhension de la logique de conception, notamment lors de la restitution d'une prise de décision qui dépend étroitement du contexte de la conception.

En situation de réutilisation, les différents modèles de DRCS permettent, sans doute, une compréhension meilleure du processus de conception, mais pour ce faire, il est indispensable de parcourir tous les modèles pour chercher une information précise puisqu'on ne dispose pas d'un accès direct et personnalisé aux informations requises.

II.4.3.5. Le formalisme DIPA

Le formalisme DIPA [Lewkowicz et al., 1999] (Données, Interprétations, Propositions, Accord) se base sur une analyse cognitive de la résolution de problèmes pour représenter la logique de conception. C'est dans ce cadre qu'il utilise des modèles de résolution de problèmes définis dans l'ingénierie de connaissances pour structurer une prise de décision.

Dans le modèle DIPA la prise de décision est représentée en trois étapes majeures :

1. Une première étape de description du problème qui permet de recueillir des données, considérées comme des symptômes dans des situations d'analyse ou comme des besoins dans des situations de synthèse;
2. Une deuxième phase d'abstraction qui part des données du problème pour leur trouver une interprétation correspondant à une cause possible dans les situations d'analyse ou à une fonctionnalité de la solution dans les situations de synthèse;
3. Une troisième phase d'implémentation qui part de l'interprétation (cause ou fonctionnalité) et qui permet d'élaborer une proposition qui prendra la forme d'une réparation supprimant la cause du symptôme (analyse) ou d'un moyen répondant à la fonctionnalité exprimée (synthèse).

Dans DIPA on voit plus clairement l'aspect collectif de la résolution de problème ainsi que l'aspect prise de décision. Par contre, des éléments du contexte de la résolution de problèmes sont omis, il s'agit notamment de ceux en rapport avec l'espace de travail tels que les références ou les techniques et outils utilisés ou même ceux relatifs aux contraintes tels que les charges de travail, les coûts ou les calendriers prévisionnels.

Dans le modèle de la résolution coopérative de problèmes, on voit que l'évolution de la résolution de problèmes est nettement représentée ce qui permet de faciliter la compréhension du processus de conception en examinant les différentes étapes mises en avant par le modèle.

Un outil support MEMO-NET [Lewkowicz et al., 1999] est fourni avec le formalisme DIPA. L'outil basé sur une plate-forme distribuée permet aux différents participants d'échanger à travers cette plate-forme. Ils peuvent soumettre une question ou répondre à une question en classant leurs contributions selon des critères prédéfinis. L'outil permet d'organiser la résolution collective de problèmes en temps réel. On constate dans l'outil que les critères d'argumentation sont imposés a priori et il n'y a pas, par conséquent, une possibilité de rajouter d'autres critères jugés fondamentaux par les participants ce qui peut causer un manque de maturité dans les représentations de la résolution de problèmes.

II.4.4. Synthèse comparative des différentes méthodes de formalisation des connaissances

Toutes les méthodes précédentes se distinguent par leur type, leur mode de recueil des connaissances, le type de connaissances manipulées et la façon de représenter ces connaissances.

II.4.4.1. Méthodes et Outils

La Figure 18 présente l'ensemble des méthodes et outils dédiés à la capitalisation des connaissances.

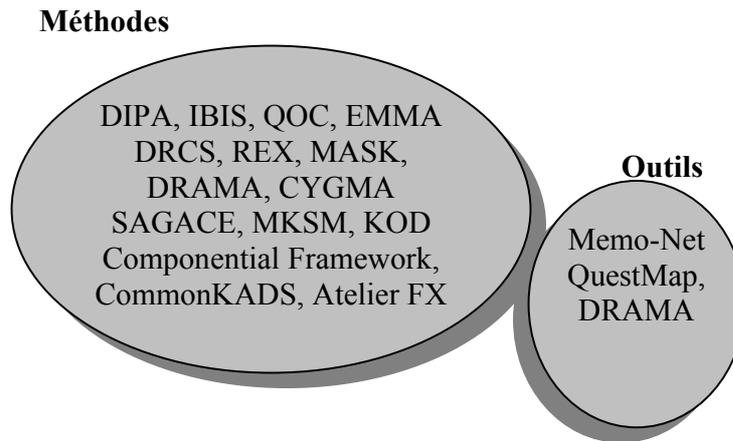


Figure 17 : Méthodes et Outils de recueil des Connaissances.

II.4.4.2. Types de Méthodes

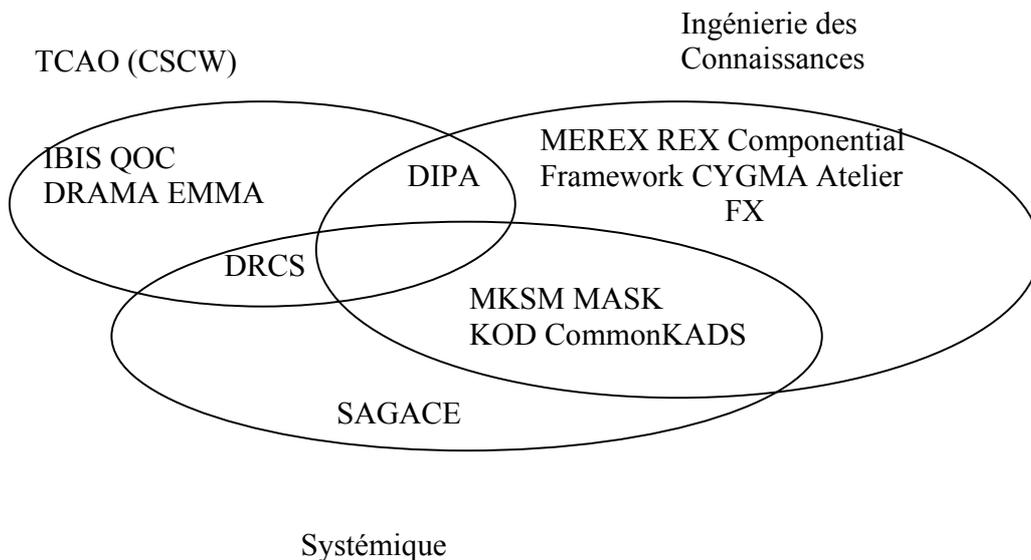


Figure 18 : Domaines d'application des méthodes de Capitalisation.

Les méthodes REX, CYGMA, MEREX sont dédiées à la capitalisation de connaissances. Les méthodes MKSM, Componential Framework, CommonKADS sont adaptées de l'ingénierie des connaissances.

II.4.4.3. Recueil et sources de Connaissances

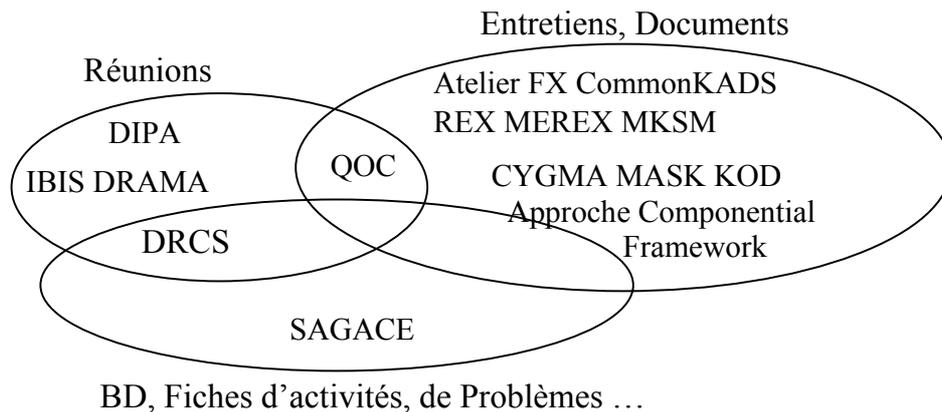


Figure 19 : Techniques de Recueil des Connaissances.

Les méthodes REX, MKSM, Componential Framework, CommonKADS, KOD adoptent le mode d'entretiens et des analyses de documents pour recueillir les connaissances. La méthode CYGMA adopte le mode des entretiens et de l'extraction à partir des documents. L'atelier FX adopte le mode d'observations et d'extraction semi-automatique à partir des documents.

Les méthodes REX, MKSM, CYGMA, Componential Framework, CommonKADS, KOD prennent leur sources de connaissances des experts et des documents. L'atelier FX prend ses sources de connaissance à partir de documents. La méthode MEREX prend ses sources de connaissances à partir des experts.

II.4.4.4. Types de Connaissances manipulées

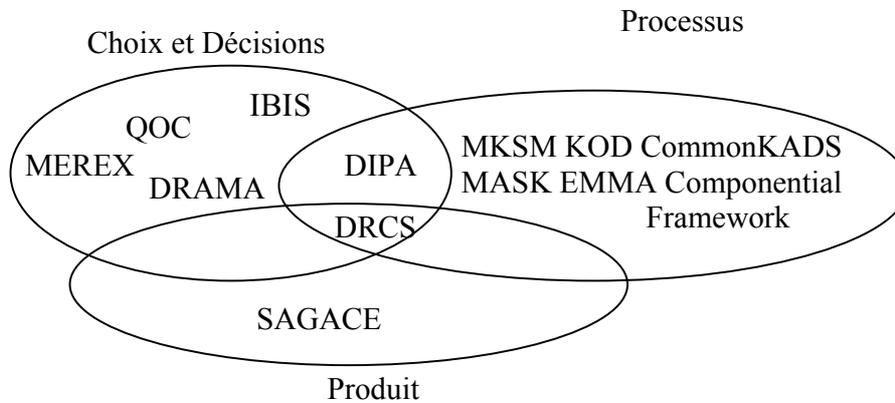


Figure 20 : Types de Connaissances manipulées.

Les méthodes IBIS, QOC, DRAMA sont des méthodes qui manipulent des connaissances dédiées à la représentation de la prise de décision du type Question/Position/Argument ou Critère.

La méthode SAGACE permet de représenter les aspects fonctionnels, organiques et opérationnels comme résultat d'un projet.

Les méthodes MKSM, Componential Framework, CommonKADS et KOD permettent de définir les typologies de connaissances définies (Information, Contexte, Signification, Tâche, Méthode, Information, Inférence, Modèle du domaine, Ontologie). Ce sont des méthodes dédiées à la représentation d'un processus.

II.4.4.5. Représentation des Connaissances

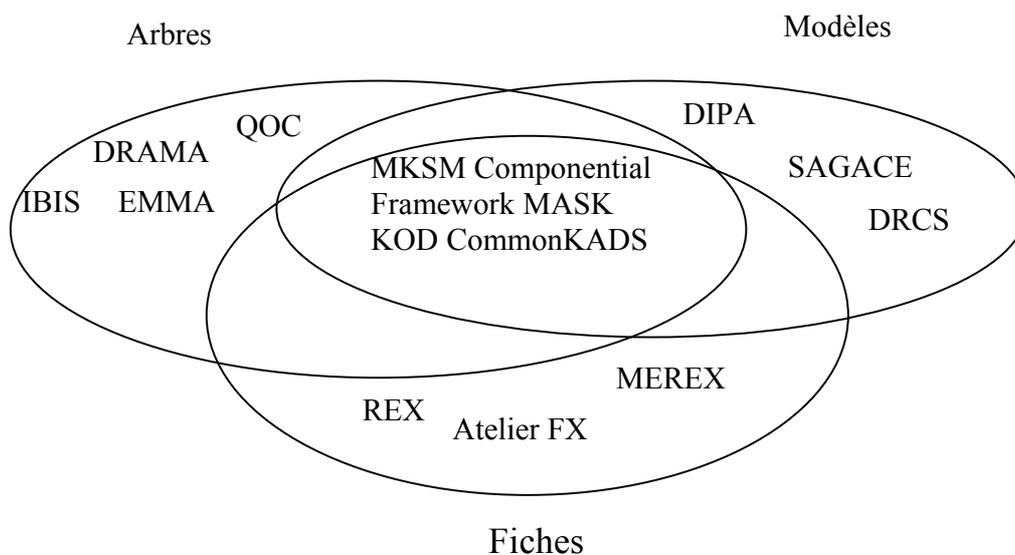


Figure 21 : Types de représentation des Connaissances.

Les méthodes IBIS, QOC, DRAMA représentent les connaissances sous forme d'arbres. La méthode DRCS représente les connaissances sous forme de graphe. Les méthodes MKSM, Componential Framework, CommonKADS et KOD représentent les connaissances sous forme de modèles de connaissances. La méthode REX représente les connaissances sous forme de mémoires d'expériences (ensemble de fiches).

Les approches IBIS, DRCS, DRAMA, EMMA et SAGACE sont munis d'outils informatiques pour la gestion des connaissances à savoir gIBIS pour l'approche IBIS, Système DRCS pour l'approche DRCS, DRAMA pour l'approche DRAMA, EMMA pour l'approche EMMA et l'outil Systémographe pour l'approche SAGACE. De même pour les méthodes REX, MKSM, Atelier FX, Componential Framework, CommonKADS, KOD respectivement les outils sont l'outil REX, l'outil MKSM, l'outil Atelier FX, l'outil KREST, l'outil KADS et enfin l'outil K-Station.

II.4.4.6. Choix d'un outil pour la formalisation de la mémoire de projet :

Ces méthodes sont intéressantes pour construire des livres de connaissances mais sont très peu adaptées pour faire des modèles à des fins « d'usage » dynamique informatique. D'après [Ermine, 1996] un livre de connaissances est un « document contenant les descriptions textuelles et graphiques des modèles de connaissances obtenus après modélisation des connaissances de certains experts de l'entreprise ». On peut décrire avec ces modèles les concepts d'un domaine donné, les connaissances métiers. Les échanges d'informations ne peuvent pas être décrits avec ces modèles. Ces modèles ne proposent pas de modèles produits couplés avec le processus de conception. Les modèles de connaissances, excepté CYGMA, sont génériques et nécessitent une adaptation pour un usage en conception. Ces différentes méthodes sont en grande majorité orientées résolution de problèmes et ne traitent pas simultanément tous les aspects d'un projet (Produit, Processus, Documents, gestion des évolutions et des versions, Ressources etc.). Elles se focalisent par exemple sur les aspects relatifs aux modèles d'activités et Domaine sans aborder les aspects de relation avec par exemple le modèle de produit.

Le langage UML s'est imposé naturellement pour la représentation des connaissances, puisqu'il est actuellement devenu un standard pour la conception orientée objet de systèmes informatique [Muller et al., 2000], standard maintenu par l'Object Management Group. De plus, ce langage et plus précisément ses diagrammes de classes est depuis peu utilisé pour représenter des connaissances sous forme d'ontologies [Cranefield et al., 1999] [Kogut et al., 2002]. D'après [Gruber, 1993] une ontologie est la «spécification explicite d'une conceptualisation». Les diagrammes d'objet peuvent ainsi être utilisés pour formaliser les connaissances sur le système particulier étudié et son modèle, à partir des classes formalisant les connaissances.

Le langage UML se trouve être parfaitement adapté à notre approche, puisqu'il permet à la fois de représenter les concepts d'un domaine au travers des diagrammes de classes et de fournir des spécifications pour le système d'information dédié mémoire de projet que nous cherchons à développer.

La méthode CommonKADS utilise maintenant les conventions graphiques d'UML [Muller, 1997] pour représenter les modèles. Un certain nombre de modèles UML trouvent leur modèle correspondant dans CommonKADS. Exemples : le diagramme d'activité dans UML a comme équivalent le modèle d'organisation dans CommonKADS qui peut aussi être utilisé pour le modèle de connaissance. Le diagramme d'états-transitions dans UML a comme équivalent dans CommonKADS le modèle de communication.

Pourquoi on n'a pas choisi CommonKADS comme outil de modélisation de notre mémoire de projet par rapport à UML :

- difficultés dans la phase d'acquisition des connaissances
- utilisation du CML (Conceptual Modeling Language) [Schreiber et al., 1999] qui est un langage semi-formel
- pas de modèle de conception particulier, (les règles sont moins contraignantes)
- les outils logiciels toujours en développement
- les modèles développés par la méthode CommonKADS sont moins flexibles dans les systèmes autres que les Systèmes à Base de Connaissances et ne peuvent pas remplacer le raisonnement humain, mais rendent les connaissances des experts plus utilisables et permettent aux non-experts de mieux fonctionner.

II.5. Du KM à la Mémoire de projet

II.5.1. Besoins en Mémoire de projet

II.5.1.1. Introduction :

La littérature sur la mémoire de projet est peu abondante. Une large part, issue du domaine de la gestion (gestion d'entreprise et gestion de projet), est d'origine américaine et revêt la forme de descriptions prescriptives. En effet, elle soutient avec force et vigueur que l'entreprise qui veut demeurer compétitive doit se donner les moyens d'apprendre de ses erreurs et de reproduire ses succès dans le domaine de la conduite des projets.

II.5.1.2. Quelques définitions :

[Tourtier, 1995] la définit comme «la définition des activités du projet, son histoire et ses résultats».

[Pomian, 1996] définit la mémoire de projet comme étant «leçons et expériences issues de projets donnés».

Pour [Matta et al., 1999a], « Une mémoire de projet est définie comme étant une mémoire des connaissances et des informations acquises et produites au cours de la réalisation des projets ». Elle distingue la mémoire des caractéristiques de projets et la mémoire du raisonnement de conception. Ainsi les connaissances liées à l'activité de conception tant qu'elles ne sont formalisées et explicitées peuvent être volatiles et par conséquent non réutilisables pas même par leur détenteur après un certain temps. Par contre, si elles sont formalisées et mémorisées, elles deviennent réutilisables [Matta et al., 1999b].

D'après [Karsenty, 2001] la mémoire de projet est définie comme étant « un simple archivage des solutions réalisées dans chaque projet : ces solutions doivent être conservées bien sûr, mais les raisons des choix ayant conduit à ces solutions aussi ».

Quant à [Longueville et al., 2003] il définit la mémoire de projet comme « une représentation explicite des connaissances acquises et produites pendant le projet ».

De l'ensemble de ces définitions se dégage l'utilisation en commun des concepts de projet, processus, produit, conception d'où notre contribution : «la mémoire de projet est la mémoire des éléments d'un produit et de son processus de conception en vue d'une utilisation ultérieure» [Bensta, 2005a].

II.5.1.3. La mémoire de projet : un outil de gestion des projets ou des entreprises

Dans la plupart des cas, la mémoire de projet est abordée dans la littérature sur la gestion de projet [Gulliver, 1989]; [Genest et al., 1990]; [Le Bissonnais, 1992]; [Herniaux, 1993]; [PMI, 1998] [Kerzner, 1998]; [Kleiner et al., 1999]; [Boy et al., 2000]; [Maders, 2000]. Cependant, quelques-unes des publications qui en traitent se situent dans le domaine plus large et diversifié de la mémoire d'entreprise [Girod, 1995]; [Pomian, 1996]; [Davenport et al., 1998]; [Weiser et al., 1998]; [Bekhti, 2003].

Dans la littérature sur la gestion de projet, la mémoire de projet est considérée comme une étape à réaliser en cours de route ou à la fin, étape au cours de laquelle on produit une trace le plus souvent écrite du projet. Cette étape, il ne faudrait pas la confondre avec ce que certains auteurs appellent «le rapport de fin d'affaire». Ce dernier vise en effet à vérifier l'atteinte des objectifs du projet (coûts et échéancier notamment). La mémoire de projet, quant à elle, vise à prendre du recul, comprendre l'origine des succès et des échecs et, plus généralement, effectuer un réel retour d'expérience [Genest et al., 1990].

Dans la littérature sur la mémoire d'entreprise, la mémoire de projet est un volet restreint d'un exercice beaucoup plus large de capitalisation de toute un ensemble d'expériences diversifiées réalisées au sein de l'entreprise par l'ensemble des gestionnaires, voire l'ensemble du personnel [Girod, 1995]; [Pomian, 1996] [Davenport et al., 1998]. En d'autres termes, les projets ne sont qu'un domaine et qu'une occasion parmi d'autres de documenter et de préserver les savoirs et savoir-faire des acteurs de l'entreprise. Selon les auteurs, la mémoire de projet revêt toutefois une importance particulière. D'une part, contrairement aux autres activités de l'entreprise, les projets étant déterminés et ciblés dans le temps, lorsqu'ils sont terminés, les équipes sont dissoutes et les pratiques ne sont plus accessibles. D'autre part, la mémoire de projet est plus répandue que la mémoire d'entreprise : elle existe depuis plus longtemps et est notamment largement développée en conduite de projets informatiques. Bénéficiant d'une tradition plus longue, les organisations se seraient davantage approprié et mettraient plus d'avantages en pratique la mémoire de projet que la mémoire d'entreprise [Pomian, 1996].

II.5.1.4. Les fonctionnalités que doivent assurer une mémoire de projet

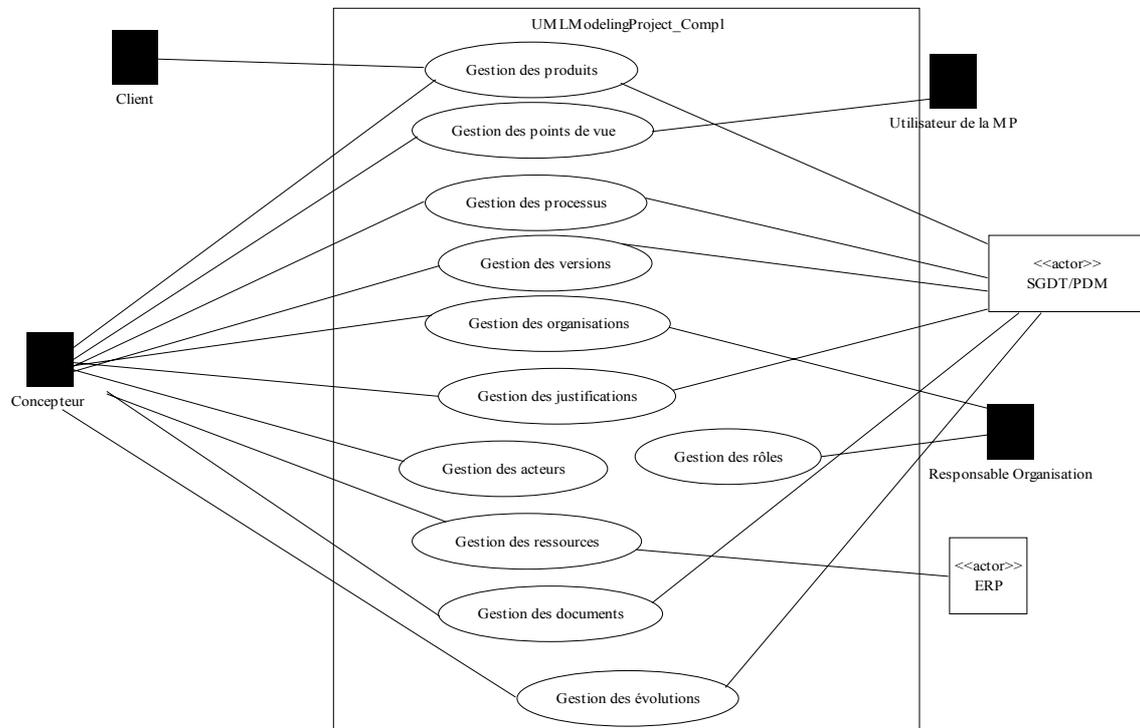


Figure 22 : Illustration des fonctionnalités d’une mémoire de projet par un diagramme de cas d’utilisation.

La Figure 23 présente les fonctionnalités que doit avoir une mémoire de projet et les acteurs qui interagissent avec cette mémoire.

A/ description des cas d’utilisations en relation avec la mémoire de projet

- **Gestion des produits** : tenu en mémoire des produits et de leurs compositions en éléments de produits.
- **Gestion des processus** : l’ensemble des processus et les enchaînements d’activités corrélées.
- **Gestion des versions** : fonction de la mémoire de projet qui permet de conserver et de restituer les versions successives d’un ensemble de données relatifs a un produit, processus, documents, etc.
- **Gestion des organisations** : gérer l’ensemble des organisations de l’entreprise et leurs décompositions en éléments d’organisation
- **Gestion des justifications** : gérer les justifications et les éléments des justifications objet des choix de décision lors des étapes de conception. Ces éléments de justification peuvent être du type document, prototype, croquis, maquette, etc.
- **Gestion des acteurs** : gérer l’ensemble des intervenants dans le projet

- **Gestion ressources** : la gestion des moyens mis en œuvre pour garantir en permanence à l'entreprise une adéquation entre ses ressources et ses besoins en personnel, sur le plan quantitatif comme sur le plan qualitatif.
- **Gestion des documents** : durant la durée de vie de la mémoire de projet des documents lui sont utiles pour vivre d'une part et d'autre part des documents qui sont générés les résultats de la mémoire de projet. Ces documents par exemple peuvent être des éléments utiles pour le début d'un processus.
- **Gestion des évolutions** : la mémoire de projet doit gérer les changements dans la définition de produits. Ce changement peut être demandé (volonté d'optimiser les performances, les coûts, es délais, etc.) ou un changement qui peut résulter d'une non-conformité : « anomalie ».
- **Gestion des points de vue** : les différents points de vue sur les éléments de la mémoire de projet que pourra avoir un utilisateur.
- **Gestion des rôles** : l'attribution des droits d'accès aux ressources.

B/ description des acteurs en relation avec la mémoire de projet

- **acteur client** : il peut contribuer à la préparation du cahier de charge qui décrit explicitement le contenu de la prestation attendue et des éventuelles contraintes concernant les conditions techniques de production, d'exploitation et de qualité des produits, des processus, des organisations, etc. éléments qui servent à la description de la mémoire de projet.
- **acteur concepteur** : il s'occupe de définir les processus et leurs objectifs, les produits de la mémoire de projet, définir les évolutions de produit, de limiter les documents utilisés/générés, les justifications à utiliser, de gérer les versions successives des produits
- **acteur SGDT/PDM** : permet d'avoir les données techniques à propos des objets techniques et des produits dans l'entreprise.
- **acteur responsable d'organisation** : il définit la structure d'organisation. Il peut même attribuer à des acteurs, les rôles de responsables de cellules.
- **acteur technicien** : il contribue par son savoir-faire en tant qu'agent de terrain, à décrire les problèmes rencontrés et à donner ses propositions et solutions
- **acteur ERP** : permet d'avoir des informations à propos des ressources et des objets de l'entreprise.
- **acteur utilisateur** : c'est l'exploitant de la mémoire de projet

II.5.1.5. Les bénéfices attendus de la mémoire de projet

Il ressort de la littérature que l'objectif premier poursuivi par la mémoire de projet est l'amélioration continue de la façon dont on conduit les projets. En marge de cet objectif, d'autres bénéfices pourraient en être attendues.

1) L'amélioration continue des projets

Comme nous l'avons déjà dit, la mémoire de projet est essentiellement un outil de gestion que des entreprises se donnent pour assurer un perfectionnement continu des projets [Genest et al., 1990]; [Gulliver, 1989]; [Meredith et al., 1989]; [Herniaux, 1993]; [Kerzner, 1998]; [PMI,

1998] ;[Kleiner et al., 1999]; [Maders ,2000]. Par là, il s'agit d'éviter de reproduire constamment les mêmes erreurs et faire échec aux conséquences néfastes des pertes d'expériences attribuables à la dissolution des équipes du projet et aux mouvements de personnel en général (mouvements internes, départs à la retraite ou vers d'autres organisations).

L'atteinte de cet objectif est fondée sur le principe de l'apprentissage par la capitalisation des expériences. Tel qu'il est cité dans la littérature sur la gestion de projet, ce mécanisme d'apprentissage est relativement simple et peut être résumé par la boucle « projet à expérience à projet ». [Herniaux, 1993] synthétise assez bien la pensée d'une majorité d'auteurs. Selon lui, il est très utile que l'équipe du projet en démarrage connaisse les expériences antérieures de conduite de projet de l'entreprise. Dès lors, chaque fois qu'un projet y est mené, le mieux est d'enregistrer un certain nombre d'informations à son propos. Ce faisant, un référentiel se crée, dont on se servira pour le perfectionnement continu des projets. Au moment de démarrer un nouveau projet, on consultera ce référentiel pour avoir connaissance des expériences similaires réalisées dans le passé. Le projet sera ensuite exécuté puis, en constituant la mémoire et en l'analysant, on enrichira le référentiel de cette nouvelle expérience.

Ainsi, l'entreprise ne risque plus de perdre son savoir-faire ou de répéter les mêmes erreurs. Au contraire, elle se donne les moyens de transmettre l'expérience acquise au cours des années.

Si elle n'est qu'un volet de la mémoire d'entreprise, la mémoire de projet participerait plus généralement à l'apprentissage organisationnel. Il existe des mécanismes suivant lesquels une organisation apprend. Ces mécanismes font en effet l'objet de propositions théoriques très élaborées dont [Pomian, 1996] donne une vue globale.

2) Etat d'avancement des projets

Une entreprise ne se limiterait pas à améliorer la gestion de projets futurs. Si cette mémoire est réalisée tout au long d'un projet donné (plutôt que seulement à la fin), elle permettrait également de faire le point à différentes étapes de ce projet en train de se faire et de réajuster le tir, au besoin [Meredith et al., 1989].

3) Evolution des connaissances centralisées

Le référentiel constitué pourrait servir à faire évoluer les « connaissances centralisées » de l'organisation, c'est-à-dire ses règles, ses procédures et ses normes formelles [Pomian, 1996]. Comme on le comprend, cela aurait pour effet de soustraire l'enrichissement effectif des projets subséquents aux seules volontés et compétences des individus qui sont ponctuellement appelés à y participer.

4) Risque de construction d'une mémoire de projet

Par ailleurs, la mémoire de projet ne comporterait pas que des avantages mais également des risques. Celui évoqué le plus souvent par les auteurs concerne la mise au jour d' « éléments noirs » que l'histoire officielle n'aurait pas retenus autrement [Pomian, 1996].

5) Amélioration de l'organisation dans une entreprise

La construction d'une mémoire de projet dans une démarche de mémoire d'entreprise, contribuerait à un ensemble plus large de bénéfices. En particulier, elle serait un moyen d'améliorer la gestion d'ensemble de l'organisation (par exemple, en favorisant la mise sur pied de structures souples), notamment ses pratiques de communication et de gestion des ressources humaines : évaluation du rendement, mobilité interne, évaluation des besoins de formation, gestion prévisionnelle de la main-d'œuvre, valorisation des compétences, etc. [Davenport et al., 1998]; [Pomian, 1996].

II.5.1.6. Etat de l'art de la mémoire de projet

Dans ce qui suit nous présentons trois propositions de modélisation de la mémoire de projet qui sont proches de nos objectifs de prise en compte de toutes les dimensions du projet. Ces propositions sont des approches alternatives aux méthodes de définition de mémoires de projet déjà citées. La flexibilité des modèles et la réutilisation de la mémoire produite sont les deux critères privilégiés.

1. Proposition de Yasmina Harani

Dans son travail [Harani, 1997] a défini à l'aide de modèles E/A :

- Un modèle de produit permettant de décrire les différentes facettes du produit à concevoir à différents niveaux d'abstraction. Ce modèle prend en compte un ensemble d'informations qualitatives (à travers l'entité « Description ») et quantitatives (à travers l'entité « Paramètre ») du produit. Les entités du modèle produit sont : Produit, Point de vue, Variable Comportement, Equation Comportement, Description, Constante, Lien, Paramètre, Nœud. Un ensemble de relations sémantiques relie ces entités entre elles,
- Un modèle de processus de conception du produit permettant de décrire le processus de conception du produit à différents niveaux de détail retraçant le pourquoi, le comment et le par qui ou par quoi relatif à chaque étape de la conception. Les entités du modèle processus de conception sont : Processus de conception, Tâche, Opérateur, Etat, Transition, Ressource,
- Un modèle de ressources permettant de représenter l'ensemble des intervenants lors du déroulement de la conception.

Le lien entre le modèle Produit et le modèle Processus de conception a été établi à partir de certains concepts appartenant aussi bien à l'un qu'à l'autre des deux modèles. En effet, chacun des deux modèles fait appel aux concepts du second modèle. L'intégration de ces deux modèles complémentaires permet d'obtenir un modèle cohérent.

La force du modèle de Harani est qu'il présente une forte intégration entre les trois concepts de base : Produit/Processus/Ressource. Cependant les modèles proposés ne prennent pas en compte les aspects documentation. L'apport des approches orientées objet aurait permis d'introduire une classification claire au niveau de ces concepts. De plus, du fait du formalisme utilisé (entité association), les modèles de Yasmina Harani ne tirent pas profit des développements plus récents de la modélisation orientée objets notamment la généralisation/spécialisation. Les modèles ne prennent pas en compte les concepts de justification des choix de décision et d'argumentation des choix de décision lors des étapes de conception.

Les processus sont limités aux processus de conception. Les ressources sont limitées aux ressources humaines. Yasmina Harani a structuré sa proposition sous forme de trois niveaux (méta modèle, spécification et réalisation) avec des liens (de spécification et d'instanciation)

entre ces trois niveaux Le modèle de spécification est complètement défini ; en revanche le niveau méta modèle est peu formalisé.

2. Proposition de Smain Bekhti :

Dans son travail, Smain Bekhti [Bekhti, 2003] s'est appuyé sur les travaux de [Matta, 1999b]. Sa mémoire de projet est décomposée en une partie résolution des problèmes et prise de décision appelée « mémoire de la logique de conception » ainsi que d'une partie représentant les caractéristiques du projet appelée « mémoire de contexte ».

La mémoire de contexte est l'ensemble d'éléments caractérisant le déroulement et l'organisation de projet de conception. Les caractéristiques principales sont les délais (définition d'un calendrier prévisionnel d'un projet), la qualité (des objectifs tracés dans le cahier de charges) et les coûts (déterminés en fonction de la qualité attendue du produit et des délais tracés).

La mémoire de la logique de conception représente les connaissances investies dans la résolution des problèmes et les prises de décisions pendant la réalisation d'un projet.

Un processus de capitalisation des connaissances a été défini. C'est un processus qui permet d'obtenir une trace structurée d'une mémoire de projet contenant le contexte dans lequel se déroule la conception ainsi que la logique de résolution des problèmes.

Smain Bekhti propose un modèle de mémoire de projet formalisé en utilisant les réseaux sémantiques.

C'est un modèle clair et aisé à interpréter. Sa faiblesse est qu'il ne prend pas en compte des aspects évolution, documentation, organisation, justification. Il a beaucoup mis l'accent sur l'aspect logique de conception mais la partie contexte (délais, coût, qualité) n'a pas été développée. Les modèles de Smain Bekhti ne tirent pas parti des développements plus récents de la modélisation orientée objets notamment la généralisation/spécialisation et de ses outils technologiques tel que les SGBDR et les SGBDOO. Les contraintes d'intégrité ne sont pas explicitées.

3. Proposition de Michel Labrousse

Dans son travail, Michel Labrousse [Labrousse, 2004], s'est beaucoup intéressé à l'aspect de l'intégration des concepts Produit/Processus/Ressource comme étant trois notions indispensables à la description d'un système d'entreprise. Il a principalement modélisé ces trois notions.

Différentes possibilités de représentation des processus ont été présentées, parmi lesquelles UML [OMG, 2003], GRAI [Doumeingts, 1984], et IDEF3 [Mayer et al. 1995]. Des exemples de modèles ont été présentés tels que : le modèle FBS (Fonction, Comportement et Structure) [Hu, 2000], le modèle MOKA [Daimler, 2000] et le modèle de Yasmina Harani [Harani, 1997]. Une synthèse de ces différents modèles a été effectuée.

Sur la base de cette synthèse un modèle Produit/Processus/Ressource a été présenté appelé « modèle FBS-PPRE » (Function/Behavior/Structure-Process/Procuct/Resource/External effect).

Le modèle de Michel Labrousse ne fait pas de distinction à la base entre Produit, Processus et Ressource qui sont généralisés en « Objet ». Les concepts de Produit, Processus et Ressources n'apparaissent alors qu'à travers des rôles que jouent les objets les uns par rapport au autres. Les objets Organisation, Justification des choix et Documentation ne sont pas pris en compte.

II.5.1.7. Conclusion

Dans une entreprise la mise en place d'une mémoire de projet ne peut être que bénéfique que ça soit sur un plan organisationnel, humain, technologique, ou économique.

L'aboutissement à une mémoire de projet pourra être un des résultats attendus d'une démarche KM dans une entreprise. Comme cité dans le paragraphe II.6.1.4 cette mémoire de projet doit assurer certaines fonctionnalités à savoir la gestion des produits, la gestion des processus, la gestion des évolutions, la gestion des versions, la gestion des ressources, la gestion des organisations, la gestion des justifications et la gestion des documents.

Une étude de l'ensemble des contributions pour la conception d'une mémoire de projet citées dans le paragraphe II.6.1.6 montre qu'aucune d'elle ne couvre toutes ces fonctionnalités. C'est un des objectifs de ce présent travail qui est d'avoir une mémoire de projet remplissant ces fonctionnalités.

Un autre objectif est celui d'indépendance des modèles du domaine d'application de la mémoire de projet. Pour ce faire, on va doter nos modèles par la caractéristique de généricité. D'un point de vue conceptuel, cela nous amène à définir un niveau fondamental de modélisation à savoir un méta modèle renfermant les concepts de base des modèles. Ce facteur de généricité favorisera la réutilisation des modèles lors de l'élaboration de la conception des éléments de la mémoire de projet. Donc au niveau de la conception il sera plus intéressant de monter dans un niveau afin de ne pas être lié à un domaine de conception particulier d'une part et de pouvoir ajouter facilement un élément de conception « plug-in d'un élément » sans avoir à toucher aux modèles existants.

Une solution possible à notre besoin en mémoire de projet pourra être les SGDT objet du prochain paragraphe ou l'on va essayer de trouver les recoupements possibles de point de vue fonctionnalités entre les SGDT et la mémoire de projet.

Un certain nombre de questions peuvent se poser pour réussir la conception et la mise en œuvre d'une mémoire de projet : Quand constituer la mémoire de projet?, Que faut-il documenter dans une mémoire de projet?, Qui est responsable de recueillir les données sur le projet et comment ?, Comment faire vivre une mémoire de projet?

Nous allons dans ce qui suit essayer de répondre à ces questions.

Quand constituer la mémoire de projet ? : La réponse est lorsqu'une entreprise ne veut plus perdre son capital dans les projets. Le défi est alors d'explicitier ce patrimoine et de le mettre à la disposition des acteurs des nouveaux projets d'où la construction de la mémoire de projet.

Un problème se pose néanmoins dans ce cas : celui de savoir quand et comment faire en sorte de réaliser une évaluation qui permette d'avoir l'heure juste sur les impacts réels d'un projet. Certains facteurs de réussite d'un projet ne peuvent en effet être évalués qu'à moyen ou long terme. Il faut donc attendre pour pouvoir jeter un regard éclairé sur un projet et prendre du recul. Mais si cette attente est trop longue, le facteur « oubli » risque d'entrer en ligne de compte et d'entacher la validité de l'évaluation [Meredith et al., 1989] ; [Kerzner, 1998].

Que faut-il documenter dans une mémoire de projet? : La mémoire de projet consisterait à documenter trois catégories d'information qui, suivant leur « ordre de popularité dans la littérature », sont les suivantes : les résultats du projet, certains événements clés et son processus. Cependant, ces catégories ne doivent pas être considérées comme mutuellement exclusives puisque les propositions des auteurs de la mémoire de projet recourent parfois plus d'une d'entre elles. Dans le domaine de la conception, une mémoire de projet puise ses ressources de plusieurs

types de documents [Matta et al., 1999a] : textes, bases de données, bases de connaissances, maquettes, dessins techniques, plannings, etc.

Qui est responsable de recueillir les données sur le projet et comment ? : Pour cela il existe deux tendances. L'une est attribuée à Pomian [Pomian, 1996] et à Kleiner [Kleiner et al., 1999], où les auteurs de la mémoire de projet sont responsables de préparer les modalités de recueil des données à mettre en œuvre pour documenter un projet. Une autre est attribuée à Lamonde [Lamonde et al., 2001] qui contient deux types de propositions : soit aux membres de l'équipe projet et à son gestionnaire, soit aux membres de l'équipe projet assistés de personnes extérieures.

Comment faire vivre une mémoire de projet ? : Une mémoire de projet ne peut vivre sans le soutien de la direction et une volonté claire de voir la conduite des projets évoluer et s'améliorer continuellement [Pomian, 1996]. Cela acquis, divers points d'action peuvent être exploités pour favoriser au mieux l'appropriation de cet outil par les chargés de projet ou par d'autres acteurs de l'entreprise. Ils touchent principalement la conception même de la mémoire et sa diffusion.

De façon générale, [Pomian, 1996] suggère, pour faire les bons choix, de cerner et de prendre en compte les utilisateurs potentiels de la mémoire, leurs objectifs et leurs besoins. Il serait même souhaitable de constituer un système de suivi permettant d'identifier les difficultés d'utilisation (voire l'absence d'utilisation) de la mémoire de projet par la clientèle ciblée afin de se donner les moyens de l'améliorer.

Pour la diffusion de la mémoire de projet [Kleiner et al., 1999] suggèrent, de faire circuler et de faire étudier l'analyse des récits collectifs par les membres de l'organisation pour favoriser au mieux leur appropriation. [Gulliver, 1989] ; [Pomian, 1996] suggèrent de créer une structure ou de désigner une personne responsable de coordonner toutes les opérations de sauvegarde et de diffusion de la mémoire, voire d'approuver les projets et de faire évoluer les méthodes de conduite de projet.

Une diffusion de la mémoire de projet doit tenir compte des réponses aux questions suivantes : quel est le scénario d'interaction souhaité entre les utilisateurs et la mémoire de projet? Quelles interfaces seront les mieux adaptées à l'environnement de l'activité des utilisateurs ? Quel sera le moyen de diffusion souhaité, selon l'environnement informatique des futurs utilisateurs et des développeurs ? L'intranet de l'entreprise sera-t-il exploité, s'il existe ? Quelle organisation sera mise en place pour la diffusion, sera-t-elle centralisée dans un service spécialisé dans la diffusion ou au contraire distribuée entre certains membres de l'entreprise ? De point de vue de l'entreprise, la diffusion sera-t-elle active (suivant une philosophie push) ou passive (suivant une philosophie pull).

Nous allons maintenant voir l'apport des systèmes de gestion des données techniques et leurs limites.

II.5.2. Les Systèmes de Gestion des Données Techniques

II.5.2.1. Son origine

Le SGDT est un outil qui aide les ingénieurs à gérer les données et les processus de développement de produit. Il conserve, hiérarchise toutes les données et informations nécessaires tout le long du cycle de vie d'un produit (conception, fabrication, maintenance et support).

Les Systèmes de Gestion de Données Techniques (SGDT) sont des SGBD spécifiques. Ils sont essentiels aux entreprises car ils représentent en quelque sorte leur mémoire. Les SGDT sont utilisés pour gérer toutes les informations relatives au produit en cours de développement. Ils ne sont pas en opposition avec d'autres outils, comme les maquettes numériques. Ils sont au contraire un complément indispensable pour conserver la trace de toutes les modifications qui ont été apportées au produit, et donc de toutes les versions du produit. [Carratt et al. 2000] les utilise d'ailleurs en complément des maquettes numériques. Ils permettent également de faciliter l'échange et le partage d'information, puisqu'ils fournissent une base commune à tous les intervenants d'un projet.

Définition : Un certain nombre de définitions existe dans la littérature du concept SGDT. La définition qui nous paraît la plus explicite dans ses termes d'une part, et d'autre part, plus proche de nos besoins est celle de Leila GZARA : « ensemble d'outils logiciels intégrés permettant de consolider et redistribuer l'ensemble des informations de définition du produit, d'en organiser, gérer et contrôler les accès, les modifications, le partage, le groupement, la sécurité, l'approbation des données techniques, créées sous différents formats et d'assurer l'archivage des données techniques dans un environnement hétérogène et distribué » [Gzara, 2000].

II.5.2.2. Fonctionnalités du SGDT

Diverses descriptions des fonctionnalités existantes ou qui devraient exister dans les SGDT sont présentées dans la littérature. Chacune de ces descriptions fait émerger un certain nombre de fonctionnalités que nous récapitulons dans ce qui suit :

- **Structurer les données** : malgré la capacité des entreprises à acquérir ou enregistrer systématiquement les informations sur le produit (sur différents supports), les informations pertinentes détenues par certains attributs ne sont pas souvent mises en avant. Ainsi, pour un composant donné par exemple, il est difficile de retrouver rapidement ses cas d'emploi ou encore les différents documents qui lui sont associés. Les intervenants sur le produit ont souvent des difficultés pour accéder à l'information dont ils ont besoin ce qui réduit l'efficacité de la gestion du produit. Une description des données techniques par les attributs (d'objets et de liens) et une structuration de ces attributs offre plusieurs possibilités de tri et de recherche sur les données.
- **Classer les données** : avec les quantités de données générées, une technique pour classer facilement et rapidement ces données est nécessaire. Grâce à la description des données par les attributs, plusieurs possibilités de classification sont offertes. La classification concerne essentiellement les composants.
- **Visualiser et stocker les données** : certaines informations produit sont définies et archivées dans le SGDT grâce aux classes et attributs associés mais d'autres informations documentant le produit tels que les fichiers des plans CAO, les fichiers de photos numérisés, etc. ne sont que référencés. Ceux-ci doivent être alors visualisés dans le SGDT. Pour certains formats de fichiers, ceci est possible par des « moteurs » de visualisation²⁸ (*Viewer*)

²⁸Ce sont des formats d'interprétation pour la représentation alphanumérique ou graphique qui permettent d'archiver la donnée sous son format d'origine.

disponibles dans les SGDT. Pour d'autres formats de fichiers, ils ne peuvent être visualisés que sur les moteurs qui les ont générés, éventuellement déportés ou importés dans le SGDT en temporaire. Toutefois, certains fichiers dits « Objets longs » ne peuvent être interprétés, ils sont donc gérés comme un tout dans une unité informatique qui peut être propre ou non au SGDT. Le SGDT doit alors assurer l'import - export de ces objets [Randoing, 95].

- **Structurer les données entre elles** : c'est-à-dire définir les liens entre les différents objets définis (classes essentiellement). Cela permet entre autres de structurer le produit en termes de nomenclatures. Ces dernières définissent la décomposition du produit selon différentes vues du produit pour servir divers acteurs de l'entreprise (nomenclature fonctionnelle, nomenclature organique d'études, nomenclature organique de fabrication, nomenclature géométrique, etc.).
- **Gérer l'évolution des données** : un produit est amené à évoluer durant son cycle de vie et les données qui lui sont rattachées sont alors évolutives. Pour garder trace des diverses modifications apportées aux informations produit, différents indices d'évolution sont affectés à ces informations pour distinguer les diverses versions ou révisions. Par ailleurs, pour changer d'indice d'évolution, une information sur le produit suit un processus de modification. Selon l'avancement de ce processus, l'information modifiée acquiert divers statuts (créée, soumise, libérée, validée, etc.).
- **Protéger les données** : par un contrôle des modifications et des accès. Différentes autorisations sont définies pour les acteurs ou les groupes d'acteurs. Une autorisation est un droit pour exécuter certaines fonctions (consulter, créer, modifier) sur les données selon le statut de ces dernières.
- **Distribuer les données** : sur tous les sites, à toutes les fonctions. Les bases de données réparties ne sont pas très courantes, surtout sur des sites distants et hétérogènes. Le transfert de données créées ou modifiées est parfois automatisé grâce à une notion « d'abonnement »²⁹ proposée dans certains SGDT.
- **Discipliner la création et l'évolution des données** : une création ou une modification d'une donnée est généralement référencée par un n° de dossier (ECO³⁰). Dans le cas d'une modification, celle-ci est initialisée par un ECR³¹ (qui a le même n° que l'ECO en général). La création ou la modification doit être validée par un responsable de l'ECO. Les modifications sont alors introduites dans le modèle de données, complété éventuellement par une recopie sur les différents sites et le dossier est fermé (l'ECO peut être alors consulté). Le suivi des différentes étapes du processus (instruction, validation, etc.) est assuré par Workflow jusqu'à la validation définitive.

²⁹ Seules les données qui concernent la liste des produits auxquels chaque site est "abonné" sont transmises [Randoing, 95].

³⁰ Engineering Change Order.

³¹ Engineering Change Request.

- **Structurer l'instruction d'un dossier et sa chronologie (Workflow)** : le workflow consiste en une représentation graphique d'entités et de relations, de type objet. A chaque entité est attachée une classe et des attributs : responsables, autorisés, durée de la tâche, début et fin au plus tôt et au plus tard, description et quantification des ressources, personnes à prévenir, dispatching, validation, etc. Cette gestion des procédures porte sur la nature des tâches, leur enchaînement et la surveillance des délais. Elle nécessite un courrier électronique. Toutefois, il n'y a pas d'ordonnancement des ressources ou de calcul des chemins critiques, ni de lien dynamique avec les données manipulées dans ces procédures.

Le fait par exemple de décider de changer de version d'un document à un moment donné du workflow, ne met pas à jour automatiquement l'objet concerné et ne propage pas l'impact de ce versionnement sur le reste des données. On pourra résumer les différentes fonctionnalités d'un SGDT en utilisant le digramme des cas d'utilisation suivant :

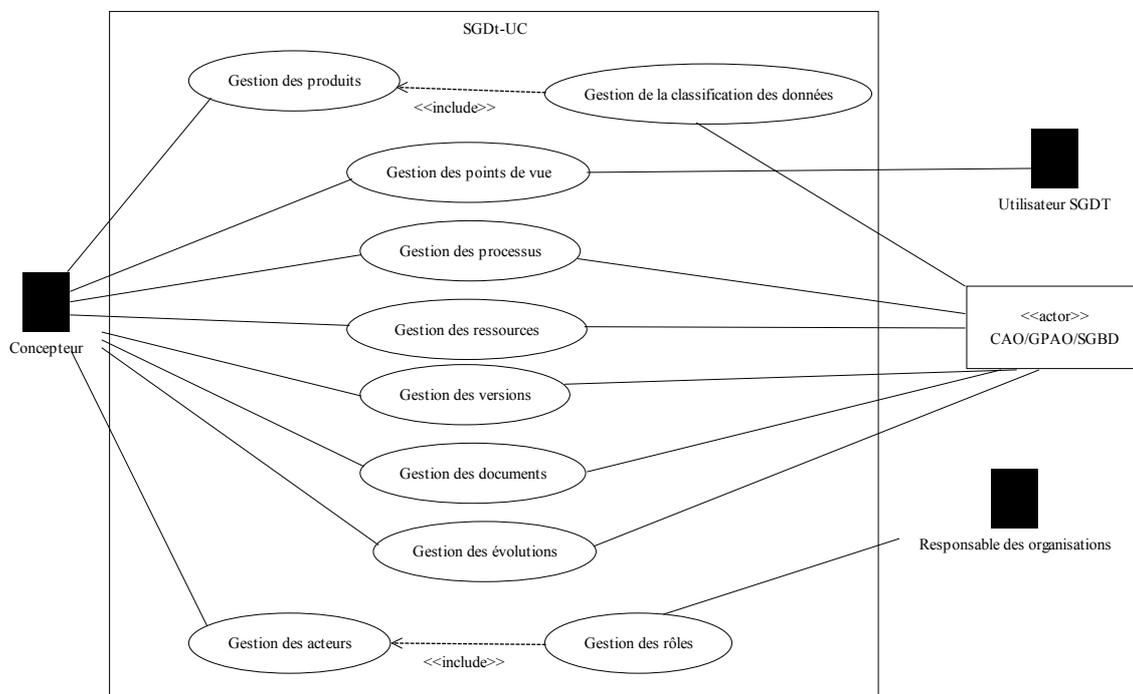


Figure 23 : Illustration des fonctionnalités d'un Système de Gestion des Données Techniques.

La Figure 24 présente les fonctionnalités les systèmes de gestion des données techniques et les acteurs qui interagissent avec tel système.

A/ description des cas d'utilisations en relation avec le SGDT

- **Gestion des produits** : tenue de produits gérés par le SGDT
- **Gestion de la classification des données** : classification des produits tenus par le SGDT

- **Gestion des points de vue** : gestion des différents points de vue sur les éléments du SGDT que pourra avoir un utilisateur.
- **Gestion des processus** : l'ensemble des processus et les enchaînements d'activités corrélées.
- **Gestion des ressources** : la gestion des moyens mis en œuvre au sein de l'entreprise
- **Gestion des versions** : fonction du SGDT qui permet de conserver et de restituer les versions successives d'un ensemble de données relatifs à un produit, processus, documents, etc.
- **Gestion des documents** : gestion des documents utilisés ou produit par le SGDT durant sa durée de vie.
- **Gestion des évolutions** : le SGDT doit gérer les changements dans la définition de produits. Ce changement peut être demandé (volonté d'optimiser les performances, les coûts, les délais, etc.) ou un changement qui peut résulter d'une non-conformité : « anomalie ».
- **Gestion des acteurs** : gérer l'ensemble des intervenants utilisant le SGDT

B/ description des acteurs en relation avec le SGDT

- **Concepteur** : il s'occupe de définir les processus et leurs objectifs, les produits à gérer par le SGDT, définir les évolutions possibles du produit, de limiter les documents utilisés/générés, de gérer les versions successives des produits.
- **CAO/GPAO/SGBD** : se sont des clients qui peuvent s'échanger de l'information avec le SGDT afin de gérer les processus, les produits, les versions etc.
- **Responsable des organisations** : il définit la structure d'organisation. Il peut même attribuer à des acteurs, les rôles de responsables de cellules.
- **Utilisateur SGDT** : il peut contribuer à la définition des différents points de vue sur les produits gérés par le SGDT.

II.5.2.3. Les Avantages d'utilisation des SGDT

A/ Cycle de développement réduit : Réduire ce cycle de développement est le premier avantage des outils SGDT. Ils permettent :

- D'accélérer la réalisation des tâches en rendant les données disponibles instantanément,
- D'organiser et automatiser les tâches et
- D'assurer d'avoir toujours la dernière version disponible des données.

B/ Productivité de la conception améliorée : Les ingénieurs de conception passent trop de temps à obtenir de l'information. Les SGDT suppriment ces temps morts et ces ingénieurs passent plus de temps sur la conception proprement dite. D'autre part, l'utilisation de SGDT permet d'améliorer la réutilisation pour chaque nouveau produit. Ceci avec l'aide des fonctions de classification qui permettent une standardisation des produits. Le système de classification permet de détecter les réutilisations possibles garantissant ainsi la réutilisation.

C/ Intégrité des données : Par la centralisation de l'accès aux données (les données peuvent être diffusées mais on y accède toujours à travers les logiciels de SGDT), l'intégrité des données est assurée.

D/ Meilleure gestion des modifications : Les versions et modifications sont signées et datées. La peur de perdre les versions précédentes lors de nouvelles modifications disparaît. Les remontées de faits techniques ou demandes de modifications sont dirigées automatiquement vers les bonnes personnes, garantissant leur prise en compte par les concepteurs.

E/ Un pas de plus vers la qualité totale (Total Quality Management) : La rigueur et les procédures de contrôles sont facilitées par l'utilisation de logiciel de gestion des données techniques. Cela permet aux entreprises de tendre facilement vers la qualité totale et d'obtenir des certifications importantes telles que l'ISO9000 [ISO, 1997].

II.5.2.4. Les limites d'un SGDT

Un grand nombre de fonctionnalités en commun sont prévues aussi bien par les SGDT que la mémoire de projet tels que la gestion des produits, des processus, des évolutions, des versions, etc. C'est pour cela qu'on dit que les SGDT sont couplés à la mémoire de projet.

La structuration des données et des produits proposés par les SGDT est statique. Ainsi dans le triple perspectif Produit-Projet-Processus les aspects dynamiques du Projet ne sont pas pris en compte. Le module «Gestion des flux de travaux», ou « workflow » inclus dans les SGDT et certaines GPAO, s'apparente plus à une gestion des procédures qu'à une véritable Gestion de Projet, de type PERT avec calcul des chemins critiques, ou du type GPAO de Process ou modélisation. Toutefois une partie importante des données stockées en SGDT est exploitée dans une GPAO très riche en transaction : l'interfaçage entre les deux progiciels est très poussé.

Les SGDT sont assez rigides au niveau de la formalisation de l'information et du type d'informations qu'ils sont capables de gérer à ce jour. En effet les SGDT, à l'heure actuelle, ne font que gérer de l'information dépourvue de toute sémantique : les SGDT sont basés essentiellement dans leur implémentation sur un module de gestion de configurations. Ce module ne fait que gérer des versions de systèmes technologiques, mais ne permet pas de gérer les raisons qui ont conduit à ces solutions, par contre dans une mémoire de projet la fonctionnalité « Justification des choix de décisions » lors des étapes de conception d'un produit est un moyen de gérer les raisons qui ont conduits aux choix des solutions et par conséquent à la gestion des versions. Ces choix qui sont archivés dans la mémoire de projet se basent sur des arguments, des idées, des propositions etc. De telle mémoire de projet ont pour intérêt de favoriser la réutilisation. Elles peuvent favoriser la coordination dans un projet en explicitant les perspectives de chaque acteur du projet [Conklin et al., 1991] et faciliter l'introduction de nouveaux membres

dans un projet ou dans un bureau d'étude. Elles peuvent ainsi avoir une fonction de formation [van Aalst et al., 1995].

Dans un SGDT les documents sont gérés dans leur globalité, comme une enveloppe, mais pas dans leur contenu, tandis que dans une mémoire de projet une telle fonction est tenue en compte qui a pour objectif, entre autres, de gérer le contenu d'un document. Leur accès, leur représentation et leur distribution dans l'organisation, ne peut se faire ainsi que de manière très rigide et prédéterminée. Les modèles de processus, qui s'exécutent en dynamique permettent de décrire les échanges de données, les signatures à fournir et la distribution de l'information. Ils ne représentent pas ce qui se passe avec l'information. Ils ne gèrent pas des historiques au sens des activités avec leurs liens avec le produit. Les SGDT ne gèrent par conséquent que des historiques de versions de produit.

II.5.2.5. Conclusion

Dans le paragraphe qui précède on vient de voir que les SGDT sont d'un grand apport pour les mémoires de projets à savoir la gestion des produits, des processus, des acteurs etc. mais ne peuvent pas les remplacer. Principalement la gestion des justifications n'est pas prévue par les SGDT, fonction importante prévue par la mémoire de projet afin de déterminer les choix, les arguments, les idées, les documents etc. qui ont été pris en compte lors des étapes de conception d'un produit idée d'apprendre, de justifier et de minimiser les erreurs du passé.

Le module de gestion des configurations permet de gérer des versions de systèmes technologiques, mais ne permet pas de gérer les raisons qui ont conduit à ces solutions.

Ces raisons sont difficiles à comprendre : se sont les décisions finales ou intermédiaires des concepteurs d'un produit pendant chaque étape du processus de conception dans un projet.

En effet, pour pouvoir comprendre l'évolution d'un projet ainsi que les décisions prises pendant son déroulement, le concepteur, en situation de réutilisation, a besoin d'examiner les informations relatives à des problèmes rencontrés dans des expériences proches, de connaître et comprendre les solutions envisagées ainsi que les options rejetées, leurs justifications et les décisions prises à la fin de chaque phase. Ainsi comprendre le contexte de la résolution des problèmes à autant d'importance que la résolution elle-même car le concepteur a besoin aussi de situer ces problèmes dans leur environnement pour pouvoir comprendre les raisons derrière l'acceptation ou le rejet de telle ou telle solution. Il est vrai que pendant le processus de conception, des traces de réalisations intermédiaires peuvent être gardées telles que les croquis, les dessins, les cahiers des charges, les lettres, les comptes-rendus, les emails, les échantillons, les maquettes, etc. Ces traces forment une partie du capital acquis pendant la réalisation du projet. Une autre partie de ce capital naît aussi de l'apprentissage des stratégies déployées pour résoudre les problèmes et surmonter les obstacles.

C'est dans le but d'un besoin de tracer les processus de conception d'une mémoire de projet que le prochain paragraphe se situe. Dans ce paragraphe on va présenter quelques définitions du concept de traçabilité, les domaines d'application de la traçabilité et la traçabilité dans le domaine de conception qui un est des objectifs de la mémoire de projet.

II.5.3. Traçabilité

II.5.3.1. Quelques Définitions

On ne trouvera pas la définition de ce terme dans un dictionnaire français car c'est un néologisme qui nous vient de l'anglais « trace ability ». Dans l'absence d'un consensus pour une définition du terme de traçabilité, nous donnons dans ce qui suit les deux définitions qu'on avait trouvé dans la littérature et qui s'approchent le plus de nos besoins destinées à fixer les idées.

Selon E.W. Bernard cité par [Bézivin, 1996], la traçabilité se définit comme « le degré de facilité avec lequel il est possible de suivre un concept, une idée ou un autre élément vers un point situé en aval ou en amont dans le même processus ».

D'après [Zamfiroiu, 1998], la traçabilité est « la capacité d'un système à fournir des liens entre les versions de constituants reflétant le flux de données et de contrôle issu des opérations qui leur sont appliquées ».

Notre contribution : Nous définissons la traçabilité comme suit « La traçabilité dans un processus de conception est la capacité de se projeter dans l'histoire en amont ou en val afin de retrouver l'état d'une justification et les liens que peut avoir cette justification avec ses différentes valeurs dans le temps ».

II.5.3.2. La traçabilité dans le domaine de conception

Dans le cadre d'un processus de conception [Bekhti, 2003] a souligné l'importance de la traçabilité dans les processus de conception. Ainsi les participants à un projet choisissent une solution parmi plusieurs en se basant sur un raisonnement collectif qui prend en compte tous les éléments influençant directement ou indirectement le processus de conception. Pendant ce processus de conception des traces de réalisations intermédiaires peuvent être gardées telles que les croquis, les dessins, les cahiers des charges, les lettres, les comptes-rendus, les emails, les échantillons, les maquettes, etc. Ces traces forment une partie du capital acquis pendant la réalisation du projet. Une autre partie de ce capital naît aussi de l'apprentissage des stratégies déployées pour résoudre les problèmes et surmonter les obstacles.

II.5.3.3. Les Domaines concernés par la traçabilité

Malgré son importance, la traçabilité du processus de conception n'est pas considérée véritablement comme un domaine à part entière de l'informatique. Par conséquent, le sujet est abordé à des degrés différents par des domaines bien établis. D'après [Zamfiroiu, 1998] les domaines recensés autour de la traçabilité sont présentés dans la figure qui suit :

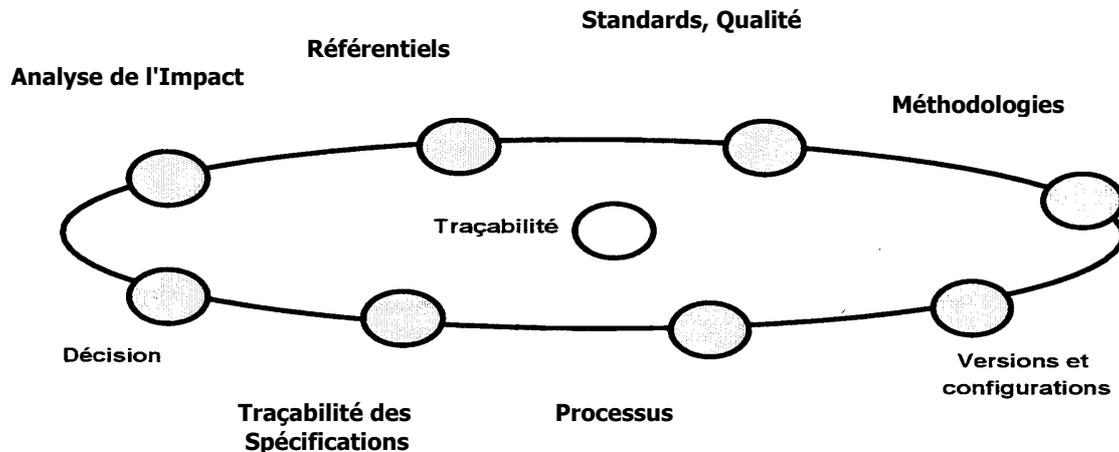


Figure 24 : Les domaines concernés par la traçabilité [Zamfiroiu, 1998].

A/ les standards : les standards de qualité ISO9000 [ISO, 1987] et plus précisément ISO9001 et ISO9000-3 [ISO, 1991], [CMM, 1991] préconisent la traçabilité comme une qualité importante des projets. Leur influence industrielle est telle que les éditeurs d'ateliers de génie logiciel ne peuvent l'ignorer. Leur objectif est la qualité des systèmes. Ainsi l'approche définie par le standard ISO9000 et plus spécialement par ISO9001 et ISO9000-3, peut être résumée sous la forme suivante [McDermid, 1991] ; [Caldeira, 1993]:

- « dits ce que tu fais » : ceci implique l'existence d'une méthodologie compréhensible, couvrant toutes les phases du cycle de vie, y compris la maintenance ;
- « fais ce que tu dits » : désignation des ressources humaines et assignation des tâches en conformité avec le processus décrit par la méthodologie ;
- « enregistre ce que tu as fait » : enregistrement des traces de l'exécution des opérations effectuées ;
- « revois tes enregistrements et réagis » : vérification du respect de la méthodologie par les opérations effectuées.

Pour le domaine de l'informatique et des standards, trois référentiels de bonnes pratiques sont actuellement à la mode :

- COBIT (Control Objectives for Business and Related Technology): [Gédalge, 2004], [Georgel, 2005].
- ITIL (Information Technology Infrastructure Library) [Dumont, 2005], [Georgel, 2005].
- CMMI (Capability Maturity Model Integration) [Basque, 2004].

Outre ces trois référentiels, il en existe d'autres, plus au moins spécifiques ou plus moins en vogue:

- PMI (Project Management Institute), [Garel, 2003].
- Prince2 (PRjects INControlled Environments), [Bentley, 2001].
- Spice (ISO 15504), [Carlier, 2005].
- ISO 17799, [Carlier, 2005].
- ISO 9000 [Carlier, 2005].

B/ les méthodologies : le domaine des méthodologies de conception souligne l'importance de la traçabilité. Les méthodologies imposent les types de constituants et les opérations qui leurs sont applicables. Certaines démarches comme les cas d'utilisation facilitent le passage entre les modèles, d'autres, au contraire, proposent des découplages entre les constituants. Ceci réduit dans certaines situations, considérées fréquentes, l'ampleur des conséquences des changements, mais ne fournit pas de solution pour la traçabilité.

C/ la gestion des configurations logicielles (SCM) : la gestion de configuration logicielle, ou SCM (Software Configuration Management), assure le stockage et la manipulation des versions et des configurations. L'objectif principal de ce domaine est le contrôle des systèmes. Les principaux concepts sont la version et la configuration. Les fonctionnalités de base de tout système de gestion de configurations logicielles comprennent le stockage des versions des constituants et le regroupement de ces versions en configurations dites « cohérentes ».

Les fonctionnalités d'un SCM sont regroupées par [Dart, 1993] en huit grands groupes, comme suit : travail coopératif (Team), construction, structure, composants, processus, contrôle, comptages et audit. La grande difficulté est de couvrir correctement tous ces aspects de façon concomitante et notamment d'assurer la traçabilité en même temps que les fonctionnalités suivantes :

- analyse de l'impact des changements,
- gestion de la cohérence,
- propagation des changements,
- gestion des versions et des configurations et
- utilisation d'un référentiel.

D/ Modélisation du processus : le domaine de la modélisation du processus vise la définition des séquences d'opérations à entreprendre sous la forme de modèles de processus, en vue de l'obtention de programmes de qualité. Le résultat est un modèle de cycle de vie pour le projet entier. Parmi les modèles de cycle de vie les plus connus, nous retrouvons le modèle en cascade, le cycle en V ou encore le modèle en spirale de Boehm [Pressman, 1992] qui illustre bien la dimension itérative et incrémentale du processus.

E/ Traçabilité des spécifications (Requirements Traceability) : le domaine de la traçabilité des spécifications, ou Requirements Traceability, a comme principal objectif de capter le discours des utilisateurs et de suivre les spécifications tout au long du cycle de vie. Ce domaine est confronté à la formalisation des spécifications émises au départ en langage naturel et transformées successivement en représentations plus conceptuelles. Le positionnement des travaux existants vis-à-vis de la traçabilité se justifie par le positionnement spécifique du domaine dans le début du cycle de vie.

F/ Processus orienté décision : l'objectif du domaine de la gestion des processus orientée décision est de canaliser et de capter les décisions des concepteurs pendant le déroulement du processus de construction. Ce domaine se trouve en continuité avec la traçabilité des spécifications. Pour offrir la traçabilité de façon convenable, la modélisation de la décision doit pouvoir fournir les services suivants :

- intégration des décisions avec les opérations et les constituants de granularité uniforme dans le temps et compatible entre opérations et constituants (accord de granularité),
- détection des flux de données indépendamment des liens statiques ou de versionnement entre les constituants et
- séparation des aspects syntaxiques et sémantiques en ce qui concerne les décisions. Le flux de décision doit être canalisé de façon non ambiguë dans le référentiel, de façon à éviter les dépendances implicites et non traçables.

G/ Analyse de l'impact : l'objectif principal du domaine de l'analyse de l'impact est l'identification des conséquences potentielles d'un changement.

Bohner et Arnold [Bohner et al., 1996] identifient deux types distincts de liens entre les constituants d'un projet, au long desquels l'impact se propage :

- les liens dits « de dépendance », établis à partir des relations statiques définies dans le modèle structurel de la méthodologie ; ces liens concernent essentiellement les constituants appartenant à la même phase du cycle de vie et concernent surtout la phase d'implémentation. Ainsi, la définition syntaxique du langage de programmation sert de modèle statique de méthodologie.
- les liens de traçabilité établis entre les constituants situés généralement dans des phases voisines, selon les opérations de transformation appliquées pendant le déroulement du projet.

Ainsi la traçabilité est donc considérée comme une branche de l'analyse de l'impact. Dans ce domaine, on distingue deux sortes de conséquences de l'impact : les effets de bord et les effets d'onde.

H/ Les référentiels : les référentiels proposent un support centralisé pour le stockage et la gestion des constituants. Les référentiels sont gérés par des SGBD et sont souvent exploités par des environnements de conception et développement, nommés aussi Ateliers de Génie Logiciel, ou des outils CASE.

L'augmentation du nombre d'outils nécessaires à la gestion des projets et leur complémentarité a mené à une demande d'intégration des outils dans des environnements de génie logiciel, I-CASE.

Selon [Simon, 1994], cette intégration doit se réaliser sur les cinq plans suivants :

- données : partage de données entre les outils ;
- contrôle : échange d'informations de contrôle ; demande et fourniture de services ;
- présentation : interface utilisateur commune ;
- processus : intégration dans une même méthodologie ;
- plate-forme : système d'exploitation commun ou utilisation des services à travers l'interopérabilité.

D'après [Zamfiroiu, 1998] il existe deux exigences requises pour un système de traçabilité envers le référentiel :

- la nécessité d'un environnement intégré réunissant le partage des constituants entre les différents outils. La spécification du volet statique de la méthodologie correspond avec la définition du schéma de la base de données.
- la possibilité d'un suivi détaillé des actions entreprises par les différents outils. Ce dernier objectif implique des mécanismes de prise en compte de spécifications déclaratives et procédurales des transformations comprises dans le modèle dynamique de la méthodologie.

II.5.3.4. Conclusion

La notion de traçabilité en qualité est définie par la norme ISO 8402 comme l'« aptitude à retrouver l'historique, l'utilisation ou la localisation d'une entité au moyen d'identifications enregistrées ». L'enjeu pour les entreprises est de répondre aux exigences de traçabilité sans engorger l'organisation d'une série d'inscriptions, de formulaires etc. Cela pose le problème du sens de l'information, de l'exploitation possible des traces pour l'activité en cours et future. Les principes des démarches qualité actuelles n'abordent pas ce problème et laissent l'entreprise seule face à cet enjeu crucial : comment faire valoir les possibilités d'intelligence organisationnelle qu'offrirait la mise en cohérence des traces ?

Les limites de la traçabilité dans une démarche qualité classique nous conduisent à élargir et enrichir l'acception du terme « traçabilité », en mettant en valeur la composante dynamique de la réutilisation des traces de l'activité et l'exploitation méthodique et réflexive de ce qui est recueilli [Cochoy et al., 1998] pour tendre vers « une organisation apprenante, en plus de qualifiante, dont le but est autant de se concentrer vers les processus qui favorisent le mode collectif et continu de construction des connaissances que la connaissance en tant que produit » [Prax, 1997].

Ces objectifs sont extrêmement larges et englobent des préoccupations très différentes, bien que toutes pertinentes pour une entreprise, qui sont examinées dans des champs de recherche distincts : le champ de la mémoire organisationnelle [Sauvagnac et al., 2000], et le champ de la logique de conception [Conklin et al., 1991], [Karsenty, 2001]. Nos préoccupations d'étude se situent dans ce dernier champ de recherche : il nous semble que la maîtrise d'ouvrage bénéficierait d'enrichir ses actuelles pratiques de traçabilité en les étendant au suivi et au traçage des décisions prises par les partenaires du projet de conception, non pas tant pour conserver la logique de construction des décisions techniques de conception que pour tracer les réseaux de contraintes qui naissent de ces décisions.

Notre approche de mise en œuvre de la mémoire de projet permettra non pas un archivage statique des données de conception mais plutôt une formalisation et une structuration de ces données dans un langage compréhensible par le concepteur, qu'elles soient stockées dans des bases de données classiques (de type SGBDR³²) ou bien dans des bases de données plus évoluées (SGBDOO³³). Dans ce cas, il nous faudra établir des liens entre l'outil que nous proposons (qui est de haut niveau) et les données techniques (qui sont de bas niveau). Pour cela, il faudrait intégrer notre système dans des outils de type middleware (tel que ceux proposés par Sherpa³⁴). Nous souhaitons ainsi pouvoir pallier au fait que la plupart des SGDT se focalisent principalement, pour ne pas dire exclusivement, sur les données relatives aux produits au détriment de celles traduisant leur processus de conception [Dickey, 1997]. L'apport de notre travail se situera alors plus particulièrement dans la proposition d'une formalisation du modèle de produit et du modèle de processus de conception (ce dernier n'étant pas à confondre avec le processus de fabrication plus souvent utilisé dans les SGDT).

³² SGBDR : Systèmes de Gestion de Bases de Données Relationnelles

³³ SGBDOO : Systèmes de Gestion de Bases de Données Orientées Objets

³⁴ Sherpa : Sherpa Corporation : Société américaine spécialisée dans les outils SGDT

II.6. Conclusion

Nous avons exposé dans ce chapitre un état de l'art du KM. Cet état de l'art comprenait les différents courants du KM, les dimensions du KM, les systèmes de gestion des connaissances, les obstacles et les enjeux qui peuvent engendrer une démarche de mise en place du KM dans une entreprise. Dans une telle démarche la mise en place d'une mémoire de projet ne sera que bénéfique soit sur un plan humain, organisationnel, technique et économique objectifs de capitaliser et préserver les connaissances dans le domaine des projets menés par une entreprise. Un outil de formalisation des connaissances dans une mémoire de projet est nécessaire.

Pour ce faire, nous avons présenté dans ce chapitre un état de l'art des méthodes de formalisation de la connaissance et on a démontré qu'aucune de ces méthodes ne couvrait nos besoins. Pour ce faire, on a choisi UML comme outil de structuration de la connaissance, car il a fait ses preuves dans beaucoup de domaines de structuration des connaissances.

Ainsi les différents modèles de la mémoire de projet, un des objectifs du présent travail, seront formalisés en utilisant le langage UML.

Cette mémoire de projet doit remplir un certain nombre de fonctionnalités, à savoir, la gestion des produits, des processus, des versions, des évolutions, des documents, des acteurs etc.

Les travaux antérieurs relatifs à l'état de l'art de la mémoire de projet montrent que ces travaux présentent quelques insuffisances. Pour pallier à ses insuffisances, on a choisit la solution des SGDT pour en construire une mémoire de projet. Cette solution aussi ne satisfait pas nos besoins du faite que les SGDT ne couvrent pas toutes les fonctionnalités que doit avoir une mémoire de projet. Les processus gérés dans de telle solution ne sont que des processus du type fabrication. Par contre dans le cas de la mémoire de projet on s'intéresse au processus du type conception.

Dans de tels processus la traçabilité est très importante du faite que les participants dans un projet choisissent une solution parmi plusieurs. Ces solutions s'appuyant sur des éléments de justification tels que les croquis, les dessins, les cahiers des charges, les lettres, les comptes-rendus, les emails, les échantillons, les maquettes, etc. Ce sont les traces du processus de conception de réalisations intermédiaires qui sont archivés. Entre autres, ces traces forment une partie du capital de l'entreprise pendant la réalisation du projet.

Enfin, les modèles de la mémoire de projet seront dotés d'une caractéristique de généricité dans un but d'indépendance des domaines d'application de la mémoire de projet. Pour ce faire et, dans notre démarche de construction des modèles nous allons remonter d'un niveau d'abstraction dans la conception des modèles afin de pouvoir garantir cette caractéristique de généricité. On veut avoir des modèles qui soient les plus génériques possibles un des objectifs de ce présent travail.

Nous allons dans le prochain chapitre présenter les différents modèles constituant la mémoire de projet ainsi que le cadre général ou le référentiel de construction de la mémoire de projet. Chacun de ces modèles sera présenté dans un diagramme de classe suivi d'un digramme d'objet.

CHAPITRE III : MODELES ET STRUCTURES PROPOSES POUR LA MODELISATION DE LA MEMOIRE DE PROJET

III.1. Introduction

L'objectif de ce chapitre est de proposer l'architecture générale du système d'information dédié à la mémoire de projet ainsi que les différents modèles qui la constitue. Ces modèles ont été développés en utilisant le formalisme UML [OMG, 2005]. Ils constituent la base de la mémoire de projet. Pour chacun de ces modèles nous présenterons le diagramme de classes associé et un ou plusieurs exemples en utilisant le diagramme d'objet correspondant. Nous indiquerons les points forts des modèles de représentation que nous avons adoptés, d'une part, pour la modélisation d'un projet de conception, et d'autre part, pour la réutilisation de la mémoire construite.

Un des objectifs de la construction et d'exploitation d'une mémoire de projet est sa réutilisation. Les utilisateurs de la mémoire de projet souhaitent comprendre la logique de conception et le contexte des projets archivés. Pour ce faire ils vont essayer de comprendre ces éléments, leur contexte et leur logique de conception afin de les réutiliser dans d'autres projets. L'entreprise peut ainsi gagner en productivité, mais également partager le savoir-faire de conception entre les acteurs dans la mesure où la mémoire est partagée. Pour atteindre l'objectif de réutilisation, l'accès aux éléments archivés doit être facile. Les utilisateurs doivent atteindre n'importe quel élément dans la mémoire de projet sans être obligés de parcourir toute la structure de la mémoire. Lors de la définition de la mémoire de projet, la structure de ces éléments doit être souple et facile à utiliser. Cette structure doit tenir compte de tous les éléments de la mémoire sans omettre aucune information utile dans la compréhension du déroulement du projet.

III.2. Architecture Générale du Système d'Information

III.2.1. Niveaux de détail

L'architecture du système d'information dédié à la mémoire de projet que nous avons développée s'articule autour de deux niveaux : niveau concepts et niveau objets.

- Niveau concepts : Il contient deux sous-niveaux :
 - Sous-niveau paquetages : c'est le niveau qui offre un mécanisme général pour la partition des modèles et le regroupement des éléments de modélisation du système d'information. L'architecture est composée d'un ensemble de paquetages (cf Figure 26). Chaque paquetage contient des classes et leurs associations. Notre décomposition en paquetages ne suit pas des critères fonctionnels; mais chaque paquetage est un groupement d'éléments selon un critère purement logique dans le but d'avoir une cohérence forte entre éléments d'un même paquetage et un couplage faible entre paquetages. La forme générale du système d'information dédié à la mémoire de projet est exprimée par la hiérarchie de paquetages et par le réseau de relations de dépendance entre paquetages. Le paquetage principal est le paquetage *Corps* qui forme la colonne vertébrale structurelle du modèle : il contient les éléments les plus génériques de la mémoire de projet.
 - Sous-niveau Classes : c'est un niveau de détail plus fin qui contient la description statique de la mémoire de projet en termes de classes et de relations entre ces classes.

- Niveau objets : c'est le niveau instances des éléments de la mémoire de projet et les liens entre eux. Il permet d'explicitier les structures complexes des diagrammes de classes en s'appuyant sur des exemples réels.

III.2.2. Les paquetages

Les paquetages du système d'information pour la mémoire de projet de conception que nous proposons sont conçus pour prendre en compte un grand nombre d'aspects de tels projets ; notre système comporte les paquetages : produits, processus, organisation, documentation, décisions, acteurs, connaissances et point de vue. Chacun de ces aspects a un intérêt dans la reconstitution de la vie d'un projet. Ils sont dépendants, et il est nécessaire de gérer les liens entre les différents aspects, comme nous le verrons plus loin.

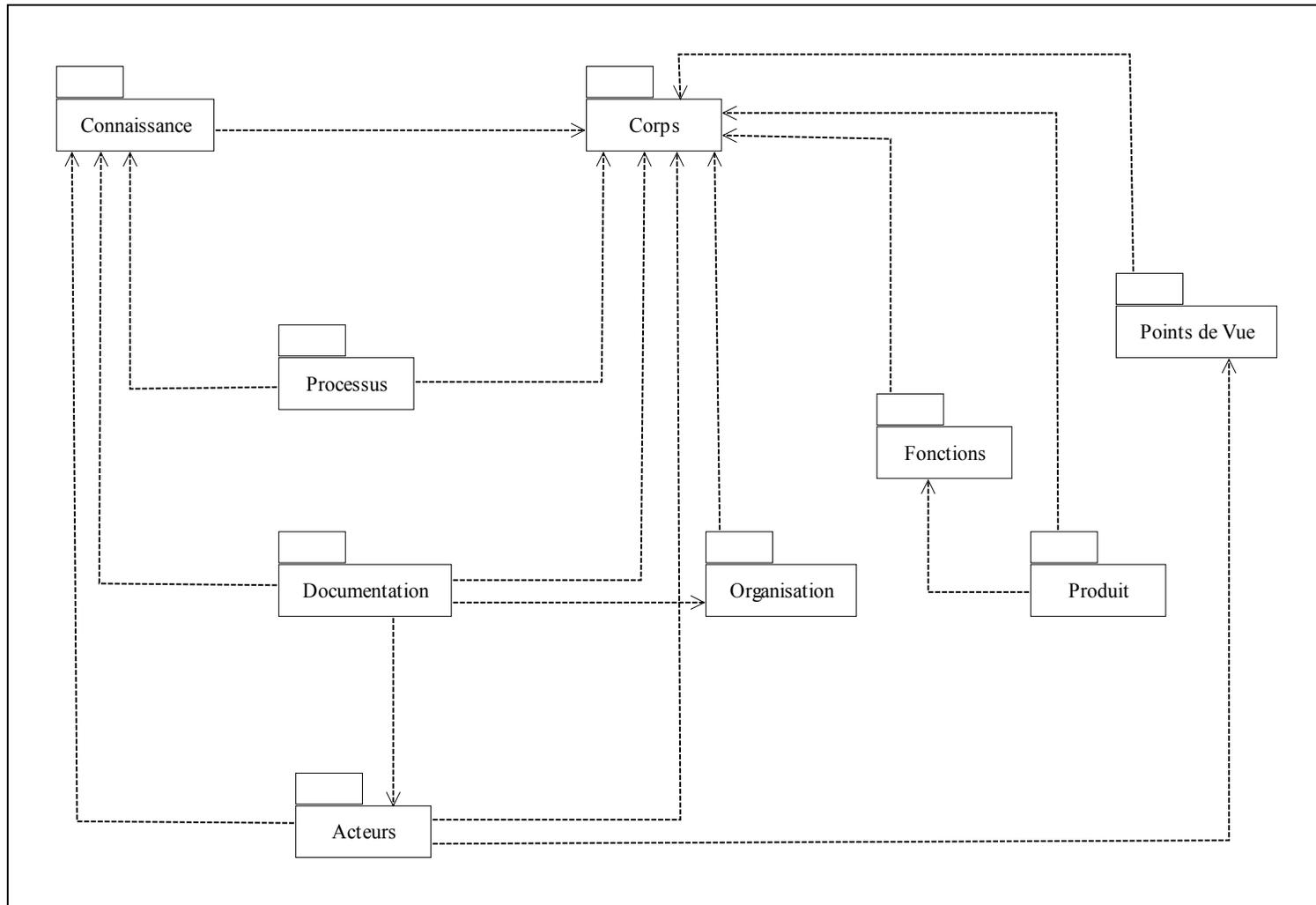


Figure 25 : Architecture Globale du Système d'Information.

Chacun de ces éléments est un des constituants nécessaires à la construction de la mémoire de projet.

- **Paquetage *Produit*** : décomposition structurelle du produit.
- **Paquetage *Processus*** : organisation des différentes étapes d'un processus.
- **Paquetage *Fonctions*** : cahier des charges fonctionnel que le produit doit satisfaire.
- **Paquetage *Documentation*** : pour gérer la structure, le contenu, la justification des choix techniques, indiquer les références à tout support d'information (image, texte, lien hypertexte...).
- **Paquetage *Organisation*** : structure de l'équipe de projet, structure d'une entreprise.
- **Paquetage *Points de Vue*** : définition des différents points de vue.
- **Paquetage *Acteurs*** : gestion des acteurs intervenant sur le projet, autorisations d'accès.
- **Paquetage *Connaissance*** : structuration de différents niveaux de connaissance.
- **Paquetage *Corps*** : il est destiné à gérer les attributs communs et les relations entre toutes les classes, ainsi que les vues. Il permet aussi de suivre l'évolution de tout objet, contrainte sur un objet et l'historique des relations entre objets. Par exemple, un rotor est un composant d'une turbopompe ; la création d'une nouvelle version du rotor va entraîner celle d'une nouvelle version de la turbopompe. Les modèles permettent de gérer toutes les versions du produit. Il est possible de « tracker » (laisser des traces permettant de retrouver l'origine de) principalement : les représentations, les choix, les décisions et leurs justifications, les solutions adoptées et celles qui ont été rejetées. Toutefois, c'est à l'utilisateur de décider ce qu'il souhaite mémoriser ou non ; par exemple, veut-il garder l'historique de la décomposition du produit ou non ? Veut-il archiver les documents de justification des décisions qui ont été prises lors d'une étape de conception d'un produit ou non ?

Bien évidemment, cet ensemble de paquetages n'est pas exhaustif. Comme nous le verrons lors de la présentation de la construction des contenus des différents paquetages, l'architecture proposée ainsi que la généralité du paquetage *Corps* permettent d'ajouter aisément d'autres concepts liés à la mémoire de projet.

III.3. Présentation des Modèles

Notre objectif n'est pas de proposer de nouveaux modèles d'organisation, processus, produit, acteurs etc. mais plutôt de montrer qu'il est possible, d'une part de bâtir des modèles sur la base de patrons de conception et, d'autre part, d'intégrer l'ensemble des constituants de la mémoire de projet en remontant au plus haut niveau d'abstraction les relations qui existent entre ces constituants.

III.3.1. Le Patron de Conception

Généralement, le terme patron désigne un modèle sur lequel travaillent les artisans pour fabriquer certains objets. C'est la traduction du terme anglais « pattern » qui désigne un modèle schématique ou un modèle simplifié d'une structure [Robert, 1993]. Ce concept a été introduit dans le domaine de l'ingénierie du logiciel par K. Beck et W. Cunningham [Beck et al, 1987]. Ils proposaient une première adaptation du concept de patron de conception (design pattern), introduit préalablement par C. Alexander [Alexander et al, 1977] dans le domaine de l'architecture des bâtiments, à la conception et à la programmation orientée objets.

Un patron tel que défini par C. Alexander « décrit à la fois un problème qui se produit très fréquemment dans un environnement et l'architecture de la solution à ce problème de telle façon qu'on puisse utiliser cette solution des millions de fois sans jamais l'adapter deux fois de la même manière ».

Quelques années plus tard, P. Coad [Coad, 1992] propose des patrons pour la modélisation conceptuelle des systèmes d'information. Il propose de faciliter l'analyse d'un système en identifiant les besoins selon sept patrons préétablis. Un patron orienté objet est alors vu comme « une abstraction d'un ensemble de classes qui peut être réutilisé encore et encore pour le développement d'applications ».

Un patron orienté objet offre alors une solution à un problème de modélisation orientée objet, en proposant des artefacts prédéfinis, adaptables à des problèmes similaires et à des technologies différentes d'implantation. Selon que le problème traité apparaît lors de l'analyse des besoins ou de la conception ou de l'implantation du système d'information, différents types de patrons sont distingués. On parle respectivement de patron d'analyse [Coad, 1996] [Fowler, 1997], de conception [Gamma et al, 1994] et d'implantation [Coplien, 1992].

Lorsqu'on observe les modèles utilisés pour représenter les objets manipulés par les différents acteurs d'un projet de conception, force est de constater une utilisation très fréquente de représentations permettant de décrire des décompositions de type arbre ou arborescence.

La recherche de généricité dans la conception de l'application informatique conduit, pour représenter de tels modèles, à utiliser le patron (ou encore schéma de conception) de la Figure 27 adapté de l'un de ceux proposés par [Gamma et al, 1994]. Ce patron comporte une classe *Racine* destinée à représenter la racine de la décomposition, car il a été constaté que celle-ci n'est jamais remise en cause. En effet, si l'on représente le WBS³⁵ d'un projet par exemple, la racine est d'emblée le nom du projet et le demeure ensuite.

Par contre, pour [Gamma et al, 1994] les feuilles doivent rester des feuilles, sans possibilité de décomposition ultérieure. Cette limite ne correspond pas à ce qui a été constaté sur le terrain,

³⁵ WBS : Work Breakdown Structure

l'utilisateur ayant souvent à décomposer des éléments qui constituaient initialement de telles feuilles ; il en résulte que le patron que nous proposons comporte une classe *ElémentRacine* représentant un nœud de la décomposition. Cette classe comporte une association réflexive (un nœud peut se composer de nœuds).

L'association entre *ElémentRacine* et *Racine* ainsi que l'association réflexive représentées sur la Figure 27 sont de type composition, mais le patron peut être adapté selon les besoins, et une ou plusieurs de ces compositions peuvent devenir des agrégations. De même, les multiplicités peuvent être adaptées.

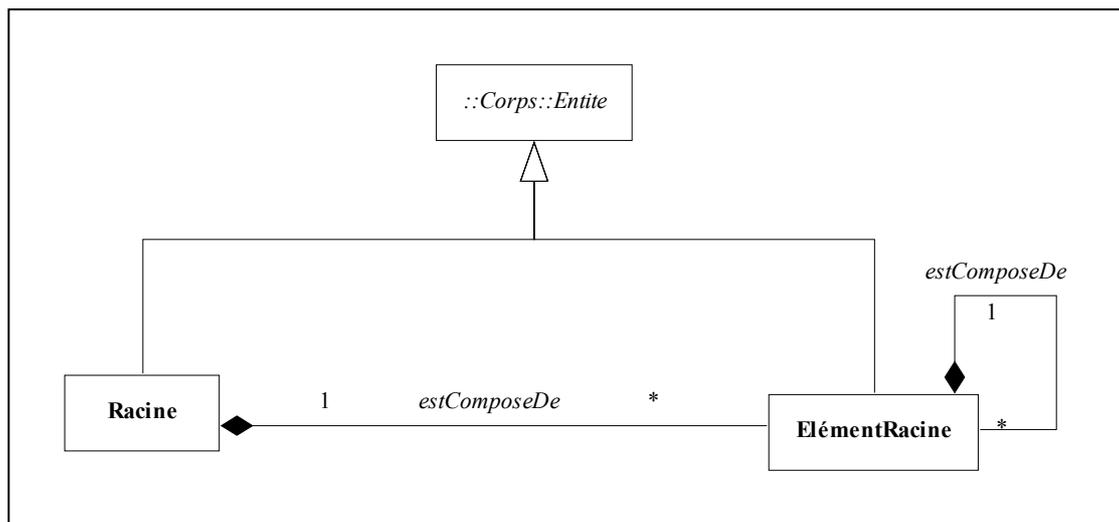


Figure 26 : Patron décrivant une décomposition arborescente.

Il est à noter qu'un tel patron peut être enrichi de contraintes décrites avec un langage tel qu'OCL (Object Constraint Language) [OMG, 2005] afin de spécifier par exemple l'impossibilité de créer une décomposition réflexive (si un *ElémentRacine_1* se compose d'un *ElémentRacine_2*, alors l'*ElémentRacine_2* ne peut se composer de l'*ElémentRacine_1*). Ces contraintes sont ainsi spécifiées une fois pour toutes.

Le patron ainsi décrit ne permet que de conserver la dernière version de l'arborescence, sans en gérer l'historique, qui sera abordé plus loin dans le présent chapitre. La classe abstraite *Entité* issue du paquetage *Corps* permet de le faire, comme cela sera présenté plus loin.

III.3.2. Modèle d'Organisation

1) Contexte

Dans la littérature, il existe un grand nombre de définitions du concept Organisation. Nous avons retenu la définition donnée par Cantzler car elle s'approche de nos besoins.

D'après [Cantzler, 1997] « Une organisation est une unité économique de coordination possédant des frontières identifiables, et fonctionnant de façon continue, en vue d'atteindre certains objectifs identifiables a priori et partagés par les membres de l'organisation ».

En effet dans la définition donnée par Cantzler, l'unité économique pourra être un projet ou une entreprise, ce qui correspond à nos besoins pour la modélisation des organisations.

Ainsi dans le cas de la mémoire de projet, l'organisation d'un projet dépend du choix d'une structure dans laquelle œuvrent des participants. Cette organisation peut être fonctionnelle, divisionnelle ou matricielle:

- **Organisation fonctionnelle :**

C'est une organisation par fonction (production, finance, commercial..) qui correspond à une spécialisation plus ou moins poussée par type de compétence.

- **Organisation divisionnelle :**

Cette organisation sépare les divisions opérationnelles, définies sur des critères de lignes de produits et/ou de marchés et relativement indépendantes les unes des autres. Dans ce contexte, chaque division se structure à son tour, le plus souvent en faisant appel à la forme fonctionnelle.

- **Organisation matricielle :**

Cette organisation croise la vision divisionnelle par produit avec la vision fonctionnelle pour au moins une fonction. « L'objectif poursuivi par cette forme organisationnelle est de combiner les avantages de la spécialisation (favoriser par l'organisation fonctionnelle) et la limitation des problèmes de coordination de l'ensemble des activités liées à un produit (obtenue par l'organisation divisionnelle) » [Giard, 1991].

2) Diagramme de classes

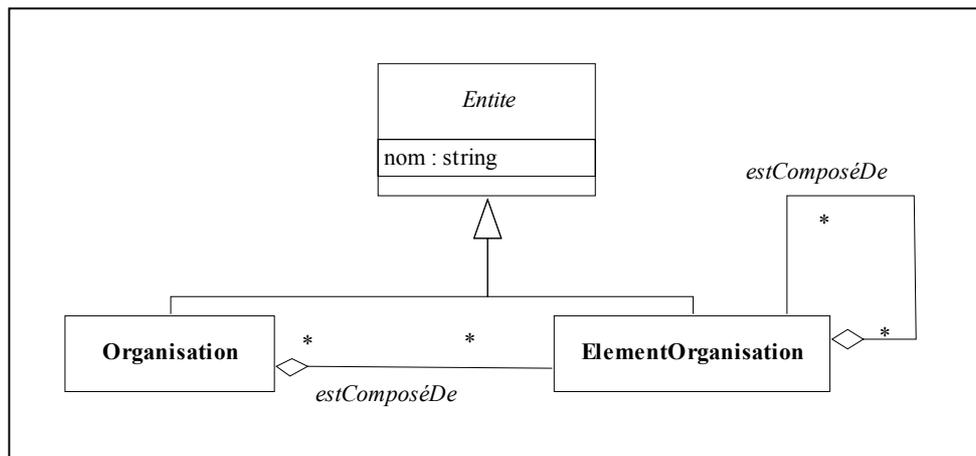


Figure 27 : Diagramme de classes de description de l'organisation.

Un élément de la mémoire de projet du type *Organisation* peut être décomposé en un certain nombre d'éléments d'organisation (*ElementOrganisation*). Un élément d'organisation peut à son tour être décomposé en éléments d'organisation. Par rapport au patron de base, les multiplicités du côté composite sont passées de 1 à * de manière à pouvoir prendre en compte des organisations dans lesquelles un élément peut appartenir à plusieurs entités hiérarchiquement supérieures.

3) Diagramme d'objets

Le diagramme d'objets de la Figure 29 (resp. Figure 30) a pour objectif de présenter le découpage hiérarchique (resp. matriciel) d'une organisation en éléments d'organisation.

Exemple 1 :

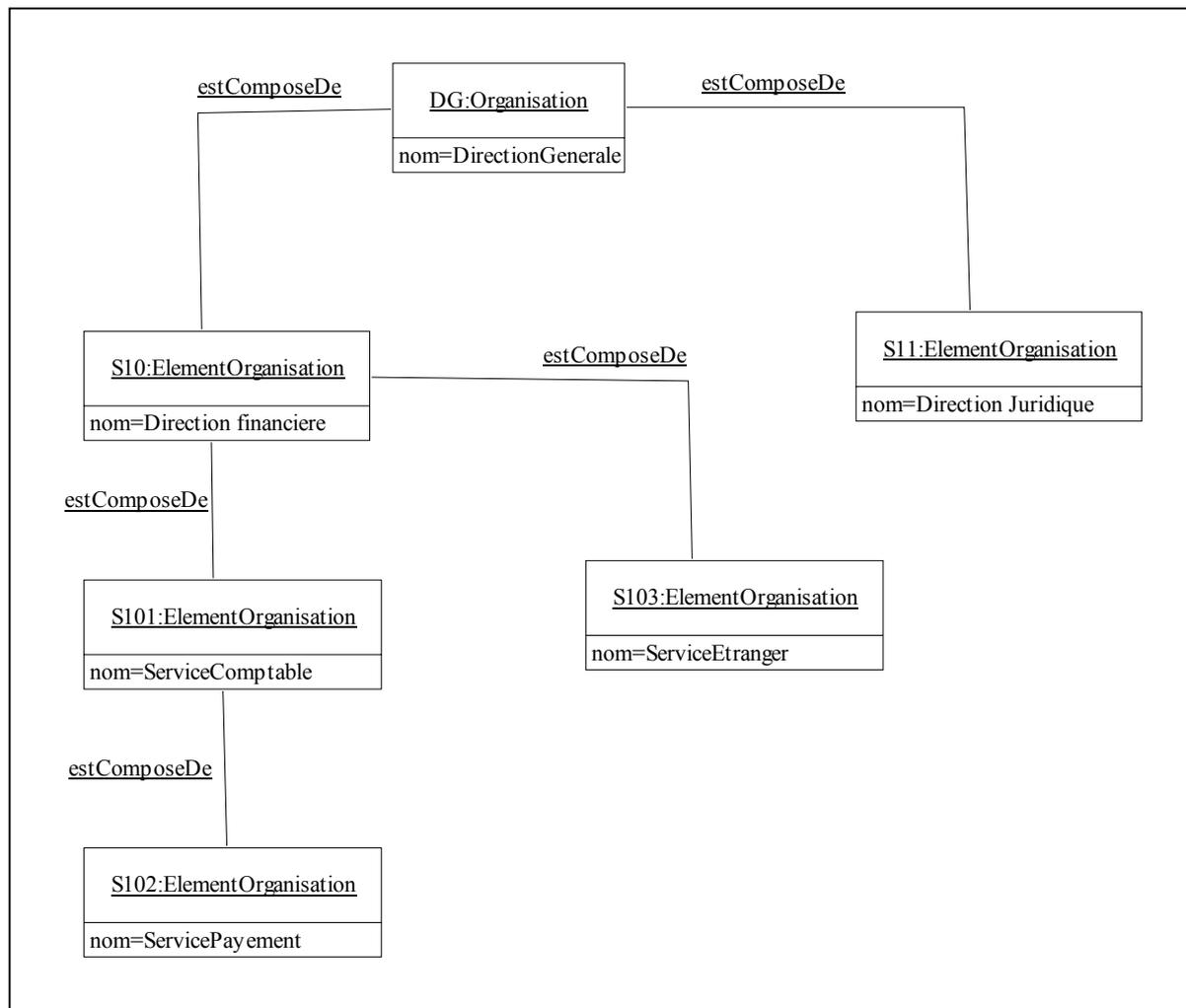


Figure 28 : Diagramme d'objets de décomposition d'une organisation fonctionnelle hiérarchique (département-sous-département).

La Figure 29 décrit un exemple d'organisation de type fonctionnelle. Ainsi un élément « DG » du type *Organisation* est composé de l'élément « S10 » et « S11 » qui sont respectivement des éléments de type *ElementOrganisation*. A son côté l'élément « S10 » est décomposé de deux éléments « S101 » et « S103 » qui sont aussi des éléments de type *ElementOrganisation*.

Exemple 2 :

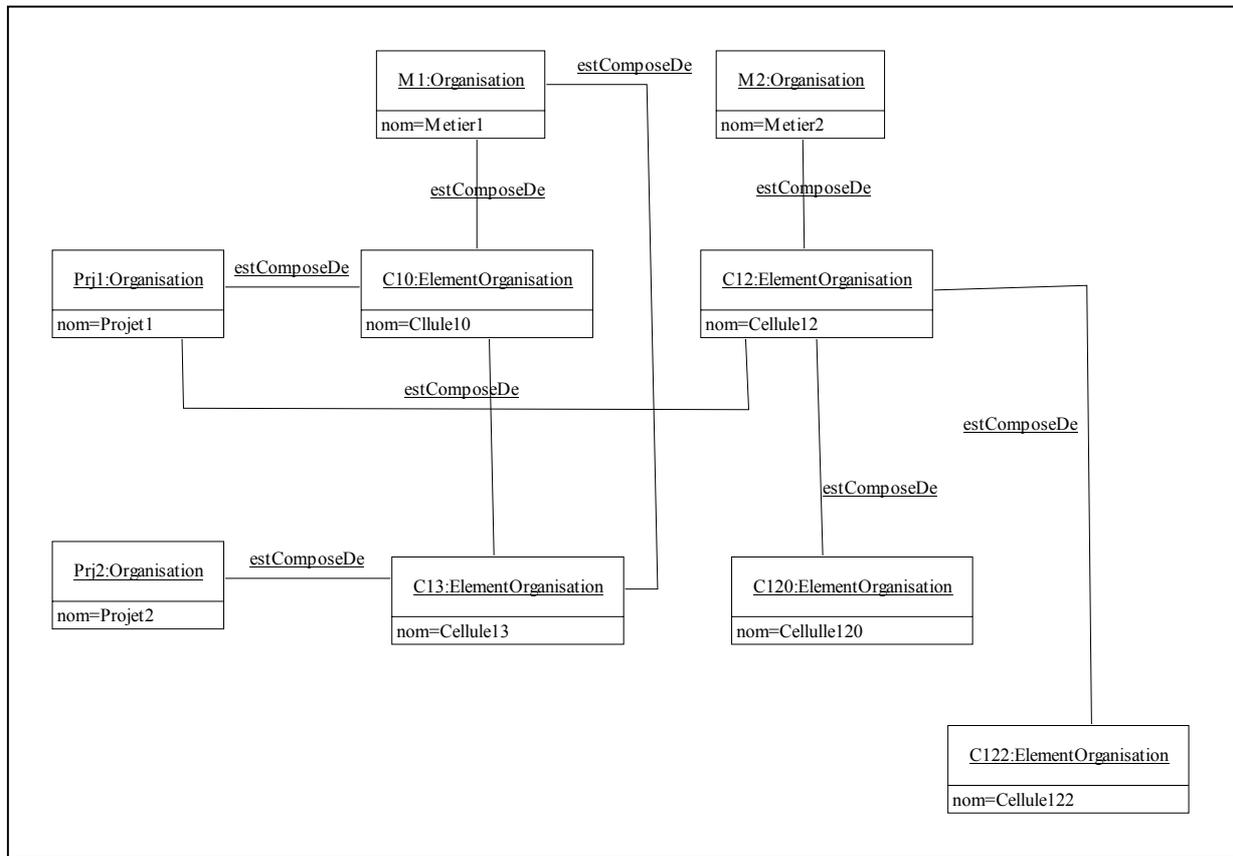


Figure 29 : Diagramme d'objets d'une organisation matricielle (département – projet).

Le « projet1 » (Prj1) du type *Organisation* est composé de la « cellule10 » (C10) et la « cellule12 » (C12). C10 et C12 sont du type *ElementOrganisation*. Le « projet2 » (Prj2) élément du type *Organisation* est composé d'une seule « cellule13 » (C13) du type *ElementOrganisation*. Le « métier 2 » (M2) élément du type *Organisation* est composé de la « cellule12 » (C12) qui est elle-même composée de « cellule 120 » (C120) et « cellule122 » (C122) qui appartiennent à la classe *ElementOrganisation*.

De même il est possible d'ajouter des contraintes OCL [OMG, 2005] pour s'assurer de la cohérence d'appartenance d'un *ElementOrganisation* à une *Organisation* par rapport à son parent.

Par exemple l'écriture de la contrainte d'agrégation entre les deux éléments *Organisation* et *ElementOrganisation* dans le langage OCL peut être écrite de la façon suivante :

Context Organisation

Inv :
self -> notEmpty() implies
elements ->forAll(elt : OrganisationElement | elt->notEmpty())

III.3.3. Modèle de Processus

1) Contexte

Quelques définitions du processus de conception ont été présentées dans le paragraphe I.7.7.1 du chapitre 1 du présent mémoire. Le modèle simplifié ne prend en compte que l'aspect structurel d'un processus. La raison est qu'on ne cherche pas dans le présent travail à avoir une meilleure représentation du modèle de processus que les modèles déjà existants dans la littérature mais d'offrir une base qui sera capable d'intégrer des modèles capable d'interagir avec n'importe quel autre modèle existant et de faire les liens nécessaires entre modèles. Quant aux aspects temporels, ils seront présentés dans le paragraphe III.3.11.

2) Diagramme de classes

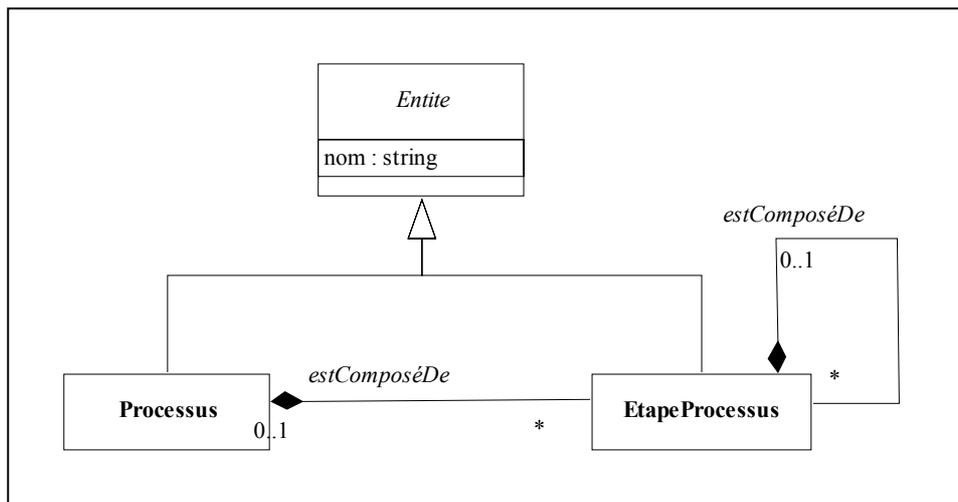


Figure 30 : Diagramme de classes de description de Processus.

Un *processus* est un élément de la mémoire de projet. Il pourra être décomposé en certaines étapes de processus de conception (*EtapeProcessus*). Chaque étape à son tour pourra être décomposée en sous étapes de processus. Une étape appartient à un et un seul processus. Ces étapes de processus sont aussi des éléments de la mémoire de projet.

3) Diagramme d'objets

Le diagramme d'objets de la Figure 32 a pour objectif de présenter la décomposition structurelle d'un processus en étapes de processus. Il s'agit de la décomposition du processus « conception Turbopompe » en étapes de processus « définition », « test » et « rédiger doc ».

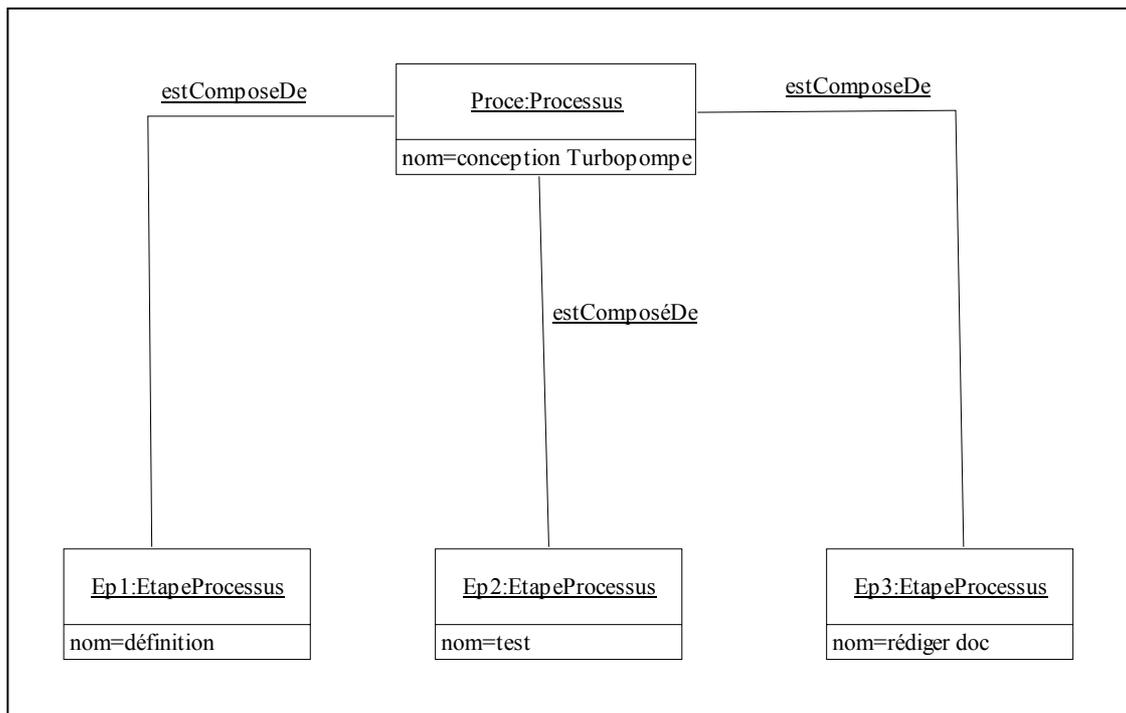


Figure 31 : Diagramme d'objets d'un Processus.

Le processus «conception Turbopompe» (Proce) est décomposé en trois étapes : « définition » (Ep1), « test » (Ep2) et « rédigerDoc » (Ep3). Ces étapes sont des éléments du type *EtapeProcessus*. Le lien entre les éléments Ep1, Ep2 et Ep3 et l'élément Proce est matérialisé par la relation *EstComposéDe*. Pour le moment on ne s'intéressera pas à la chronologie des étapes d'exécution des processus, qui sera traité plus loin dans le présent chapitre.

III.3.4. Le modèle de Produit

1) Contexte

Le modèle de produit que nous présentons ici sert à représenter la décomposition structurelle du *produit*. Il s'agit de modéliser l'arbre décrivant l'ensemble des constituants du produit et leurs regroupements en sous-ensemble (*ElementProduit*). Ceci permet notamment d'établir la nomenclature du produit (liste des constituants).

2) Diagramme de classes

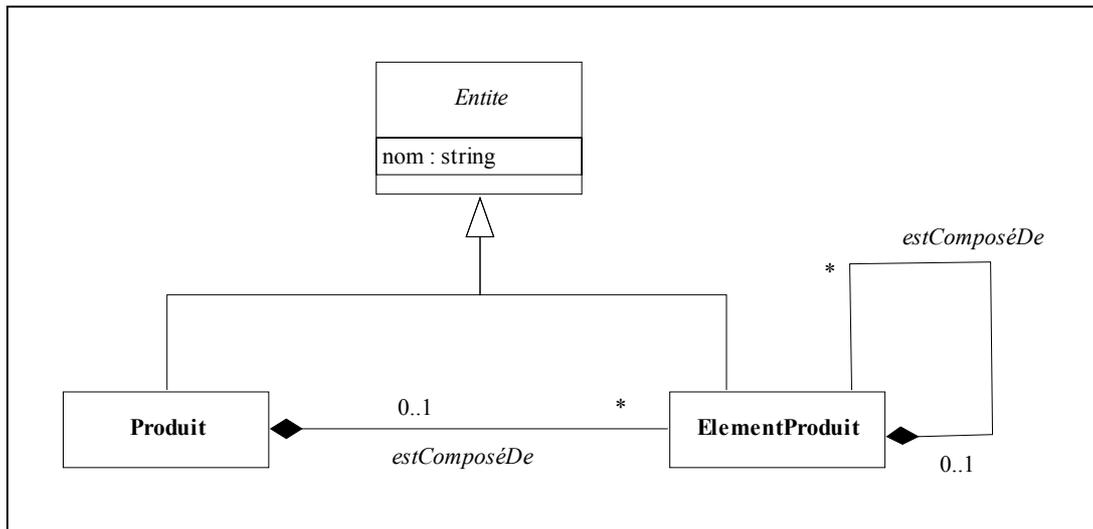


Figure 32 : Diagramme de classes de description de Produit.

Un produit peut se décomposer en éléments de produits (*ElementProduit*), eux-mêmes, pouvant être décomposé en d'autres éléments de produits dont le nombre de niveaux a priori non défini. Concernant les associations, il s'agit bien de compositions, car la suppression d'un produit ou de l'un de ses éléments entraîne la suppression de la partie de l'arbre qui lui est rattachée.

3) Diagramme d'objets

Le diagramme d'objets de la Figure 34 présente la décomposition structurelle du produit « Turbopompe » en éléments de produits « Rouet », « Palier », « joint1 » et « joint2 ».

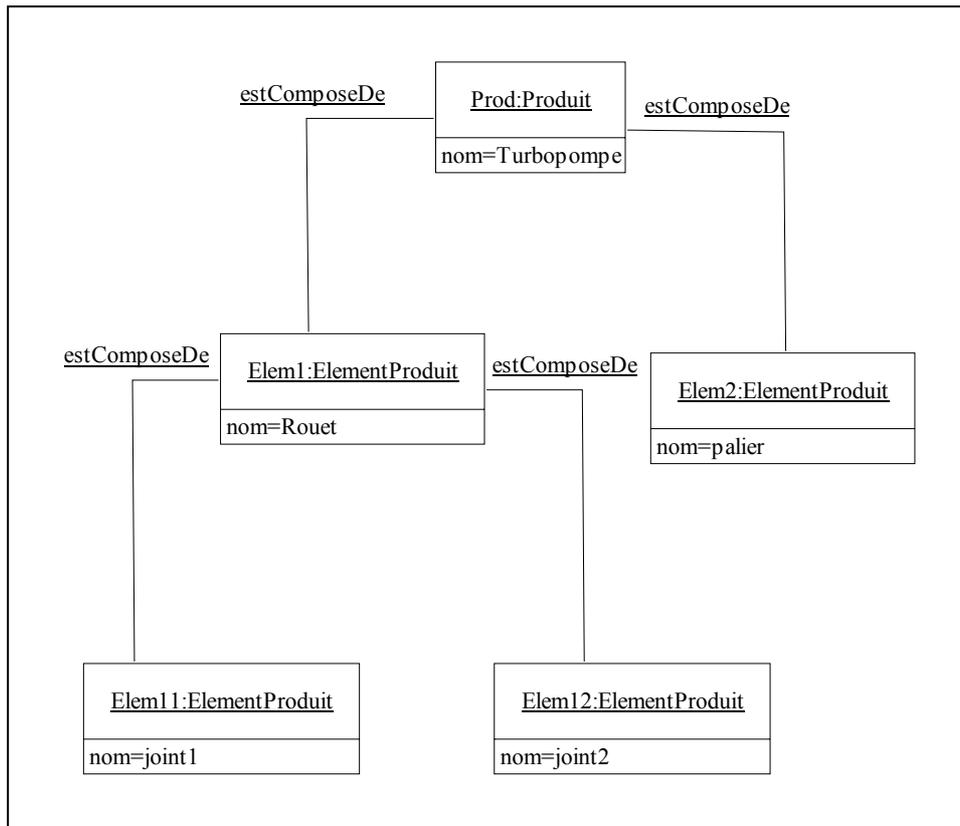


Figure 33 : Diagramme d'objets d'un Produit.

III.3.5. Modèle de Fonctions du Produit

1) Contexte

C'est généralement suite à une analyse fonctionnelle³⁶ qu'un produit est défini en termes de fonctions. Plusieurs définitions sont données à ce concept [AFNOR, 1990], [PDMIC, 1998]. L'ensemble de ces définitions s'accorde sur le fait qu'une fonction est une expression d'un service attendu par un produit.

Définitions

D'après l'AFNOR X50-150 une fonction peut être définie comme suit :

- **Fonction de service** : Action attendue d'un produit (ou réalisée par lui) pour répondre à un élément du besoin d'un utilisateur donné. Il faut souvent plusieurs fonctions de service pour répondre à un besoin. Elles comprennent les fonctions d'usage

³⁶ Démarche qui consiste à recenser, caractériser, ordonnancer, hiérarchiser et valoriser les fonctions [AFNOR, 90]

traduisant la partie rationnelle du besoin, et les fonctions d'estime traduisant sa partie subjective.

- **Fonction technique** : Action interne au produit (entre ses constituants), choisie par le concepteur-réalisateur, dans le cadre d'une solution, pour assurer des fonctions de service. Une fonction technique répond à un besoin technique du concepteur-réalisateur et peut être ignorée par l'utilisateur final du produit.
- **Fonction de transfert** : C'est une fonction qui correspond à une relation entre deux ou plusieurs éléments extérieurs avec le produit ou par son intermédiaire.
- **Fonction de contrainte** : C'est une fonction qui correspond à une relation directe d'un extérieur avec le produit.
- **Critère d'appréciation d'une fonction** : Caractère retenu pour apprécier la manière dont une fonction est remplie, ou une contrainte respectée. Dans la mesure du possible, le critère d'appréciation doit être accompagné d'une échelle permettant de situer son niveau. Pour une même fonction, il y a souvent plusieurs critères d'appréciation de natures différentes.

2) Diagramme de classes

Le diagramme de classes de la Figure 35, présente les différentes fonctions d'un produit. Une fonction (*Fonction*) peut se spécialiser en Fonction Contrainte (*FonctionContrainte*), Fonction Transfert (*FonctionTransfert*), Fonction Technique (*FonctionTechnique*) ou en Fonction Service (*FonctionService*) ce qui explique la multiplicité 1 du côté du Produit au niveau de la relation possède.

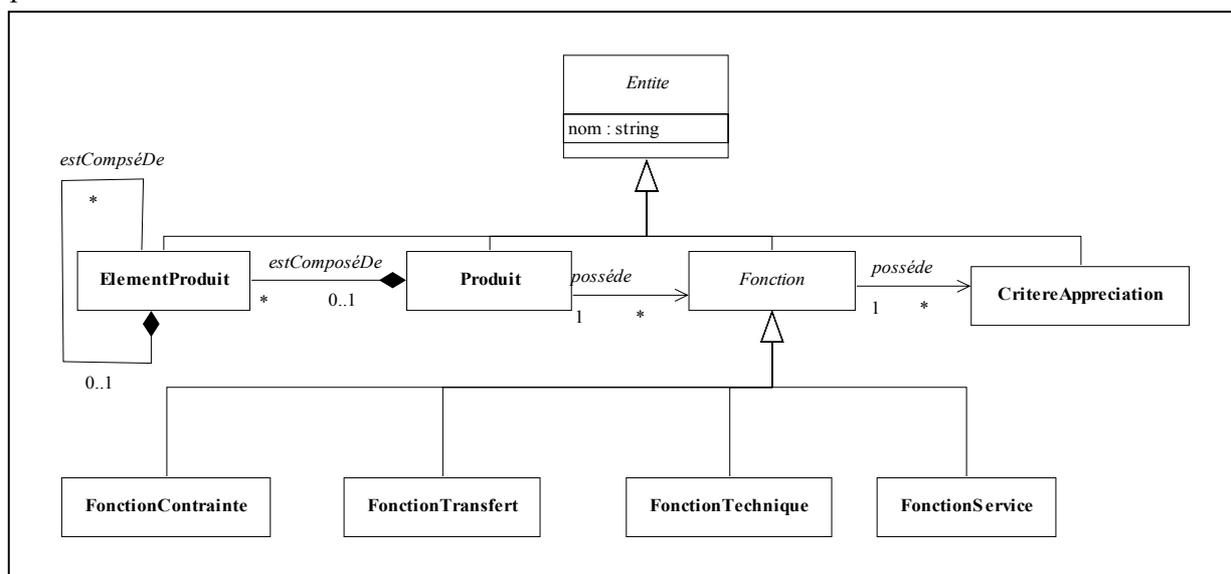


Figure 34 : Diagramme de classes de description des Fonctions de Produit.

3) Diagramme d'objets :

Le diagramme d'objets de la Figure 36 décrit quelques-unes des fonctions du produit « treuil d'hélicoptère » et quelques critères d'appréciation qui correspondent à ces fonctions.

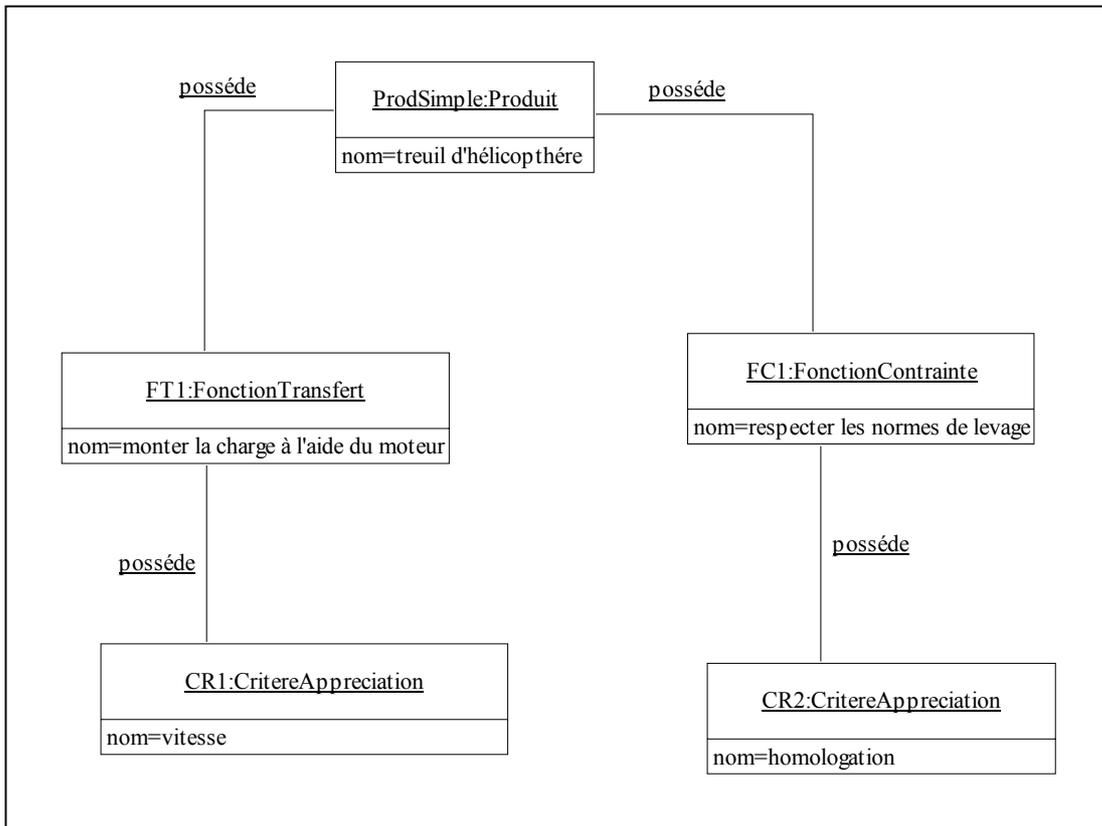


Figure 35 : Diagramme d'objets de description des Fonctions de Produit.

III.3.6. Les Documents

1) Contexte

Plusieurs définitions du concept de document existent dans la littérature [BNAE, 1992] [Maurino, 1993] [AIT, 1997] [IPDMUG, 1997] [PDMIC, 1998]. En particulier pour L. Gzara [Gzara, 2000] : « Un document est donc un support d'enregistrement de connaissances (un conteneur) qui a pour finalité de décrire en partie un objet technique ou les processus qui lui sont attachés ». C'est la définition qui s'approche le plus de nos besoins car on y trouve la notion de connaissance et d'objet technique, deux concepts qui nous intéressent dans la modélisation de documents. Le modèle proposé ne prend en compte que la structure hiérarchique du document ceci afin de ne pas décrire explicitement une structure-type « statique » (partie, section, sous-section, chapitre...) qui risquerait d'être inadaptée dans certains cas particuliers. La classe

abstraite *ElementDocument* permet de décrire de manière générique n'importe quelle décomposition d'un *document*. La structure arborescente du *document* est alors représentée par l'association réflexive « *estComposéDe* ».

2) Diagramme de classes

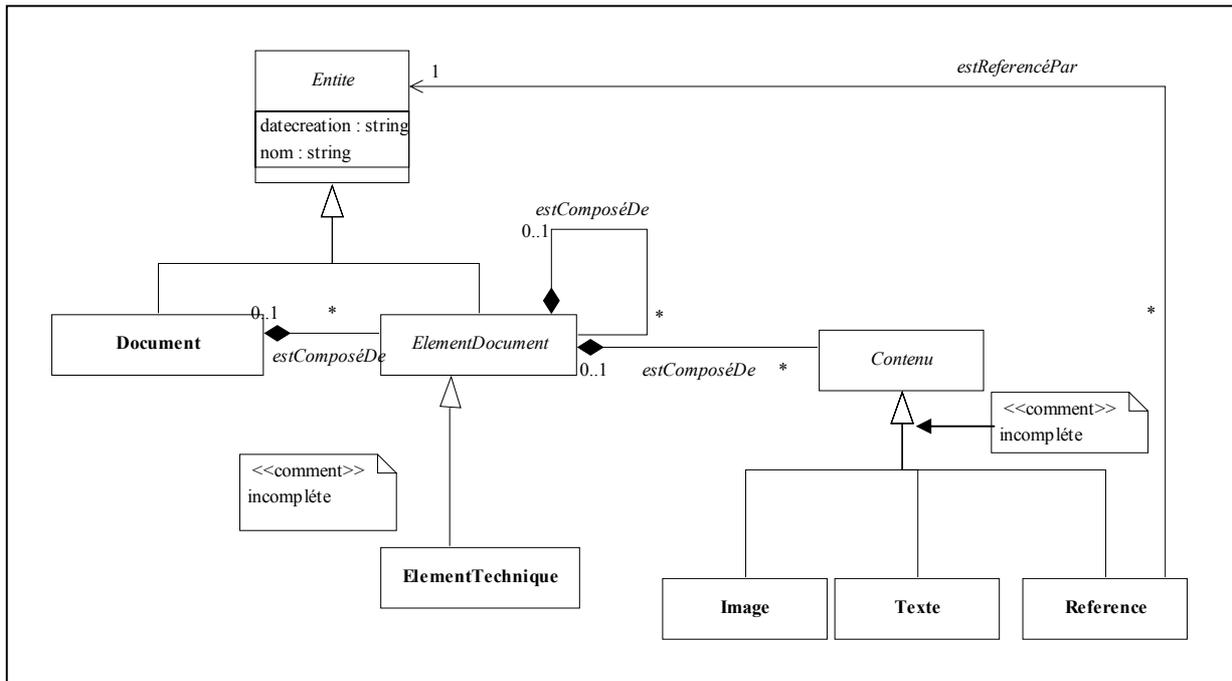


Figure 36 : Diagramme de classes de description de Document.

Un *ElementDocument* se compose de *Contenu* pouvant être une *Image*, du *Texte* ou une *Référence* à une autre entité, ou un autre type d'élément non encore défini (d'où la mention {incomplete}). Il existe une autre spécialisation de la classe abstraite *ElementDocument* qui sera présentée plus loin.

La classe abstraite *ElementDocument* permet de décrire de manière générique n'importe quelle décomposition d'un *document*. La structure arborescente du *document* est alors représentée par l'association réflexive « *estComposéDe* ».

3) Diagramme d'objets

Le diagramme d'objet de la Figure 38 montre la décomposition hiérarchique du document « Dossier de définition de la Turbopompe ». Il illustre un exemple d'application du diagramme de classes de description de document présenté à la Figure 37.

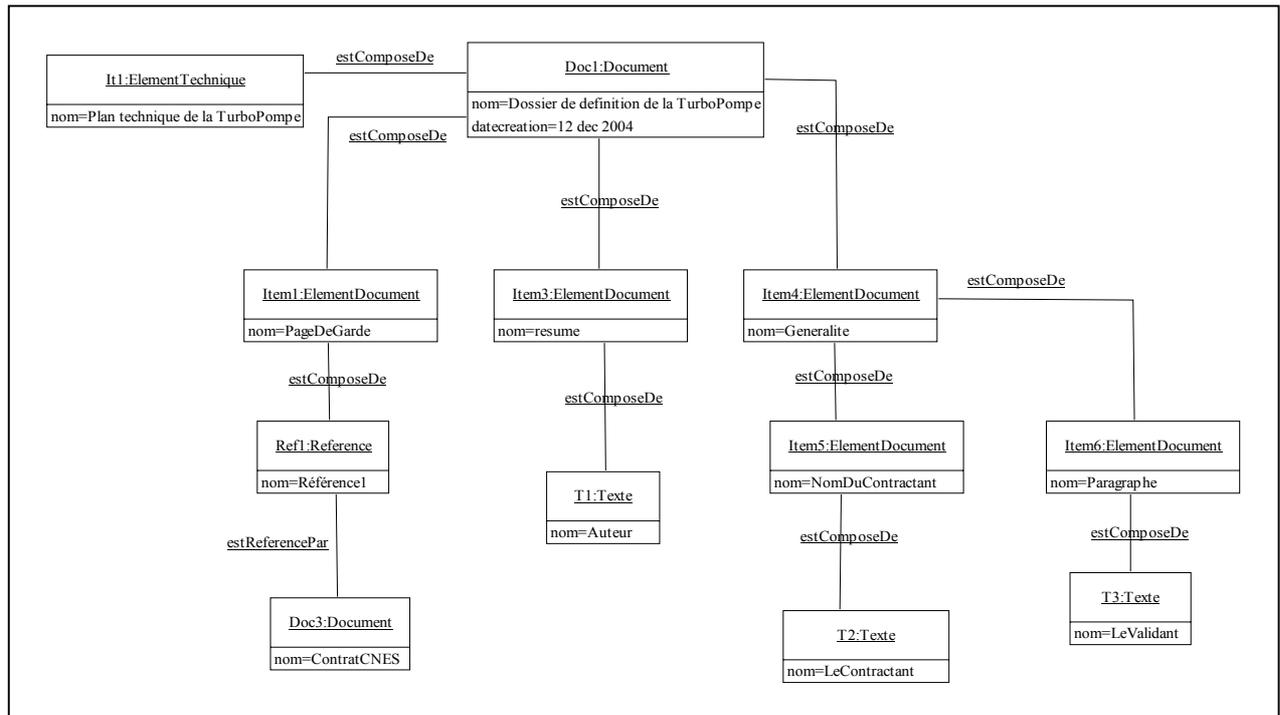


Figure 37 : Diagramme d'objets de description de Document.

Un élément « Dossier de définition de la turbopompe oxygène » (Doc1) du type *Document* est composé des éléments « PageDeGarde » (Item1), « résumé » (Item3) et « Généralité » (Item4) des éléments du type *ElementDocument*. L'élément Item3 est composé de l'élément « Text1 » du type *Texte*. L'élément Item1 est composé d'un élément « Ref1 » du type *Référence* qui référence un élément « ContratCNES » (Doc3) du type *Document*. L'élément Doc1 est composé d'un élément « Plan technique de la Turbopompe » (It1) du type *ElementTechnique*.

4) Lien avec l'élément Acteur

4.1. Diagramme de classes

Le modèle suivant illustre le lien entre deux éléments n'appartenant pas aux mêmes paquets. C'est un modèle qui permet de déterminer l'*acteur* d'une opération de validation d'un document d'une part et d'autre part de déterminer l'auteur d'un *document*. Ce type de modèle aide à garder une trace des responsabilités vis-à-vis d'un *document*.

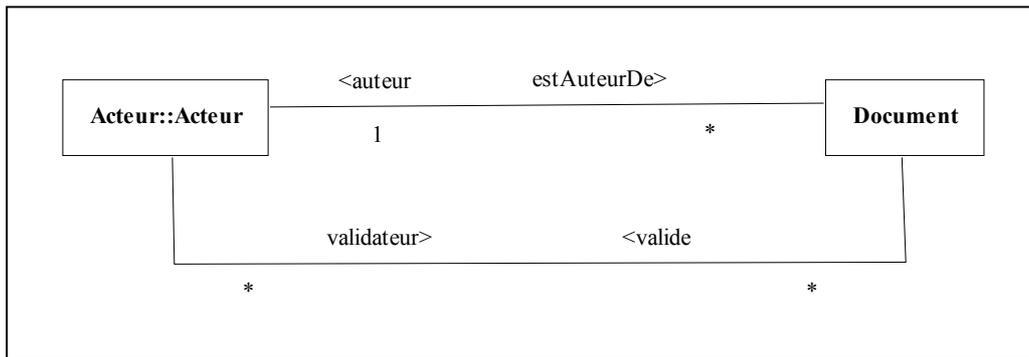


Figure 38 : Diagramme de classes faisant le lien entre Acteur et Document.

4.2. Diagramme d'objets

Le diagramme d'objet de la Figure 40 présente un exemple d'application du diagramme de classes de la Figure 39. Il s'agit de déterminer l'auteur d'un document et qui l'a validé.

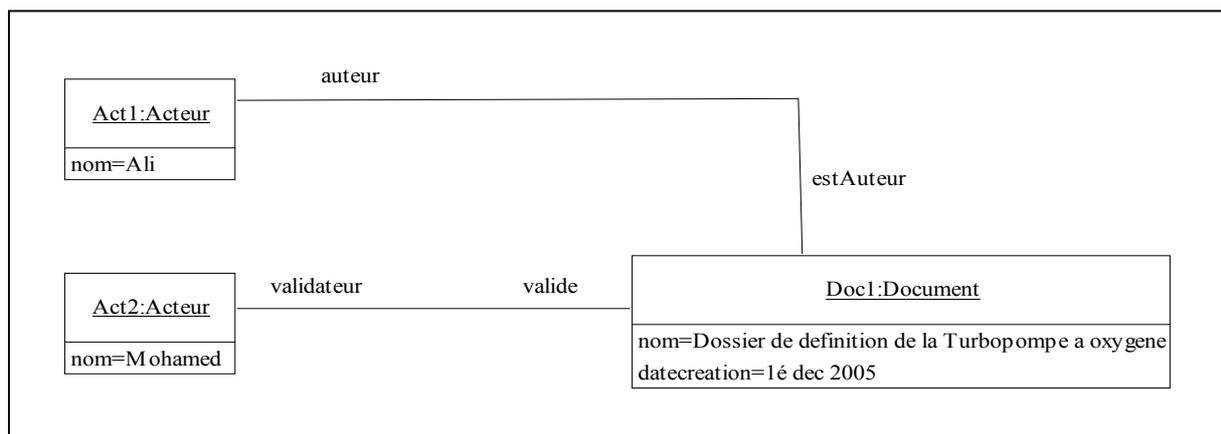


Figure 39 : Diagramme de d'objets de lien entre Acteur et Document.

Un élément « Ali » (Act1) du type *Acteur* est auteur de l'élément « Dossier de définition de la turbopompe » (Doc1) du type *Document* validé par l'élément « Mohamed » (Act2) du type *Acteur*. L'élément Act2 est validateur du document Doc1.

Cet exemple montre le moyen de trouver qui est l'acteur d'une opération de validation d'un document dans une mémoire de projet. C'est un moyen de tracer un choix de décision sur un élément du type *document*.

III.3.7. Lien entre les différents modèles

1) Contexte

Dans le présent paragraphe on s'intéresse à expliciter davantage les relations entre éléments issus de paquetages différents. Ainsi comme le montre la Figure 26, il existe des liens entre les différents paquetages. La classe abstraite *Entité* issue du paquetage *Corps* (voir Figure 26) contient tous les attributs communs aux différents éléments de la mémoire de projet, comme la *date de création*, la *date de la dernière modification*... Ceci permet de gérer une partie de l'historique de la mémoire de projet.

2) Diagramme de classes

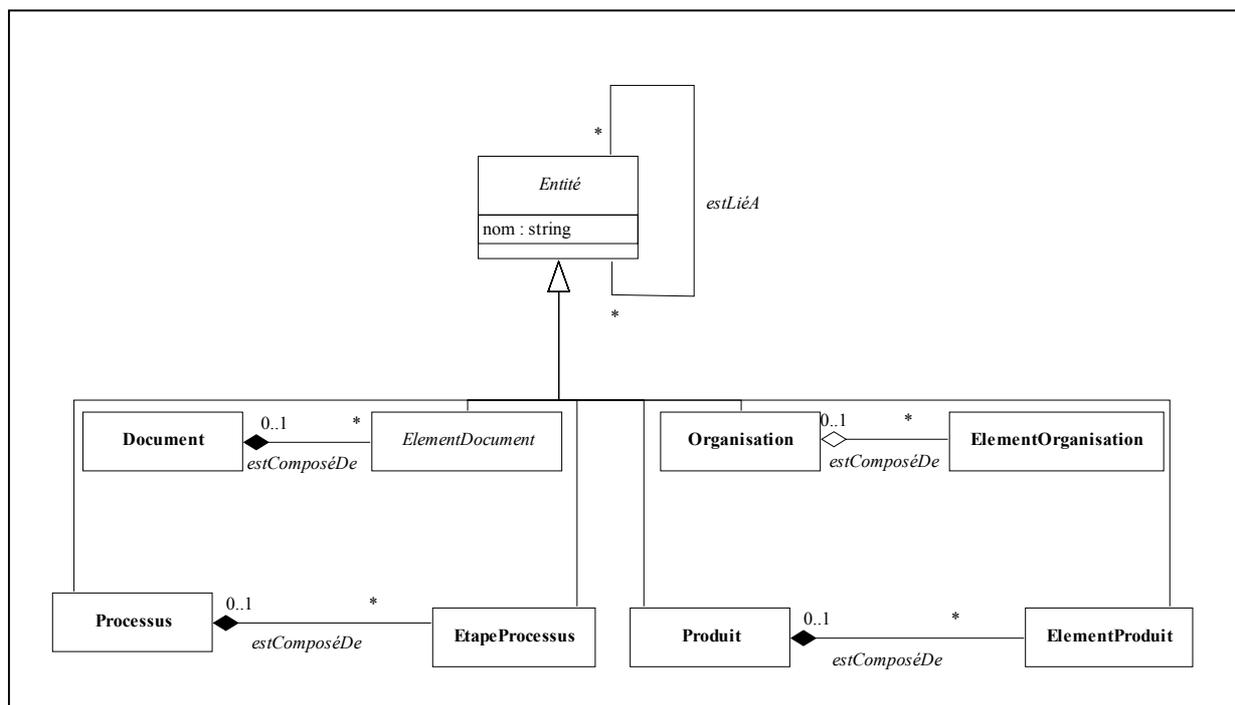


Figure 40 : Diagramme de classes de description de liens entre les différents modèles.

La Figure 41 présente une partie des éléments qui peuvent constituer la mémoire de projet. Un élément de cette mémoire de projet peut être du type *Document*, *ElementDocument*,

Organisation, ElementOrganisation, Processus, EtapeProcessus, Produit, ElementProduit. Ces éléments sont reliés entre eux par le biais de la classe *Entité*. La relation « *estLiéA* » a été placée au plus haut niveau d'abstraction de manière à permettre la mise en relation d'éléments quelconques de la mémoire de projet entre eux avec une sémantique simple. Des relations à sémantique plus riche seront présentées dans le paragraphe III.3.11.

3) Diagrammes d'objets :

Le diagramme d'objet de la Figure 42 a pour objectif de présenter le lien qui peut exister entre le produit « Turbopompe » et le processus « conception de Turbopompe ». Le diagramme d'objet de la Figure 43 a pour objectif de représenter le lien qui existe entre le produit « Turbopompe », le processus « conception de Turbopompe » et les deux organisations « Projet1 » et « Projet2 ». Ces deux diagrammes d'objets illustrent des exemples d'application du diagramme de classes de description de lien entre des éléments n'appartenant pas aux mêmes paquetages. Le lien entre les éléments n'appartenant pas au même paquetage est matérialisé par la relation « *estLiéA* ».

Exemple 1 : Lien *Produit-Processus*

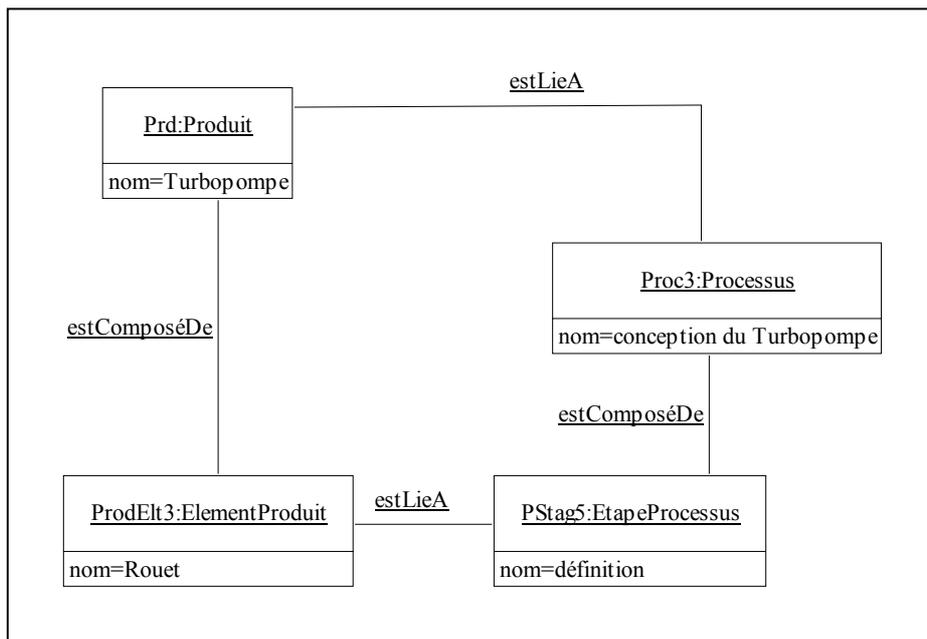


Figure 41 : Diagramme d'objets de description de lien Produit-Processus.

Un élément « Turbopompe » (Prd) du type *Produit* est composé d'un élément « Rouet » (ProdElt3) du type *ElementProduit*. L'élément Prd du type *Produit* est lié à l'élément Proc3 du type *Processus* grâce à la relation du type *estLiéA*. De même l'élément de produit ProdElt3 du type *ElementProduit* est lié à l'élément PStag5 élément du type *EtapeProcessus* au moyen de la relation du type « *estLiéA* ».

Exemple 2 : Lien entre *Produit-Processus-Organisation*

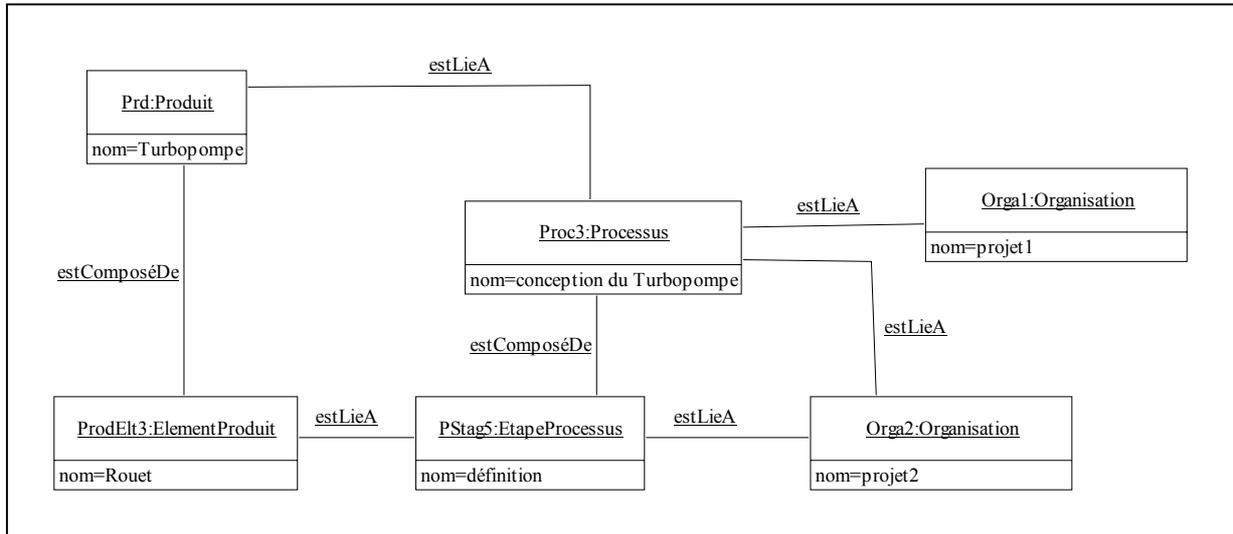


Figure 42 : Diagramme d'objets de description de lien Processus-Organisation.

L'élément « Turbopompe » (Prd) est de type *Produit*. Il est d'une part lié à l'élément « Conception du Turbopompe » (Proc3) élément du type *Processus* et d'autre part il est composé de « Rouet » (ProdElt3) élément du type *ElementProduit*. Ce dernier est lié à l'élément «définition» (PStag5) élément du type *EtapeProcessus*. L'élément Proc3 est lié aux deux éléments « projet1» (Orga1) et « projet2 » (Orga2) du type *ElementOrganisation*. L'élément PStage5 est lié à l'élément Orga2. L'ensemble des liaisons entre ces éléments est matérialisé par la relation « *estLiéA* ».

III.3.8. Modèle de Paramètres

1) Contexte

[Escande, 1996] définit le concept paramètre comme étant « un concept qui a une structure complexe qui lui permet d'avoir un fort pouvoir de description : il assure la gestion du processus de raisonnement, la communication avec l'utilisateur, l'évaluation des solutions, la trace et l'explication du raisonnement suivi ou encore la gestion ».

[Cantzler, 1997] a défini le concept de paramètre comme étant « un objet caractérisant d'un point de vue qualitatif ou quantitatif un autre objet ». A travers le concept paramètre, sont représentées toutes les grandeurs imposées ou à déterminer pour le produit. Ces spécifications peuvent être de différentes natures telles que : mécanique, thermique, électrique, etc. Elles sont caractérisées par leur unité de mesure.

D'après la définition d'Escande le concept paramètre est utilisé dans le raisonnement, évaluation de solutions, trace et explication du raisonnement qui n'est autre qu'une partie qui concerne la

logique de raisonnement dans une mémoire de projet. Tandis que celle de Cantzler se focalise sur l'élément produit sans se préoccuper du domaine d'application objet autour duquel émerge la conception (mécanique, électrique, etc.).

Ainsi il est souhaitable de gérer la notion de paramètre dans la mémoire de projet. C'est un élément qui sert à caractériser les éléments de la mémoire de projet.

2) Diagramme de classes

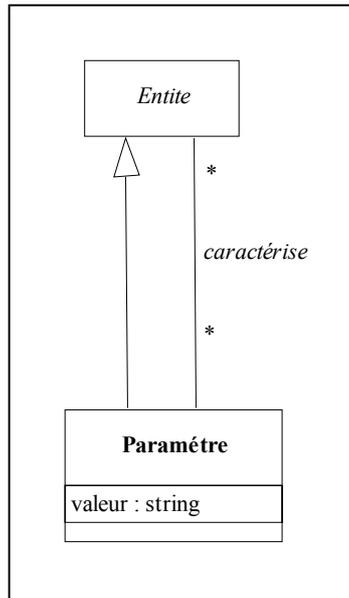


Figure 43 : Diagramme de classes de description de Paramètres.

Un élément de la mémoire de projet peut être du type *paramètre*. Un paramètre peut concerner, un produit, un processus, un point de vue etc. Un élément de la mémoire de projet peut être concerné par plusieurs éléments du type *paramètre*. Un paramètre est essentiellement caractérisé par un nom qu'on peut hériter de la classe *Entité* et une valeur.

3) Diagramme d'objets

Le diagramme d'objet de la Figure 45 a pour objectif de présenter les paramètres du processus « Conception du produit Turbopompe ». C'est un exemple d'application du diagramme de classes de description de paramètres de la Figure 44.

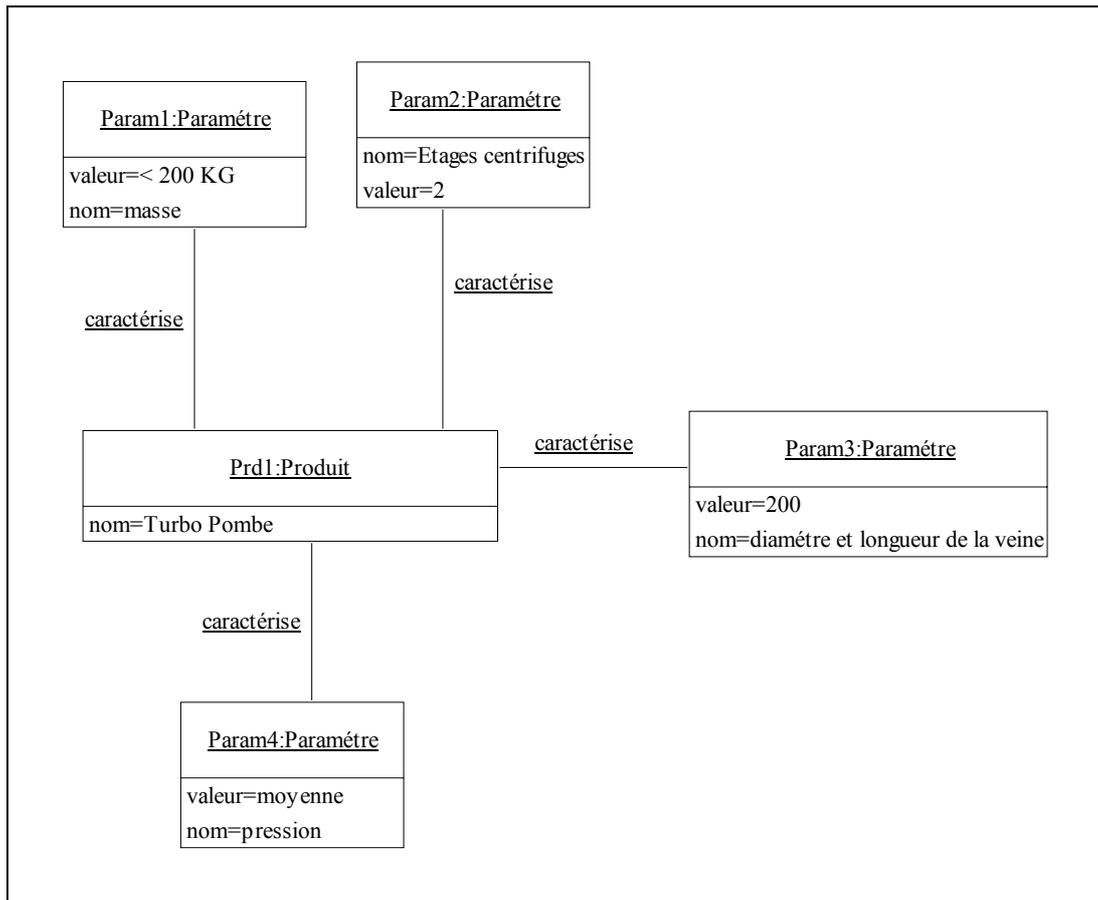


Figure 44 : Diagramme d’objets de description de Paramètres.

L’élément « Conception du produit Turbopompe » (Proc1) est un élément du type *Processus*. Il est caractérisé par les paramètres masse» ayant « une valeur < 200 » (Param1), « Etages centrifuges » ayant « une valeur=2 » (Param2), « Diamètre et longueur de la veine » ayant « une valeur =200 » (Param3) et « Pression » ayant une « valeur moyenne » (Param4). Param1, Param2, Param3 et Param4 sont des éléments de type *Paramètre*.

III.3.9. Modèle de Contraintes

1) Contexte

[Harani, 1997] a défini le concept de contrainte comme étant « une relation entre un ou plusieurs objets, un objet pouvant être une variable numérique ou symbolique ».

Ainsi et d’après Fenves [Fenves et al, 1985], les contraintes sont utilisées pour garantir l’intégrité des données du produit pendant sa conception et assurer leur conformité lors de sa réalisation.

La conception d’un produit ne peut être considérée comme achevée que si toutes les contraintes qui le définissent ont été satisfaites et que l’intégrité sémantique est conservée [Eastman, 1981].

Selon [Djeraba, 1993] la notion de contrainte est une expression logique qui contribue à maintenir la cohérence dans une base de connaissance.

[Cantzler, 1997] a défini le concept de contrainte comme étant une affirmation ou restriction sur la nature, l'état, ou la valeur d'un objet, ou sur le lien causal de cet objet avec d'autres objets.

Ainsi il est important de tenir compte, lors de la modélisation du modèle de produit et du modèle de processus de conception, des contraintes liées à un produit et à sa démarche de conception.

Ainsi les contraintes jouent plusieurs rôles :

- elles permettent de lier un nombre quelconque de paramètres des composants du produit
- elles permettent d'exprimer des relations entre les différentes étapes du processus
- elles permettent de définir les aspects dynamiques et d'exécution du modèle de processus de conception qui sont de même définis sous forme de contraintes.

D'après les définitions ci dessus, une contrainte peut concerner un produit, un composant de produit, un processus, des étapes de processus, et d'autres objets existants dans la base de connaissances. Les contraintes permettent de garantir la cohérence des éléments.

2) Diagramme de classes

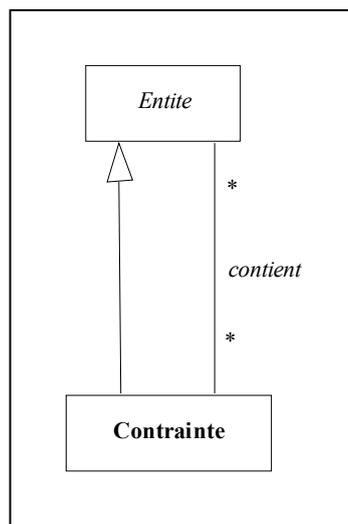


Figure 45 : Diagramme de classes de description de Contraintes.

Une *contrainte* peut concerner plusieurs éléments de la mémoire de projet : produit, processus, etc. Un élément de la mémoire de projet peut être concerné par plusieurs contraintes.

2) Diagramme d'objets

L'objectif du diagramme d'objet de la Figure 47 est de montrer, d'une part, l'exemple du produit « Turbopompe » sachant que la contrainte sur ce produit est que les étapes de sa conception doivent se faire au bureau d'étude Méthodes et Organisation, ceci d'une part et d'autre part une

contrainte sur le document de conception qui doit contenir un nombre de pages inférieur ou égal a 100.

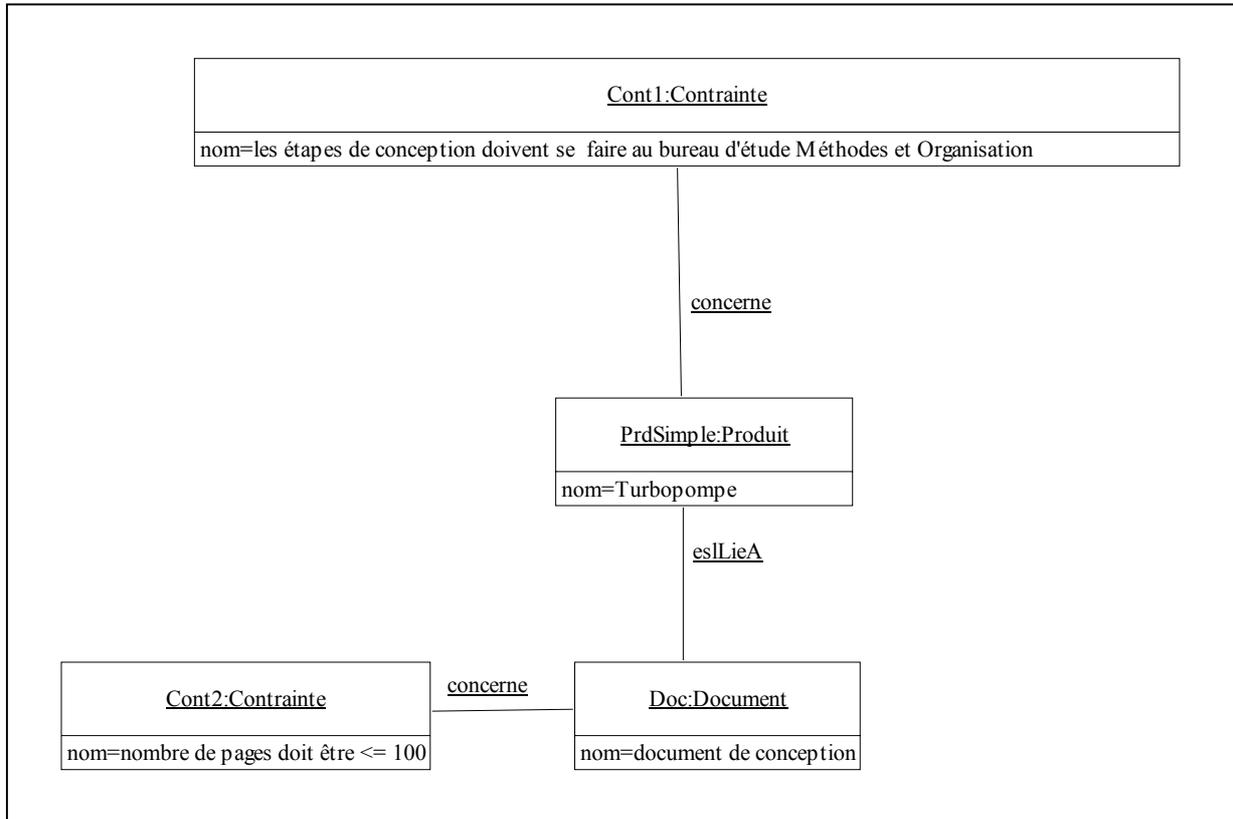


Figure 46 : Diagramme d'objets de description de Contraintes.

L'élément « les étapes de conception doivent se faire au bureau d'étude Méthodes et Organisation » (Cont1) élément du type *Contrainte* est une contrainte qui concerne le produit « Turbopompe » (PrdSimple) élément du type *Produit*. Il est lié à l'élément « document de conception » (Doc) du type *Document*. L'élément doc est concerné par la contrainte « nombre de pages doit être <=100 » (Cont2) élément du type *Contrainte*.

III.3.10. Modèles de Variantes vs Versions

1) Contexte

Un objet dans la mémoire de projet peut avoir plusieurs états au cours de sa vie. Un nouvel état d'un objet est créé suite à un changement de son état actuel. Ce nouvel état est une nouvelle solution issue d'une solution existante.

La discussion qui se pose pour l'instant est ce qu'on utilise dans notre présent travail de mémoire le concept de version ou de variante. Avant de se décider quel terme à utiliser version ou variante, nous allons présenter quelques définitions.

D'après le dictionnaire [Robert, 2000], le concept de variante est défini comme étant une forme ou solution légèrement différente mais voisine.

D'après le dictionnaire [Robert, 2000] le concept de version est défini comme étant chacun des états d'un texte qui a subi des modifications.

Comme on remarque sur l'ensemble de ces deux définitions, le concept de variante est beaucoup plus vaste, il s'agit d'une solution variante dans le temps, tandis que le concept version est attaché à un objet du type texte.

Ainsi une variante d'un produit n'est qu'un résultat de la modification, de la substitution ou du déplacement d'éléments d'un produit conçu au préalable. Les variantes historiques se suivent dans le temps. Plusieurs variantes peuvent être conçues simultanément. Elles découlent d'un même raisonnement. Elles sont les solutions d'une modélisation commune car la variante est un élément qui ne se crée que lorsque les éléments qui le composent existent.

On pourra dire que la variante est un ensemble de modèles différents d'une même famille à un moment donnée. Exemple la Clio 1.2, la Clio D et la Clio 16V tandis que la version est une évolution dans le temps d'un modèle donné exemple la Clio 2 est une nouvelle version de la Clio. En résumé on peut dire que la notion de variante est une notion importante dans une mémoire de projet qui permet de suivre l'évolution des éléments.

2) Diagramme de classes

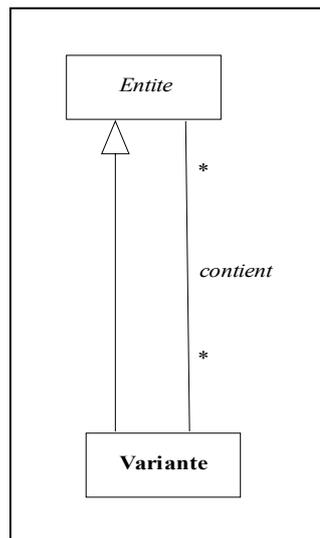


Figure 47 : Diagramme de classes de description des Variantes.

Une *variante* est un élément de la mémoire de projet. Elle peut contenir plusieurs éléments de la mémoire de projet. Un élément de la mémoire de projet peut intervenir dans la composition de plusieurs variantes.

2) Diagramme d'objets

L'objectif du diagramme d'objets de la Figure 49 (resp. 50) est de montrer la structure d'une variante v1012 (resp. v1013).

Exemple 1 :

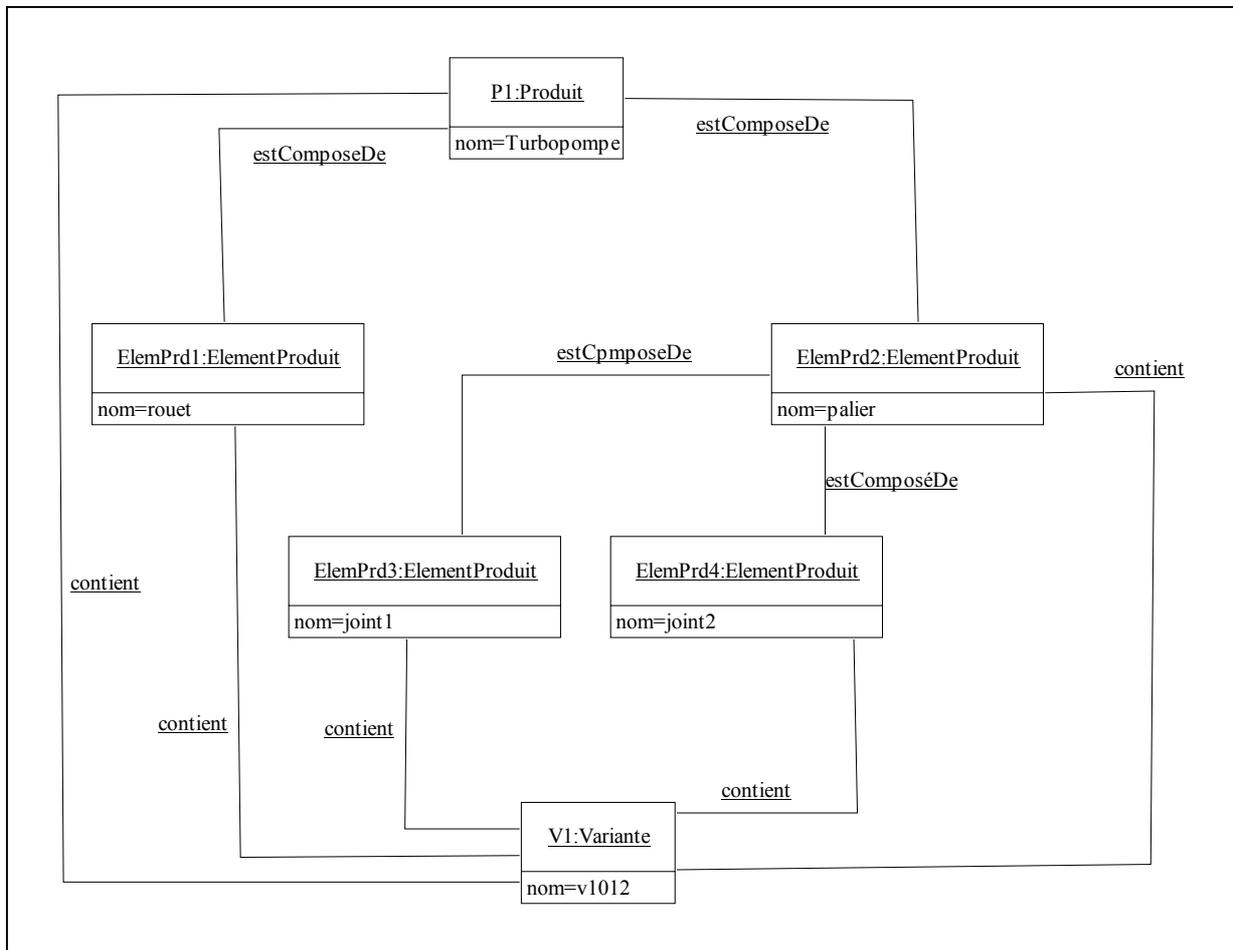


Figure 48 : Diagramme d'objets de description de Variantes.

L'élément « v1012 » (V1) du type *Variante* contient un élément « Turbopompe » (P1) du type *Produit* qui est composé de deux éléments « rouet » (ElemPrd1) et « palier » (ElemPrd2) tous les deux éléments du type *ElementDeProduit*. L'élément ElemPrd2 est composé de deux éléments « joint1 » (ElemPrd3) et « joint2 » (ElemPrd4) éléments du type *ElementProduit*. Tous ces éléments sont contenus dans la variante V1.

Exemple 2

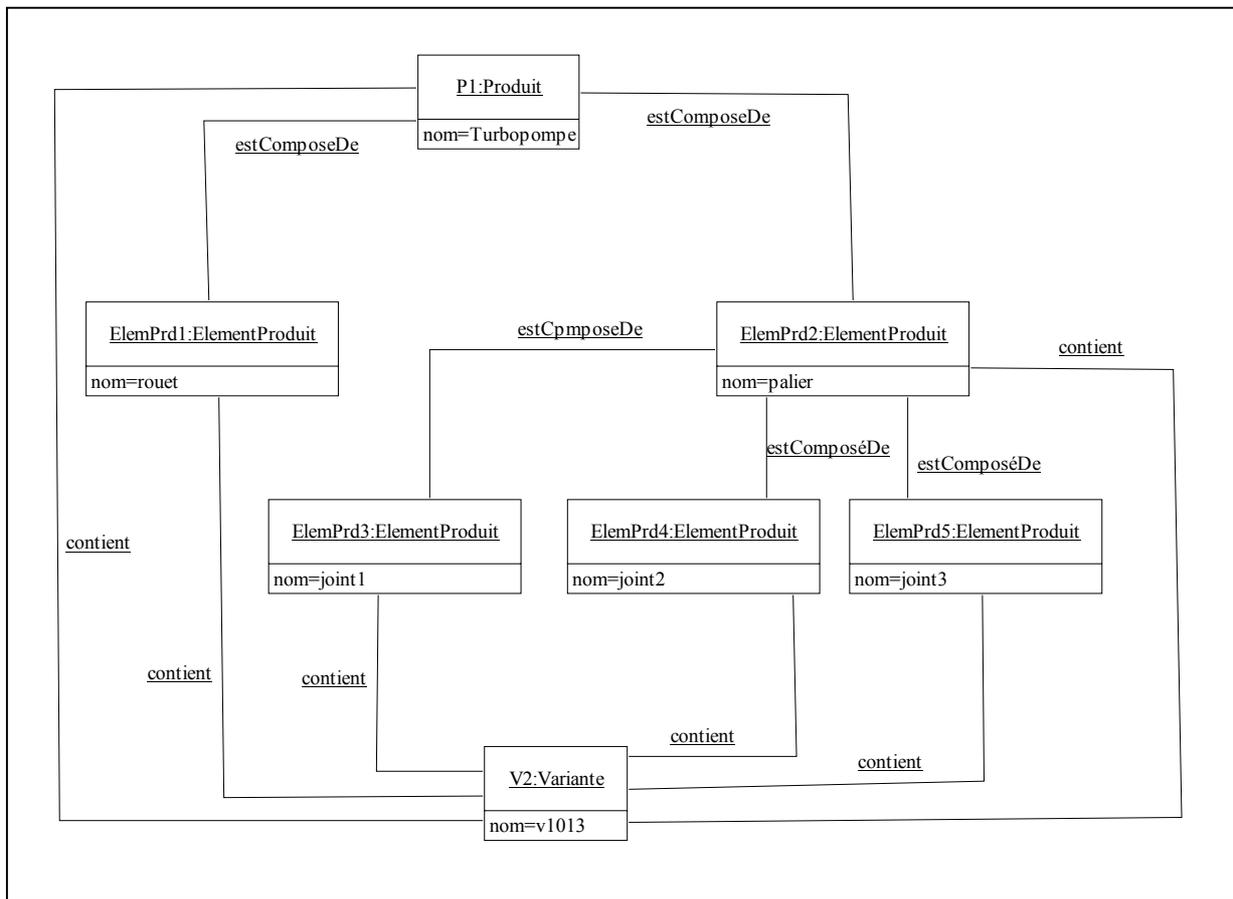


Figure 49 : Diagramme d'objets de description de Variante.

La Figure 50 présente une modification dans la structure du produit « Turbopompe » de la Figure 49 et ceci en ajoutant dans un élément « joint3 » (ElemPrd5) du type *ÉlémentProduit*. De ce fait une nouvelle variante « v1013 » (V2) du type *Variante* est constituée. Ainsi la nouvelle variante « v1013 » contiendra les éléments « Turbopompe », « rouet », « palier », « joint1 », « joint2 », et « joint3 ». On remarque que le lien entre les variantes n'existe pas encore, point qui sera présenté plus loin.

III.3.11. Le Modèle des Relations «Evoluées» entre les différents éléments de la mémoire de projet

1) Contexte

Comme nous l'avons déjà vu, la relation « *estLiéA* » se limite à relier deux éléments de la mémoire de projet n'appartenant pas aux mêmes paquetages. Dans ce qui suit on présentera d'autres types de relations basées sur des aspects fonctionnels, structurels et temporels.

2) Diagramme de classes

L'objectif du diagramme de classes de la Figure 51 est de proposer au concepteur des relations sémantiquement plus riches que la relation « *estLiéA* » afin de pouvoir lier les éléments de la mémoire de projet entre eux. Les différents modèles seront présentés progressivement.

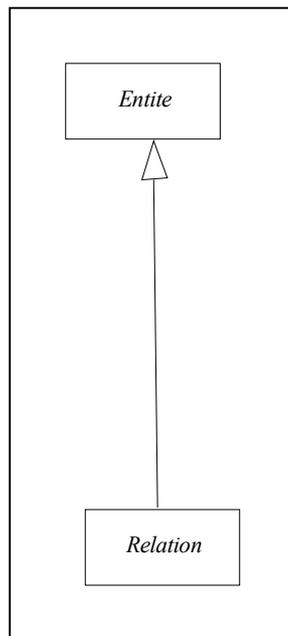


Figure 50 : Diagramme de classes de description de Relations.

Un élément de la mémoire de projet peut être un élément du type *Relation*. Un élément du type *Relation* peut être un élément de la mémoire de projet. La Figure 52 présente en détail les différents types de relations dont on aura besoin, d'une part pour lier les différents éléments de la mémoire de projet, et d'autre part, pour pouvoir gérer l'historique des éléments.

3) Diagramme de classes de raffinement des relations

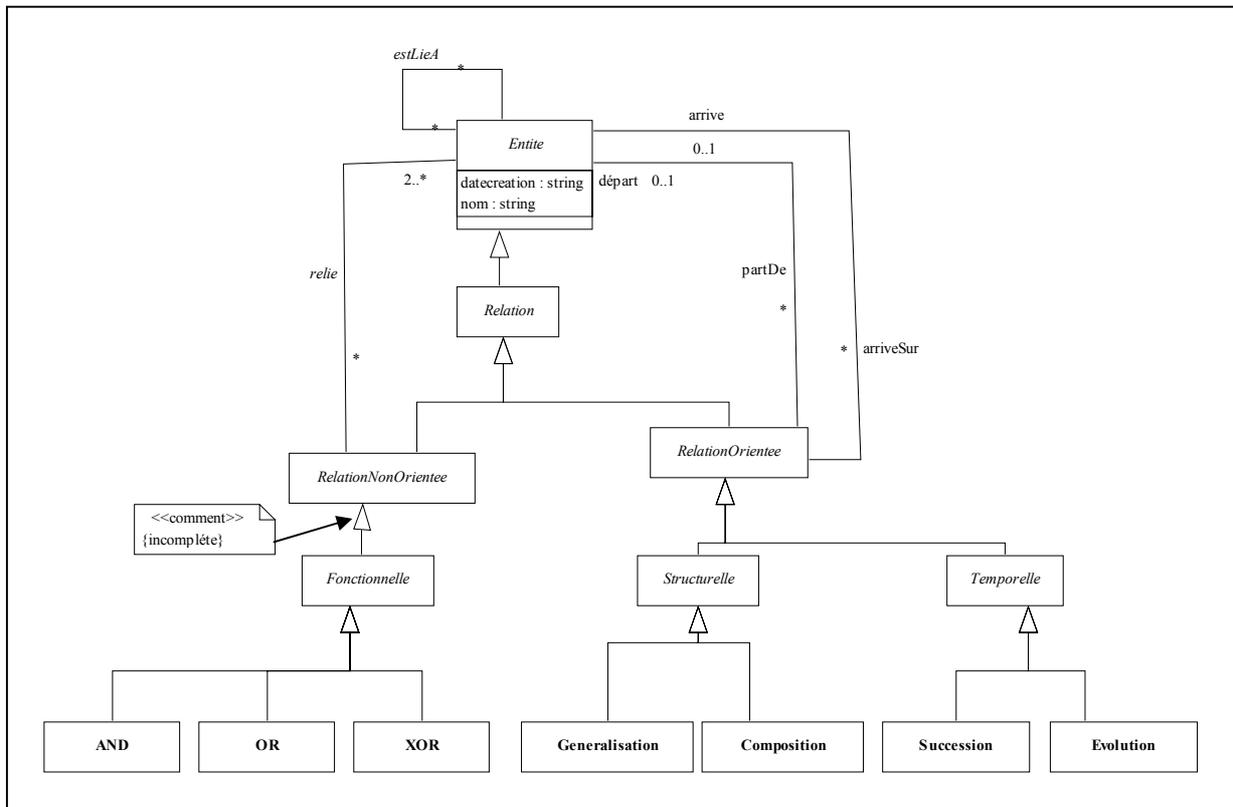


Figure 51 : Diagramme de classes de description des différents types de relations.

La Figure 52 présente les différents types de relation qu'on peut avoir dans notre mémoire de projet. Ces relations servent à construire les différents liens entre les éléments de la mémoire de projet. Toutes les relations sont des spécialisations de la classe abstraite *Entité*. Ces relations sont classées en deux catégories à savoir les relations orientées et les relations non orientées.

3.1) Les relations orientées : Une relation du type orientée permet de relier deux ou plusieurs éléments de la mémoire de projet avec un sens de lecture. Il existe deux types de relations orientées à savoir les relations structurelles et les relations temporelles.

a) Les relations structurelles : On distingue deux types de relations

- Les relations de généralisation : Ce sont des relations de factorisation des différents éléments de la mémoire de projet.
- Les relations de composition : Elles permettent de traduire une relation composant-composite. Ainsi dans la relation du type Composition fournie par le langage UML il n'est pas possible d'ajouter un attribut permettant de gérer, par exemple la date de création de la composition d'où l'intérêt de construire de telle

relation. La relation PartDe a comme origine un élément composite et comme arrivée un élément composant. Inversement la relation du type ArriveSur a comme origine un élément composite et comme arrivée un élément composant.

b) Les relations temporelles : ce sont des relations où l'aspect temps intervient dans la relation entre les éléments de la mémoire de projet intervient. On distingue deux types de relations temporelles.

- Les relations de succession : La chronologie entre les nœuds d'une arborescence peut également être représentée (classe *Succession*), pour indiquer par exemple l'ordre des phases d'un processus.
- Les relations d'évolution : La classe *Evolution* est destinée à gérer l'historique des évolutions, par exemple pour garder en mémoire qu'une personne a été responsable d'un lot de travail jusqu'à une date donnée avant de confier cette responsabilité à un autre acteur.

3.2) Les relations non orientées : Ce sont les relations qui représentent des aspects fonctionnels entre les différents éléments de la mémoire de projet. La relation « Relie » permet de relier au moins deux éléments de la mémoire de projet par l'intermédiaire d'une des relations du type AND, OR ou XOR. Comme on peut le remarquer sur la figure 27, il existe une note au sens d'UML au niveau des relations du type fonctionnelle « incomplète » exprimant le fait que d'autres relations peuvent venir enrichir le modèle, celui-ci n'étant pas exhaustif.

- Les relations du type AND : elles permettent d'effectuer une conjonction logique sur deux éléments de la mémoire de projet.
- Les relations du type OR : elles permettent d'effectuer une disjonction logique sur deux éléments de la mémoire de projet.
- Les relations du type XOR : elles permettent d'effectuer une opération d'exclusion logique sur deux éléments de la mémoire de projet.

4) Choix des relations

A travers ce paragraphe nous allons présenter un outil d'aide au concepteur lui permettant de faire le choix des éléments de la mémoire de projet a relier et le type de relation a mettre en place entre ces éléments. Ainsi trois cas de figure se présentent :

1 cas : La relation du type « *estLiéA* » est une relation permettant de relier les éléments entre eux sans gérer l'historique de ces éléments.

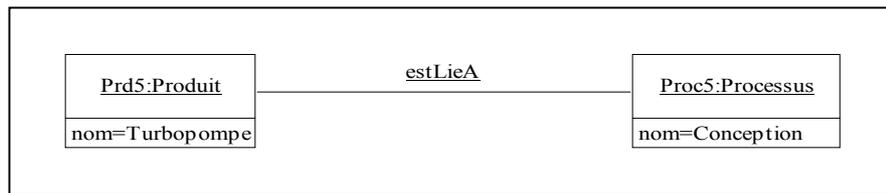


Figure 52 : Lien entre deux éléments n'appartenant au même paquetage sans historique

Un élément « Turbopompe » (Prd5) du type *Produit* est lié à un élément « Conception » (Proc5) du type *Processus* par la relation « *estLiéA* ».

2 cas : Le concepteur souhaite relier deux éléments de la mémoire de projet sans en gérer l'historique. Dans ce cas, on pourra utiliser la relation « *estComposéDe* » à sémantique plus riche entre deux classes indiquant un lien d'appartenance.

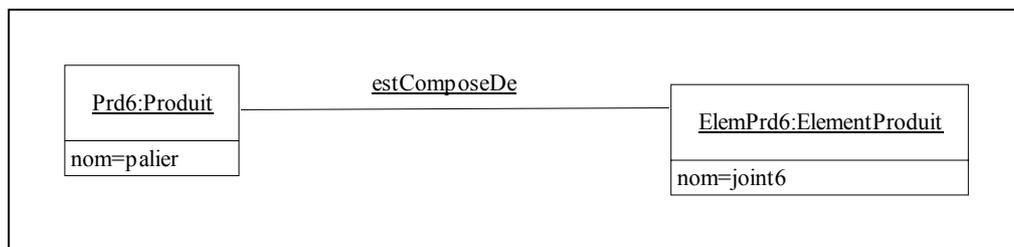


Figure 53 : Lien entre deux éléments du même paquetage sans historique.

Un élément « palier » (Prd6) du type *Produit* est composé d'un élément « joint6 » (ElemPrd6) du type *ElementProduit* par la relation du type « *estComposéDe* ».

3 cas : Le concepteur souhaite gérer l'historique d'un élément de la mémoire de projet. Pour se faire, on pourra utiliser la relation du type « *Evolution* ». Deux cas de figure peuvent se présenter :

a- Les éléments appartiennent au même paquetage

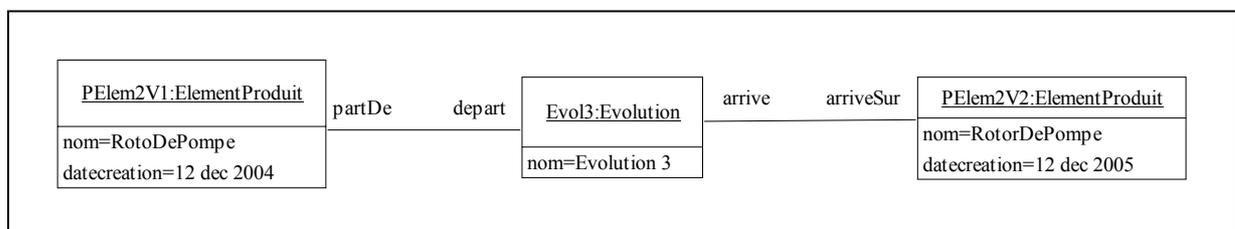


Figure 54 : Lien entre deux éléments du même paquetage avec historique.

L'élément « RotorDePompe » (PElem2V2) est du type *ElementProduit*. Par exemple cet élément a été modifié dans ses caractéristiques afin d'avoir une nouvelle variante de ce même produit « RotorDePompe » (PElem2V1 élément du type *ElementProduit*. La relation « Evolution 3 » (E3) du type *Evolution* a permis de relier les deux éléments entre eux en utilisant les sémantiques des relations « partDe arriveSur » ou « départ Arrive ». Ceci permet de tracer ces éléments grâce à la relation Evol3.

b- Evolution d'une instance d'une classe A vers une classe B

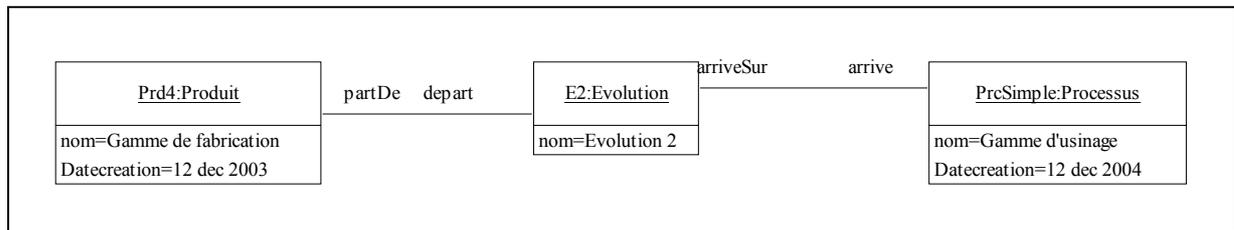


Figure 55 : Lien entre éléments qui ne sont pas du même paquetage avec historique.

L'élément « Gamme de fabrication » (Prd4) du type *Produit* lors de son passage à l'usage devient un élément « Gamme d'usinage » (PrcSimple) du type *Processus*. Ainsi le produit a subi une évolution d'une phase de produit à une phase où est il est devenu processus. Cette évolution est matérialisée par l'élément « Evolution 2 » (E2) du type *Evolution*.

5) Diagramme d'objets

L'ensemble des diagrammes d'objets suivants présentent des applications du diagramme de classes de la Figure 52.

Exemple 1 : Application à la relation de type succession

L'objectif du diagramme d'objet suivant est de montrer une application de la relation du type Succession. Il sert à décrire des étapes d'exécution d'un processus.

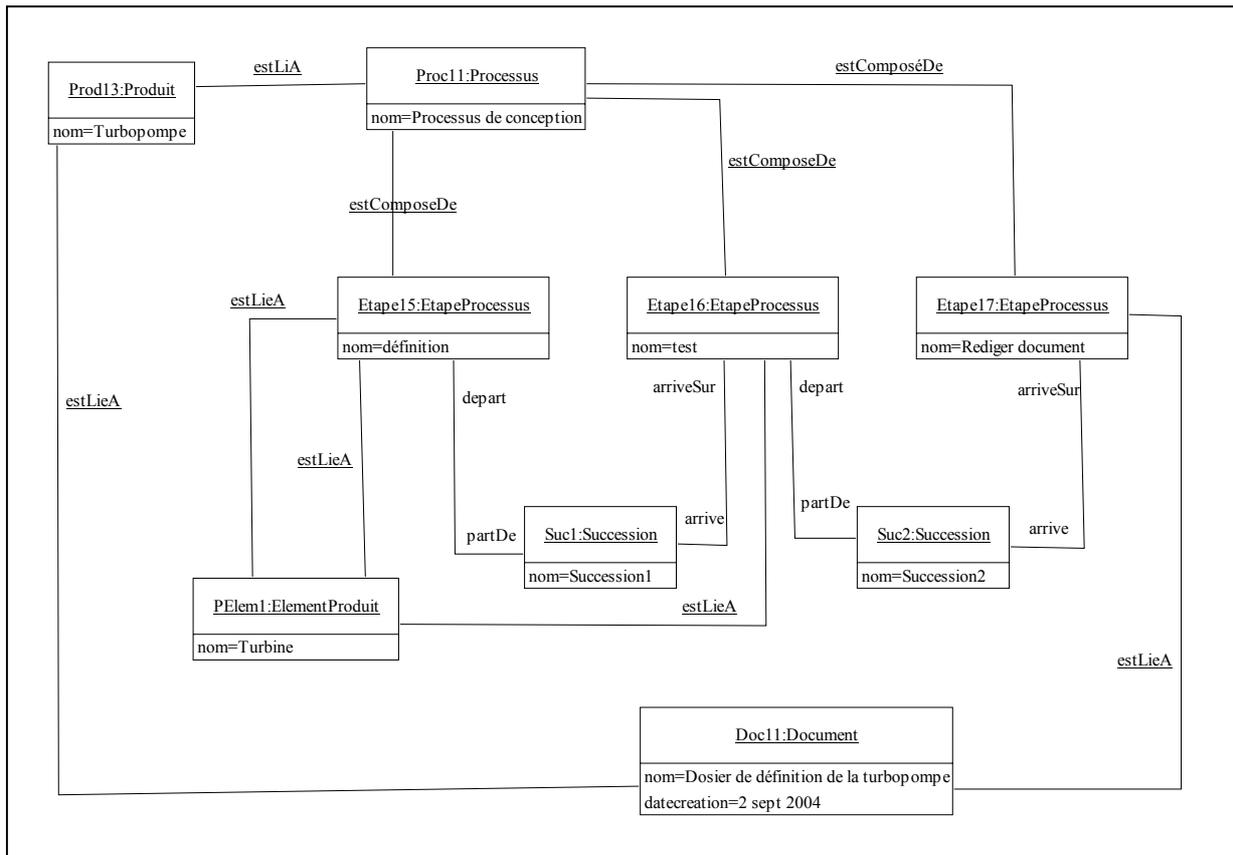


Figure 56 : Diagramme d'objets UML de description de la relation du type Succession.

Le produit « TurboPompe » (Prod13) du type *Produit* est lié au processus « ProcesusDeConception » (Proc11) du type *Processus* par la relation « estLiéA », à l'objet « définition » (Etape15) élément du type *EtapeProcessus* et à l'élément « Dossier de définition de la turbopompe à oxygène » (Doc11) élément du type *Document*. L'élément « définition » (Etape15) est suivi de l'élément « test » (Etape16) élément du type *EtapeProcessus*. La relation de succession est réalisée par l'élément « Succession1 » (Suc1) élément du type *Succession*. De même l'élément « test » (Etape16) est suivi de l'élément « redigerDoc » (Etape17) élément du type *EtapeProcessus* et ceci est réalisée par la relation « Succession » (Suc2) élément du type *Succession*.

Exemple 2 : Application à la relation de type composition

L'objectif du diagramme d'objet suivant est de montrer une application de la relation du type Composition. Il sert à décrire la composition du produit « Turbopompe ».

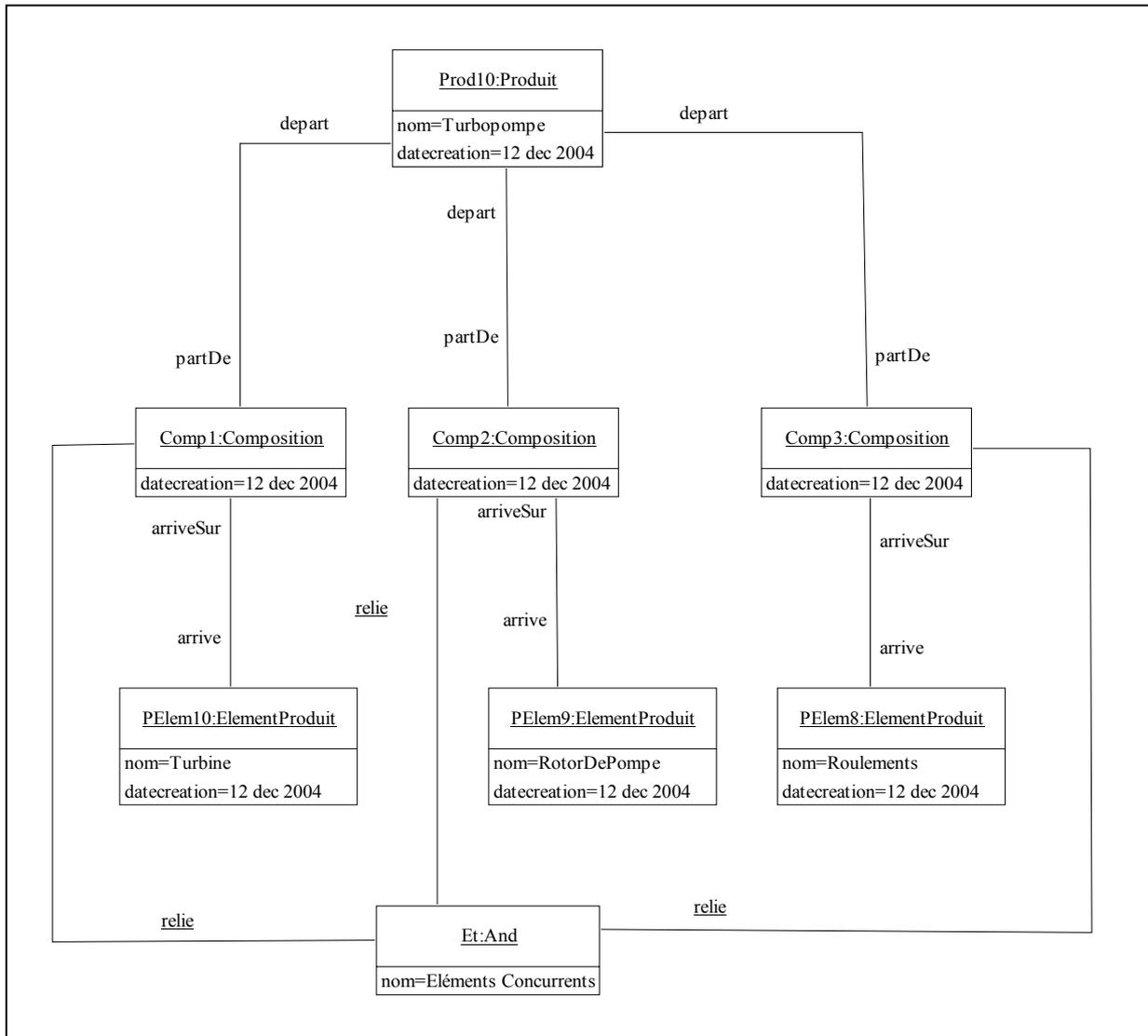


Figure 57 : Diagramme d'objets de description de la relation du type Composition.

L'élément « TurboPompe » (Prod10) du type *Produit* est composé de « Turbine » (PElem10), « RotorDePompe » (PElem9), et « Roulements » (PElem8), éléments du type *ElementProduit*. Les relations Comp1, Comp2, et Comp3, sont des éléments du type *Composition* permettent d'indiquer les éléments composant de l'élément « TurboPompe » (Prod10). Les relations binaires du type « partDe ... arriveSur » et « départ ... arrive » indiquent les rôles que peuvent jouer ces relations. L'attribut date de création dans les relations du type *Composition* permet d'indiquer un historique sur la date de création de la composition d'un élément du type *Produit*.

Exemple 3 : Application à la relation de type généralisation

L'objectif du diagramme d'objet suivant est de montrer l'utilisation de la relation du type généralisation. Cette relation sert à décrire la généralisation d'un élément de produit par rapport à un produit.

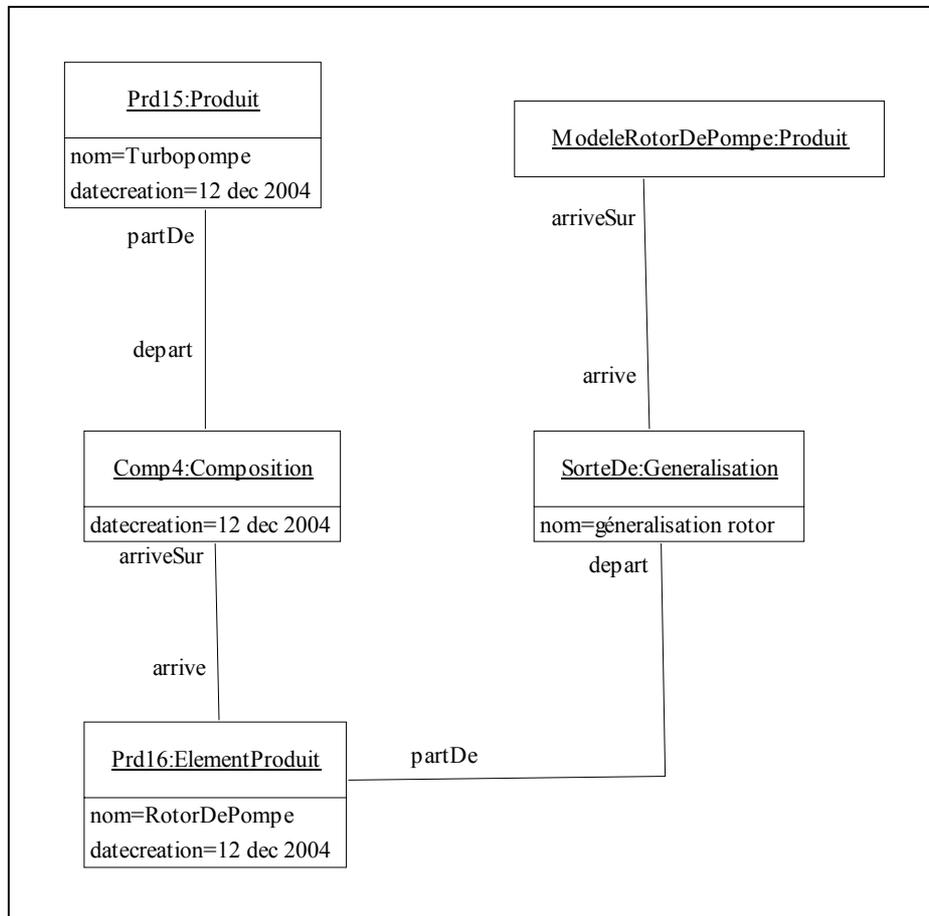


Figure 58 : Diagramme d'objets de description de la relation du type Généralisation.

Le produit « TurboPompe » (Prod15) du type *Produit* est composé d'un élément « RotorPompe » (Prd16) du type *ElementProduit*. Cette composition est matérialisée par la relation Comp4 du type *Composition*. Elle sert aussi à définir l'élément composant et l'élément composite. L'élément « RotorDePompe » (Prd16) est en quelque sorte un élément générique de l'élément ModeleRotorDePompe du type *Produit*. Il s'agit d'une relation appelée SorteDe élément de type *Généralisation*.

On souhaite mémoriser l'historique d'évolution des éléments de la mémoire de projet. A travers l'ensemble des exemples suivants on va montrer que le mécanisme d'évolution permet d'assurer la traçabilité de tous les objets de la mémoire de projet.

Pour ce faire considérons les deux exemples suivants à savoir l'exemple 4 et l'exemple 5. Chacun d'eux représente une variante composée d'un certain nombre d'éléments de la mémoire de projet.

La variante 2 représentant l'exemple 5 est issue de la variante 4 représentant l'exemple 4 et ceci suite à une modification de la date de création d'un élément de la variante 1.

Exemple 4 : Application à la relation de type Composition-Variante : Variante 1

La Figure 60 est un diagramme d'objets qui montre qu'une variante regroupe tous les objets qui décrivent une configuration donnée. Elle comporte à la fois le produit, ses constituants et les relations qui les relient. Les relations sont du type composition.

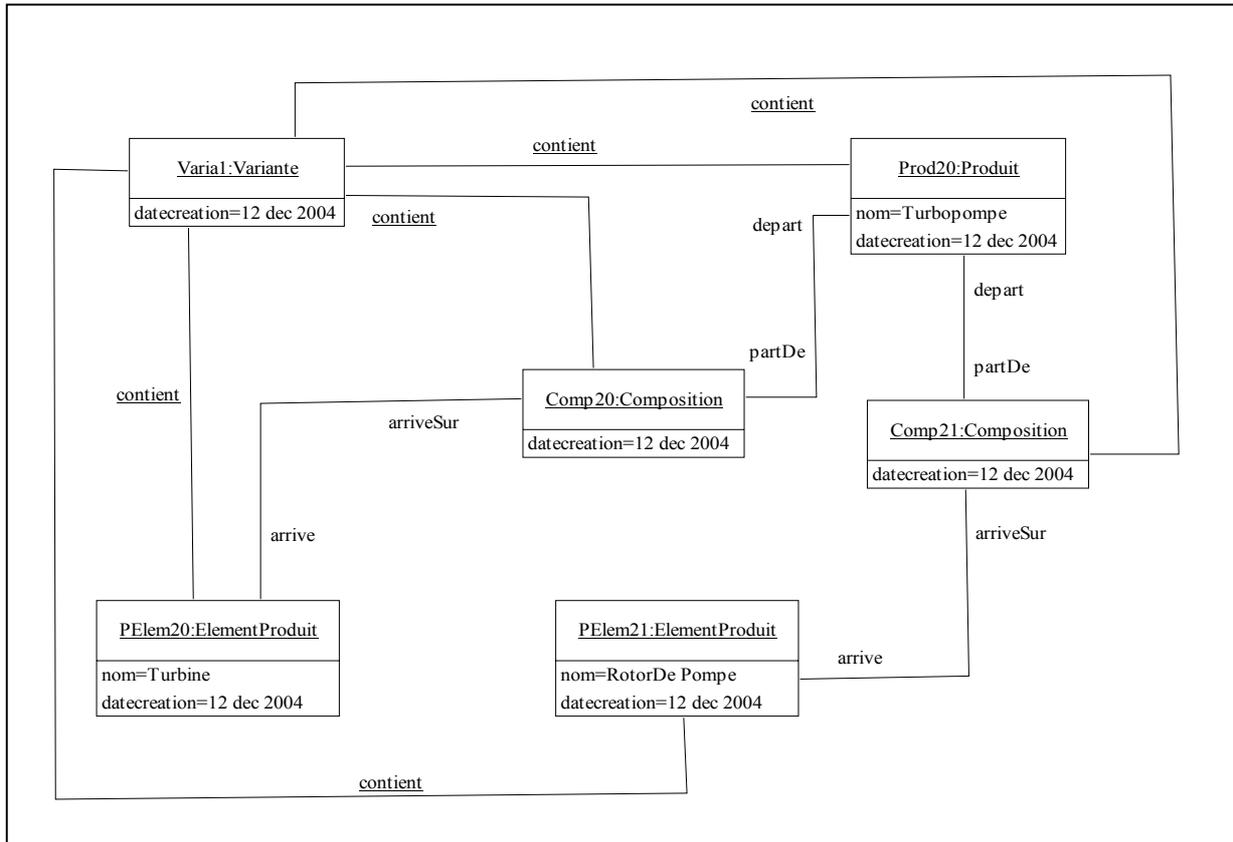


Figure 59 : Diagramme d'objets de description de la relation du type Composition-Variante.

L'élément « Turbopompe » (Prod20) du type *Produit* est composé des deux éléments « Turbine » (PElem20) et « RotorDePompe » (PElem21) éléments du type *ElementProduit*. Ces deux compositions sont matérialisées par les deux relations Comp20 et Comp21 éléments du type *Composition*. L'élément « Varial » élément du type *Variante* contient les éléments Prod20, PElem20, PElem21, Comp20 et Comp21.

Exemple 5 : Application à la relation de type Composition-Variante : Variante 2

Une nouvelle variante du produit Prod20 de l'exemple 4 précédent est créée et ceci suite à une modification dans la date de création d'un de ses constituants. D'où l'on a une nouvelle variante Varia2 du type *Variante*.

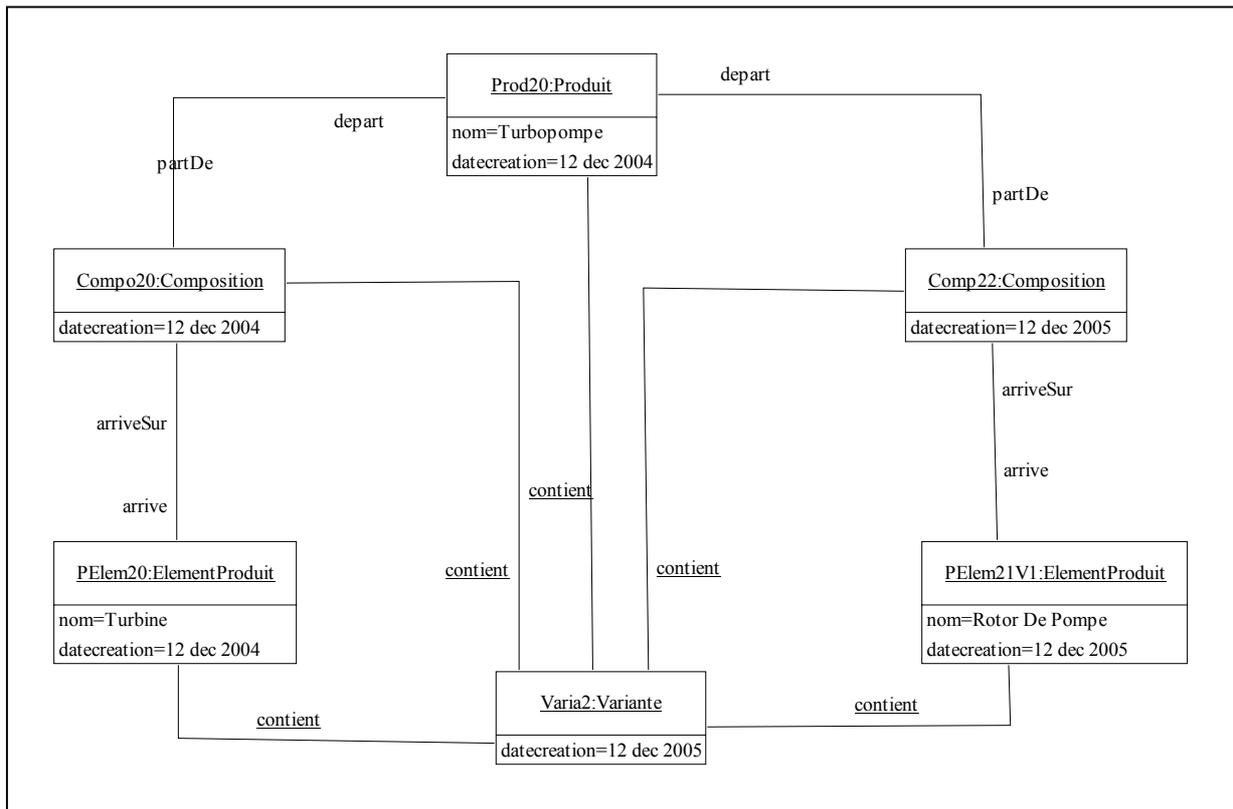


Figure 60 : Diagramme d'objets de description de la relation du type Composition-Variante.

Exemple 6 : Application à la relation de type Composition-Evolution

L'objectif du diagramme d'objets suivant est de montrer qu'on est capable de tracer l'évolution entre deux éléments par exemple du type *ElementProduit* de la mémoire de projet. Cet exemple se base pour sa construction sur les deux exemples précédents.

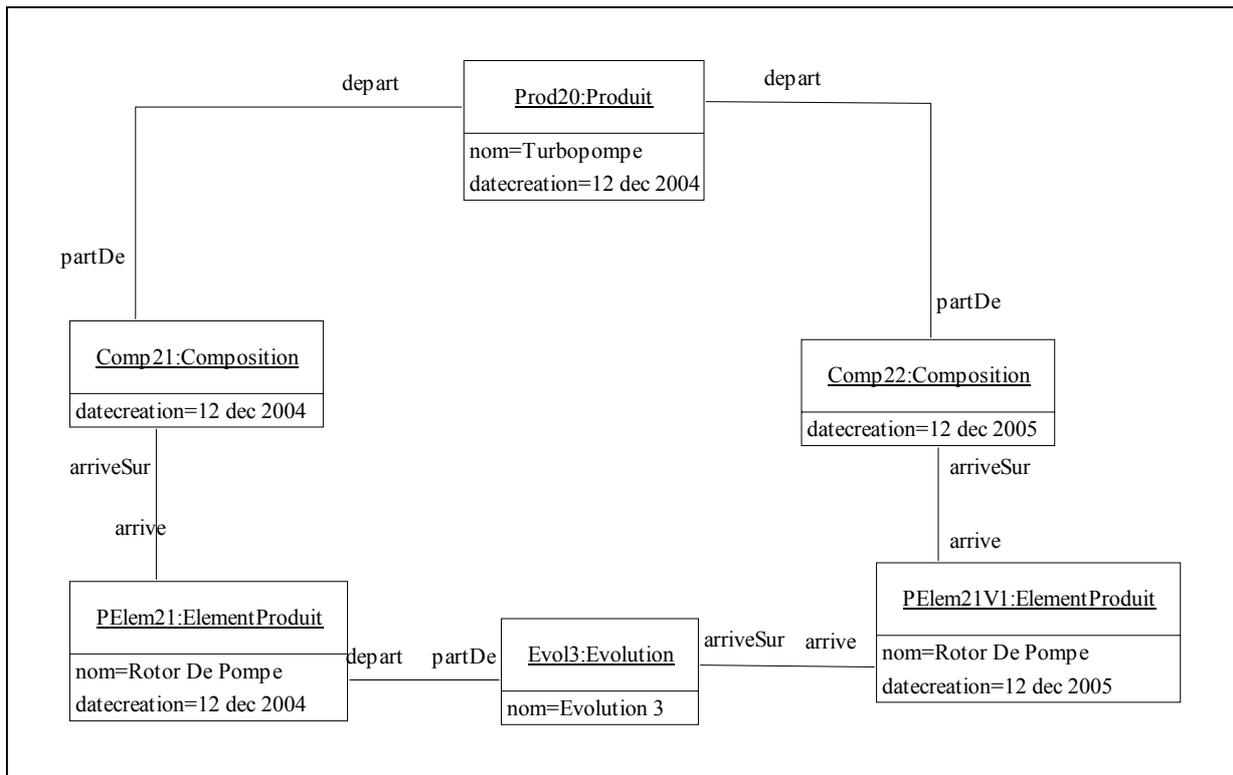


Figure 61 : Diagramme d'objets de description de la relation du type Composition-Evolution.

Le lien entre les deux éléments PElem21 et PElem21V1 est réalisé par la relation « Evolution3 » (Evol3) du type *Evolution*. Il s'agit d'un lien qui matérialise l'historique de l'évolution entre ces deux éléments.

Exemple 7 : Application à la relation de type Variante-Evolution

L'objectif du diagramme d'objet suivant est de montrer l'évolution entre les deux variantes de l'exemple 4 et l'exemple 5 précédent.

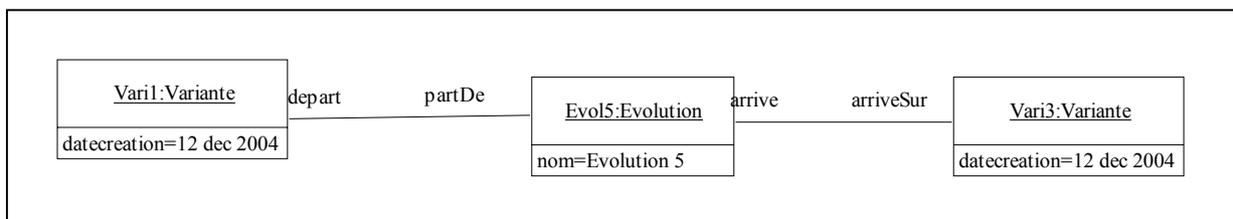


Figure 62 : Diagramme de classes de description de type Variante-Evolution.

Exemple 8 : Application à la relation de type Composition/Xor

L'objectif du diagramme d'objets suivant est de montrer l'utilisation de la relation du type Xor. Elle sert à appliquer une exclusion entre les deux compositions d'un produit « Turbopompe ».

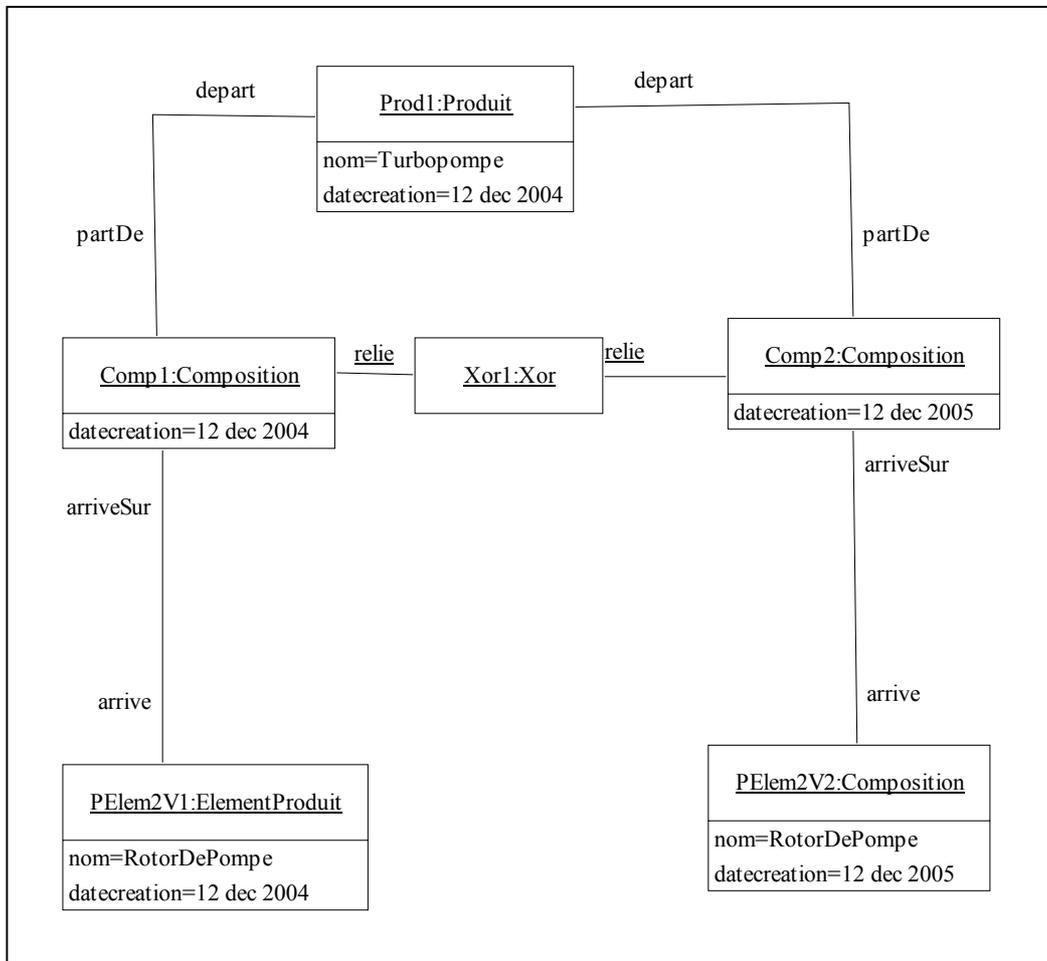


Figure 63 : Diagramme d'objets de description de type Composition-Xor.

L'élément « Turbopompe » (Prod1) est composé de l'élément « RotorDePompe » (PElem2V1) élément du type *ElementProduit*. La relation de composition est matérialisée par l'élément Comp1 du type *Composition*. L'élément « Turbopompe » (Prod1) est composé de l'élément « RotorDePompe » (PElem2V2) élément du type *ElementProduit*. La relation de composition est matérialisée par l'élément Comp2 du type *Composition*. Le lien entre les deux éléments Comp1 et Comp2 matérialisé par un lien Xor1 du type *XOR* exprimant le choix de composition selon une condition entre ces éléments. Ainsi à un instant donné le produit « Turbopompe » va être constitué de l'élément de produit « RotorDePompe » ayant comme date de création « 12 dec 2004 » ou de l'élément de produit « RotorDePompe » ayant comme date de création « 12 dec 2005 » mais pas les deux au même temps.

Exemple 9 : Application à la relation de type Composition-And

L'objectif du diagramme d'objets suivant est de montrer l'utilisation de la relation du type And. Elle sert à appliquer une conjonction entre les éléments de la mémoire de projet. Cette conjonction est faite entre les compositions d'un produit « Turbopompe ».

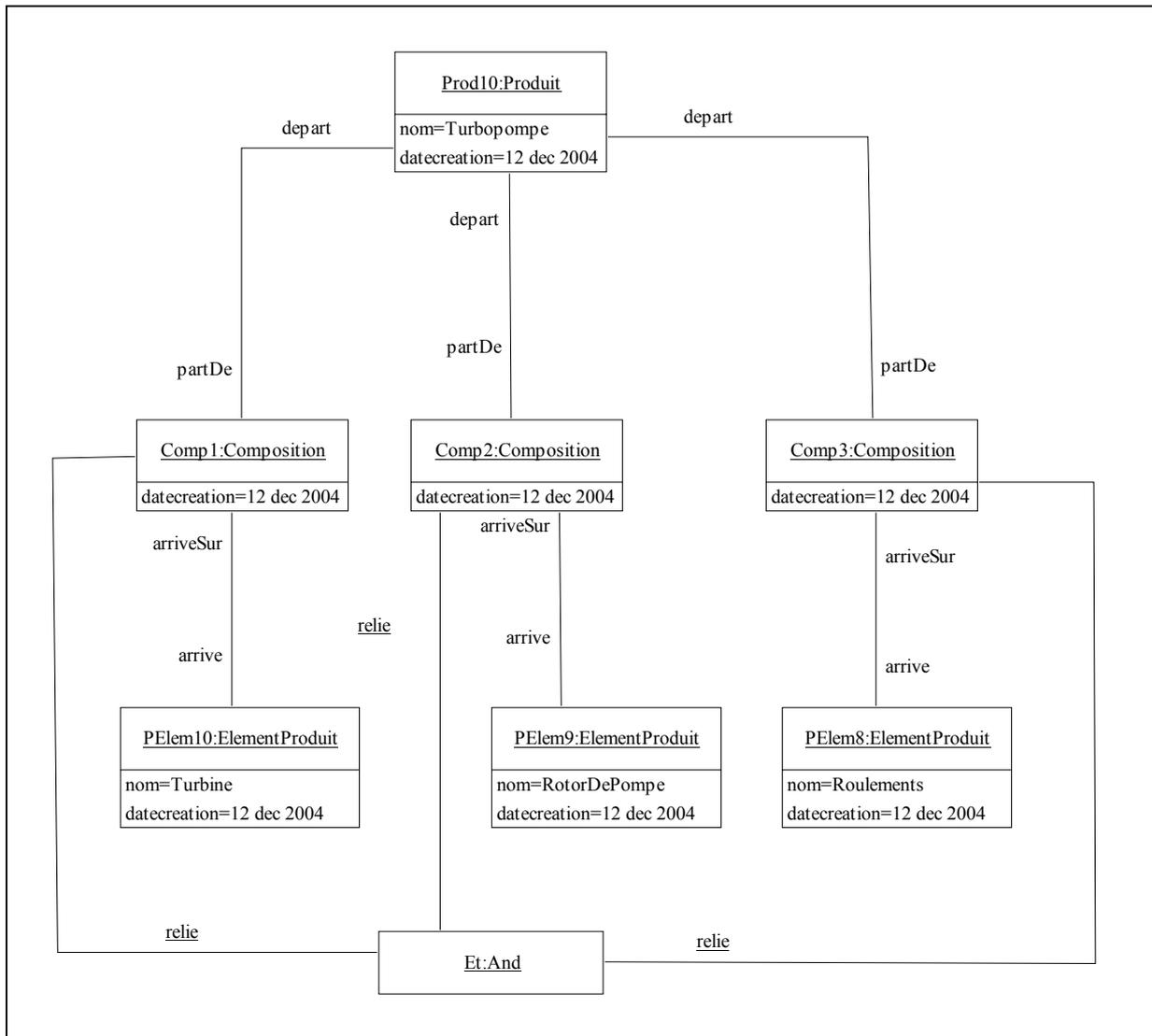


Figure 64 : Diagramme d'objets de description de type Composition-And.

L'élément « TurboPompe » (Prod10) du type *Produit* est composé de « Turbine » (PElem10), « RotorDePompe » (PElem9), et « Roulements » (PElem8), éléments du type *ElementProduit*. Les relations Comp1, Comp2, et Comp3 sont des éléments du type *Composition*. Elles permettent de décrire les éléments constitutifs de l'élément « TurboPompe » (Prod10). Les relations binaires du type « partDe ... arriveSur » et « départ ... arrive » sont des relations qui indiquent le

composant et le composite. L'attribut date dans la relation du type Composition permet d'indiquer un historique sur la date de création de la composition d'un élément du type *Produit*. L'élément And1 du type *And* permet de faire coexister ensemble les éléments PElem10, PElem9 et PElem8 éléments du type chacun *ElementProduit* afin de donner une existence à l'élément Prod10 du type *Produit*. Ce qui permet de garantir à un instant donné l'existence des éléments Comp1, Comp2, et Comp4 de type Composition.

III.3.12. Le Modèle des Ressources

1) Contexte

Définitions du concept de Ressource :

[Labrousse, 2004] a proposé une définition du concept de ressource comme étant un élément contribuant au processus sans en être l'objet.

Dans le cas des ressources humaines et d'après [Gzara, 2000] un acteur est un profil particulier de personnes physiques dans l'entreprise. Un profil est défini par l'association d'une personne physique à un groupe de personnes appartenant au même métier de l'entreprise et à un niveau hiérarchique dans ce groupe.

On peut dire que la notion de ressource et d'acteur peuvent concerner les éléments de la mémoire de projet. C'est pour cela qu'on s'intéresse à leur modélisation comme étant des éléments de la mémoire de projet.

Cité par [Abdelouleh, 2004] et d'après [Hachette, 2001] « un acteur est un individu, un groupe ou une institution, considéré du point de vue des rôles qui lui sont impartis dans la société (dans notre cas, c'est l'entreprise). C'est aussi une personne qui joue un rôle dans une action ».

Définitions du concept Rôle :

D'après [Hermosillo et al, 2002] le concept de rôle fait référence globalement à un ensemble de comportements mis en œuvre par un individu relativement à sa position dans l'organisation.

[Gardarin, 2003] a défini le concept de rôle comme étant un ensemble de droits sur les objets caractérisé par un nom, pouvant intervenir comme sujet dans le mécanisme d'autorisation.

[Heurtel, 2005] définit le concept de rôle comme étant un regroupement nommé de privilèges (systèmes et objets) qui peut être attribué en tant que tel à un utilisateur ; cet utilisateur reçoit alors automatiquement les privilèges contenus dans le rôle.

D'après le dictionnaire [Robert, 2000] le privilège est le droit, avantage particulier possédé par quelqu'un et que les autres n'ont pas.

D'après [Heurtel, 2005] et dans une base de données Oracle, les droits des utilisateurs sont gérés avec la notion de privilège. Il s'agit du droit:

- d'exécuter un ordre SQL en général (par exemple, créer une table) : c'est la notion de privilège système ;
- d'accéder à un objet d'un autre utilisateur (par exemple, mettre à jour les données de la table CLIENT) : c'est la notion de privilège objet. Les privilèges peuvent être attribués directement aux utilisateurs ou par l'intermédiaire de rôles.

En résumé on peut dire que la notion de rôle est un concept qui nous intéresse afin de délimiter les rôles que peuvent avoir un acteur dans une mémoire de projet. C'est pour cela qu'on s'intéresse à sa modélisation comme étant des éléments de la mémoire de projet.

D'après ces définitions un acteur est caractérisé par un rôle, et d'après [Gardarin, 2003] de tel rôle a des droits c'est-à-dire des privilèges.

Définitions du concept de Compétence :

D'après [GDT, 2004] la notion de compétence est un ensemble de savoirs, de savoir-faire et de savoir-être requis pour exécuter adéquatement certaines tâches ou réussir dans l'exercice d'une fonction, et qui peuvent être mis en œuvre sans apprentissage nouveau.

[Labrousse, 2004] définit la notion de compétence comme étant la capacité à exploiter des connaissances et des ressources pour exercer une activité dans un contexte contraint donné et atteindre un objectif.

D'après ces définitions on peut remarquer que la compétence est associée à des connaissances, à des acteurs et à des ressources. De ce fait, on s'intéresse à sa modélisation comme étant un élément de la mémoire de projet afin de mener à bien un projet.

2) Diagramme de classes

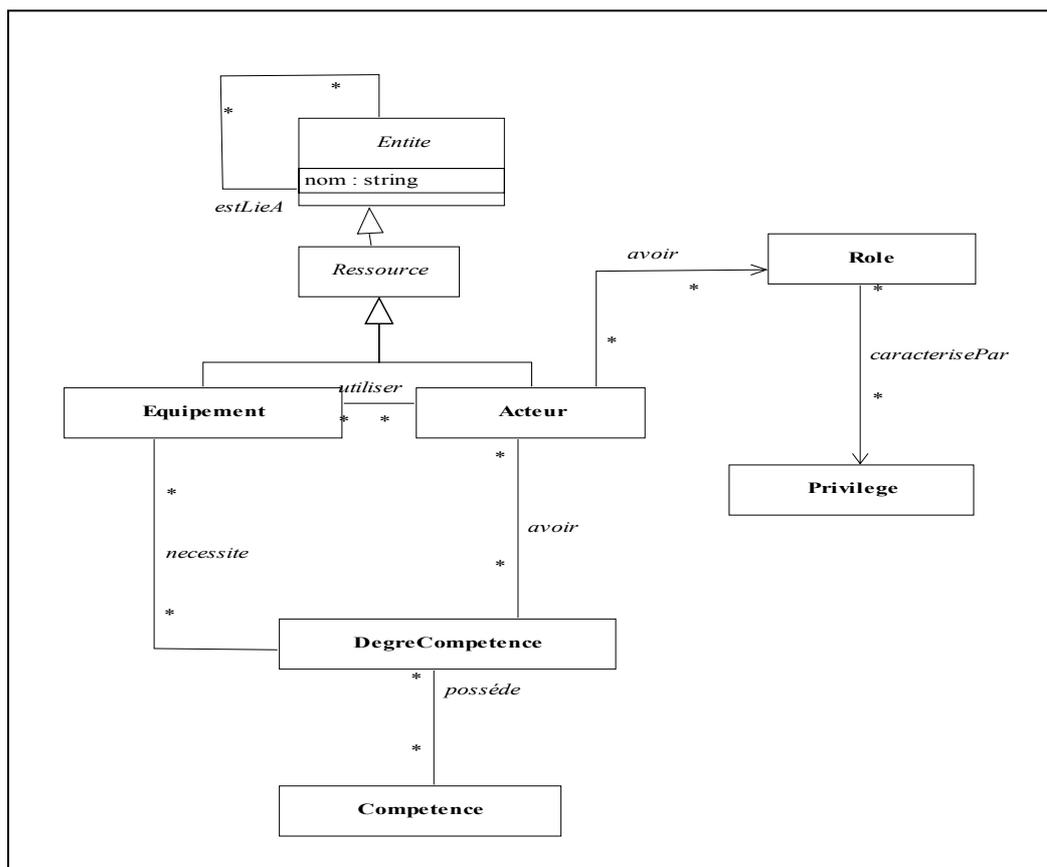


Figure 65 : Diagramme de classes de Ressource.

Un élément de la mémoire de projet peut être du type *Ressource*. Cet élément peut être du type *Equipement* ou du type *Acteur*. Un élément du type *Acteur* peut utiliser plusieurs éléments du type *Equipement*. Un élément du type *Acteur* peut avoir certains éléments du type *Rôle*. Nous nous sommes inspirés des bases de données pour définir les rôles qui sont caractérisés par des éléments du type *Privilege*.

Un élément du type *Equipement* nécessite des éléments du type *DegréCompétence*. Ces degrés de compétence sont relatifs à un élément du type *Compétence*.

3) Diagramme d'objets

L'objectif du diagramme d'objets suivant est de montrer sur un équipement du type « Outil CAO mécanique » l'ensemble des acteurs qui l'utilisent, le rôle, le privilège et la compétence et le degré de compétence que chacun possède.

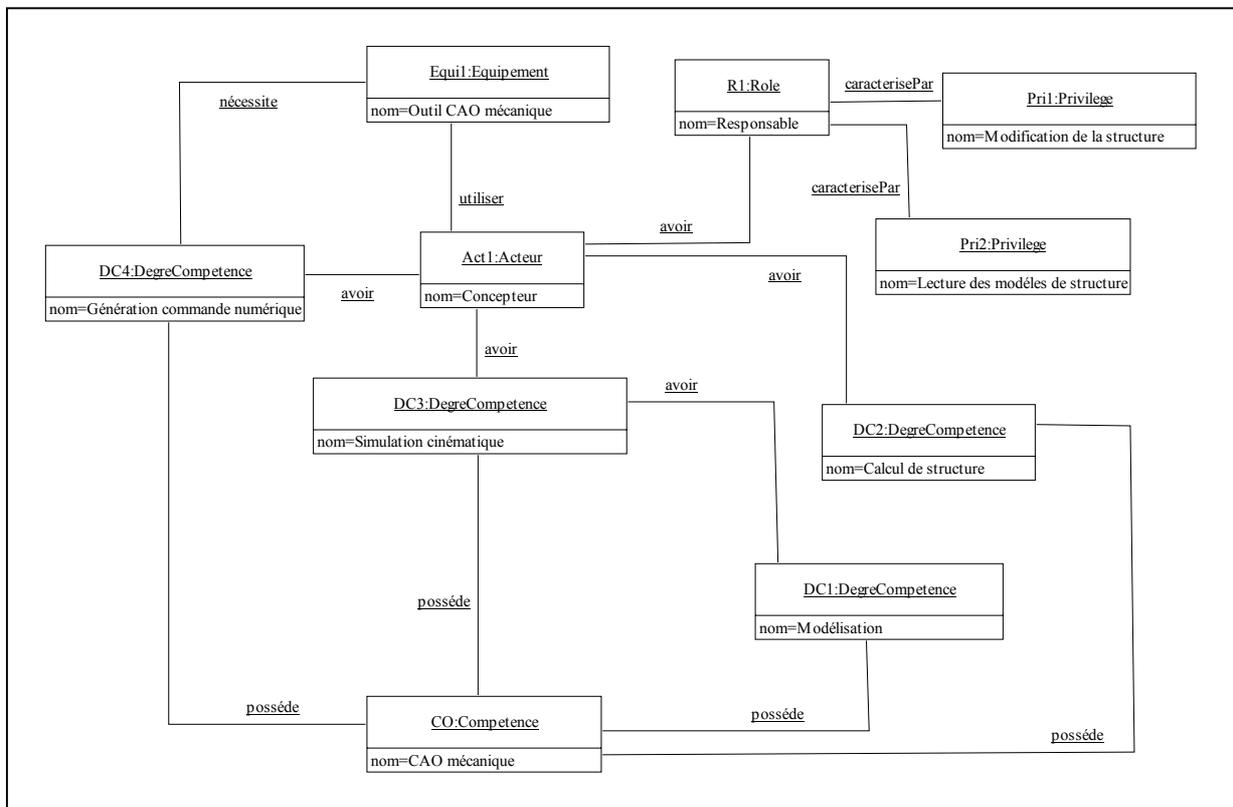


Figure 66 : Diagramme d'objets de description des ressources.

Un élément « Outil CAO mécanique » (Equi1) du type *Equipement* est utilisé par un élément « Concepteur » (Act1) du type *Acteur*. L'élément Act1 a des degrés de compétence « Modélisation » (DC1), «calcul de structure», « simulation cinématique » (DC3) et « génération commande numérique » (DC4) éléments du type *DegréDeCompétence*. Les éléments DC1, DC2, DC3 et DC4 sont des degrés de compétence de la compétence « CAO mécanique » (CO) élément du type *Compétence*. L'élément Act1 a un rôle « responsable » (R1) du type *Role*. L'élément R1

est caractérisé par deux éléments « modification de la structure » (Pri1) et « lecture des modèles de structure » (Pri2) éléments du type *Privilege*.

III.3.13. Le Modèle de Justification

1) Contexte

Le concept de justification est largement employé en intelligence artificielle. Dans ce domaine et en particulier dans le contexte des systèmes à base de connaissances, les justifications sont utilisées pour :

- Maintenir la cohérence dans des raisonnements;
- Explication de l'historique d'application des règles utilisées dans un raisonnement;
- Aide à la décision.

D'après le dictionnaire [Larousse, 2000] la justification est une preuve d'une chose par titres ou par témoins.

[Cantzler, 1997] a défini le concept de justification comme étant un ensemble de raisons, d'arguments, ou de preuves qui expliquent dans un certain domaine de validité le résultat d'une activité cognitive.

L'une des conditions à satisfaire pour réussir à créer des mémoires de projet en conception est de minimiser l'implication des concepteurs car on souhaite exploiter les solutions archivées afin de pouvoir les réutiliser dans d'autres projets. Ceci peut conduire à chercher et à exploiter au mieux l'information habituellement produite dans tout projet pour identifier les raisons des choix ou justifications qui ont été faits. Dans notre cas, on s'intéressera aux justifications du type document car les concepteurs s'appuient sur leur expérience pour effectuer des choix de conception dont la justification est répertoriée dans des documents. C'est le rôle du paquetage *Documentation* (voir Figure 42) qui permet de gérer la structure, le contenu, la justification des choix techniques, et d'indiquer les références à tout support d'information (image, texte, lien hypertexte...).

2) Diagramme de classes

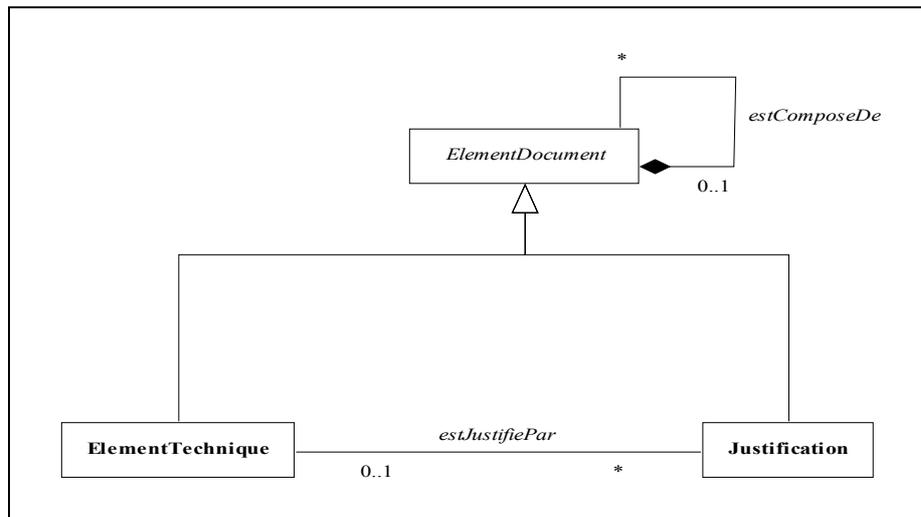


Figure 67 : Diagramme de classes de la Justification.

Un élément de la classe *ElementDocument* (classe abstraite) peut être soit un *ElementTechnique* (par exemple un ensemble potentiel de principes technologiques pour l'architecture d'une turbopompe), soit une *Justification* (par exemple la raison pour laquelle une architecture particulière est choisie, ou pourquoi une autre est abandonnée).

3) Diagrammes d'objets

L'objectif des diagrammes d'objets relatifs aux Figures 69 à 73 est de présenter l'évolution de la structure du produit « Turbopompe ». Pour des raisons de lisibilité, l'ensemble des instances est présenté sur plusieurs diagrammes. La Figure 60 illustre à l'aide d'un diagramme d'objets la description d'une variante (V1). Cette variante est constituée d'une instance de la classe *Produit* à savoir la Turbopompe qui est un élément racine de la décomposition, et d'une partie de ses constituants instances de la classe *ElementProduit*, qui décrit les nœuds de la décomposition.

La Figure 61 montre la structure de la nouvelle version (V2) : un joint *joint3* a été ajouté au palier pour résoudre un problème d'étanchéité dynamique.

L'historique de l'évolution entre les versions V1 et V2 est préservé grâce à l'objet *E1*, instance de la classe *Evolution* (Figure 69).

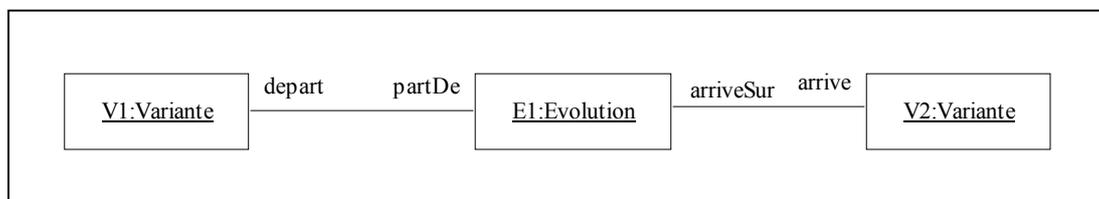


Figure 68 : Diagramme d'objets d'évolution de deux éléments du type Variante.

La figure 70 (resp. 71) illustre la situation suivante : La variante V1 (resp. V2) est documentée par un document *D1* (resp. *D2*) instance de la classe *Document*. Ce document est composé d'un

objet *It1* (resp. *It2*) instance de la classe *Element Technique*. Cet élément technique qui décrit les éléments techniques du palier est justifié par une justification *J1* (resp. *J2*) instance de la classe *Justification* et référence, par l'intermédiaire de l'objet *Cd1* (resp. *Cd2*) de la classe *Référence*, la version *V1* (resp. *V2*).

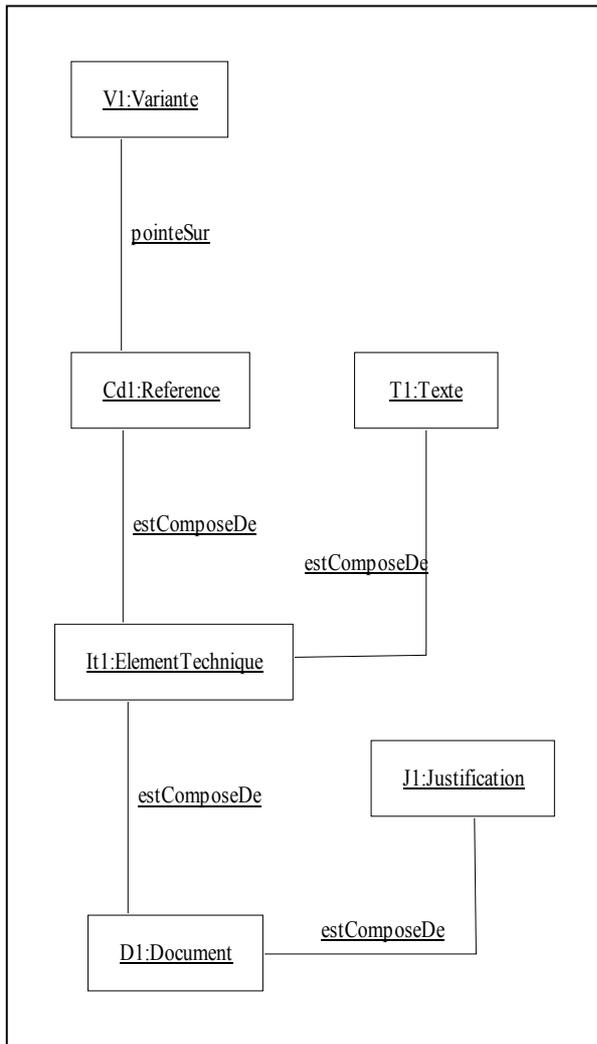


Figure 69 : Diagramme d'objets de description de la Variante V1

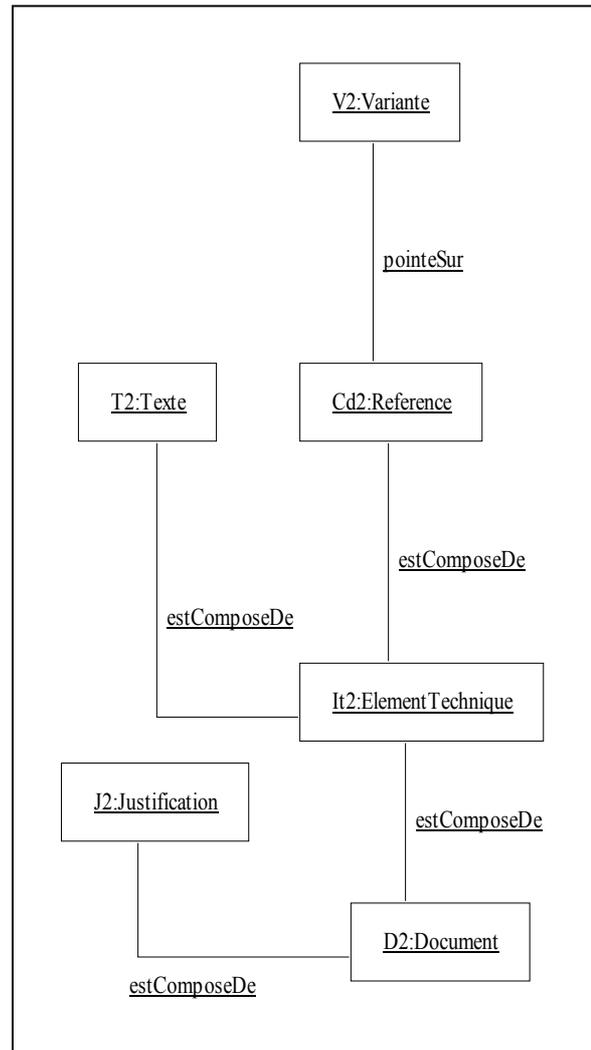


Figure 70 : Diagramme d'objets de description de la Variante V2

La Figure 72 illustre le cas de la traçabilité au niveau des justifications qui se traduit par la création d'une instance *E2* de la classe *Evolution* qui relie les justifications *J1* et *J2*. Cette évolution *E2* est reliée à l'évolution *E1* pour conserver une trace entre l'évolution des versions et des justifications correspondantes. De manière à également introduire explicitement des éléments

d'historique entre les documents, une instance *E3* de la classe *Evolution* entre les documents D1 et D2 a été créée.

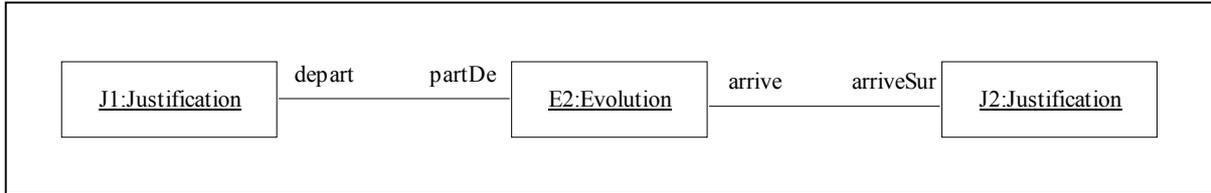


Figure 71 : Traçabilité au niveau des justifications.

L'ensemble de ces instances et de leurs liens est représenté sur la Figure 73.

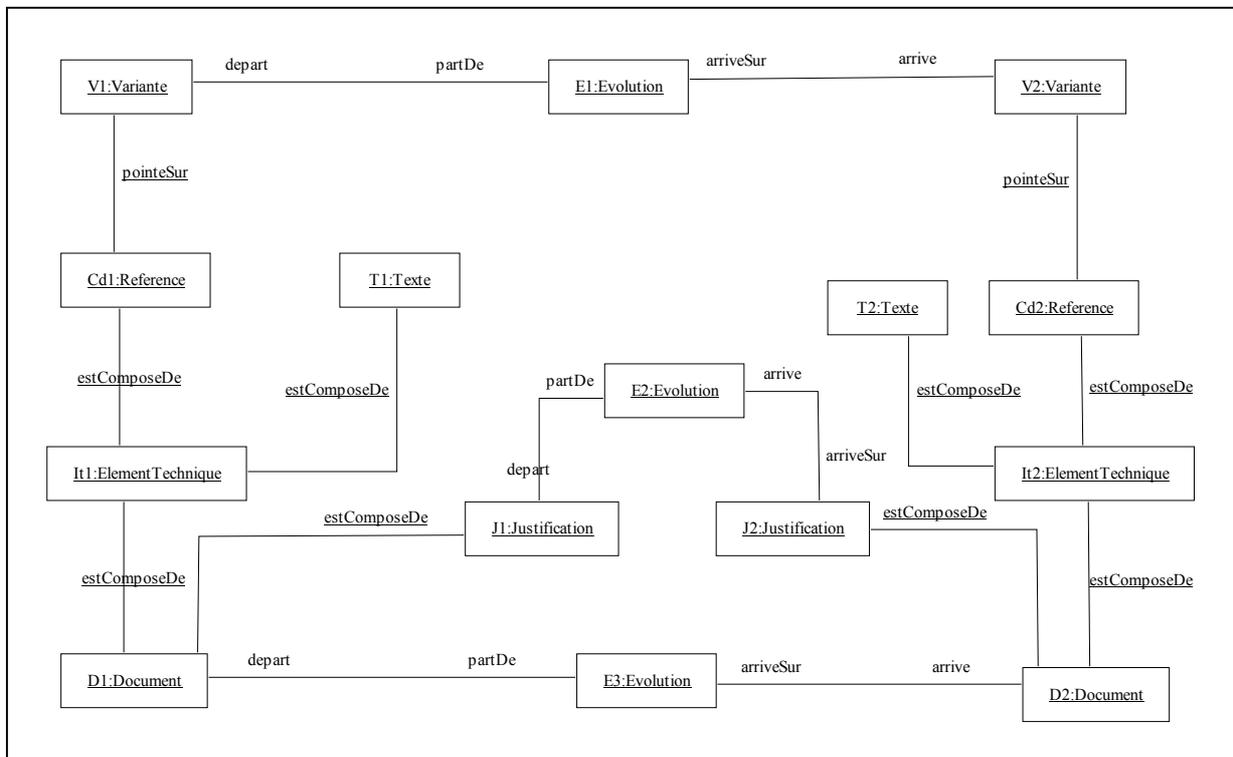


Figure 72 : Diagramme d'objets décrivant l'ensemble des relations d'évolution.

Dans cet exemple, les justifications concernent les variantes. Mais compte tenu, d'une part, de l'existence de la classe *Entité*, super classe de tous les éléments de la mémoire de projet, et, d'autre part, le fait de lever au plus au niveau d'abstraction de toutes les relations entre les entités ainsi que des références de la documentation, il est possible de construire des documents comportant des justifications sur n'importe quel élément de la mémoire de projet et de suivre ainsi l'évolution de ces justifications

III.3.14. Le Modèle des points de Vue

1) Contexte

[Marino et al, 1990] définit le point de vue comme étant «une perspective d'un intérêt à partir duquel un expert examine la base de connaissances».

Dans sa définition Ribière [Ribière, 2002] s'est basée sur celle de Marino mais en s'intéressant de plus à deux dimensions, à savoir la dimension contextuelle où l'avis (the focus) d'un expert est décrit, et la dimension de la personne d'où l'angle de vue d'un expert est décrit :

- L'avis décrit le contexte du travail de l'expert. Plusieurs experts peuvent avoir le même avis. Cependant, on caractérise le point de vue par la dimension personnelle : l'angle de vue ;
- L'angle de vue décrit les caractéristiques d'un expert ou d'un groupe d'experts. Il peut décrire le nom de cet expert, son domaine d'application, son niveau d'expertise ou compétences, son expérience dans d'autres domaines intéressants pour l'application, son rôle et sa place dans l'organisation.

De ce fait la définition de Ribière est : «un point de vue est une interface permettant l'indexation et l'interprétation de la vue composant les éléments de la connaissance. Un point de vue est caractérisé par un avis et l'angle de vue. » [Ribière, 2002].

Dans le cas de nos travaux, on s'intéresse à la construction d'un modèle de points de vue qui sera indépendant des éléments modélisés (produit, processus, document, etc.), et de son domaine d'application. Quant à notre démarche d'élaboration du modèle elle sera progressive.

2) Diagramme de classes

Version 1 : Notre objectif est de gérer plusieurs points de vues c'est-à-dire de proposer un modèle de points de vue par domaine d'application de l'acteur. La structure de modélisation la plus appropriée pour la classification est la généralisation/spécialisation ce qui nous amène à proposer le modèle de la Figure 74 sur laquelle la classe *Acteur* se spécialise selon les différents points de vue pris en compte.

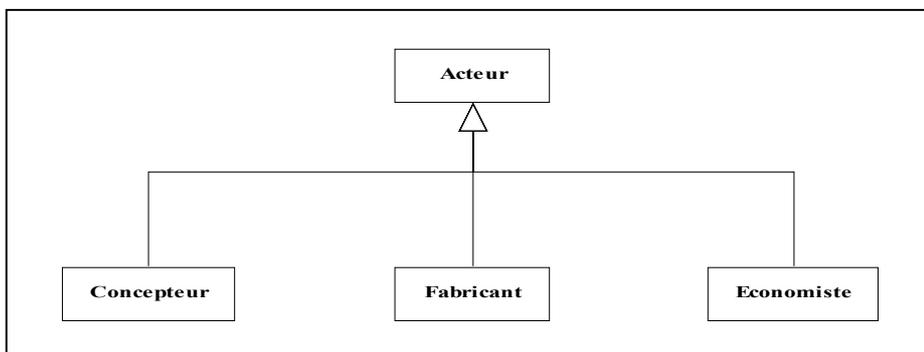


Figure 73 : Diagramme de classes de description de Point de Vue (Version 1).

Dans une telle version, il n'est possible d'allouer qu'une vue unique à chaque acteur car les acteurs sont des instances des classes feuilles. Le nombre de vues définies a priori est non évolutif par acteur.

Version 2 : Suite aux insuffisances relevées dans le diagramme de la Figure 75, nous allons introduire la classe *Vue*.

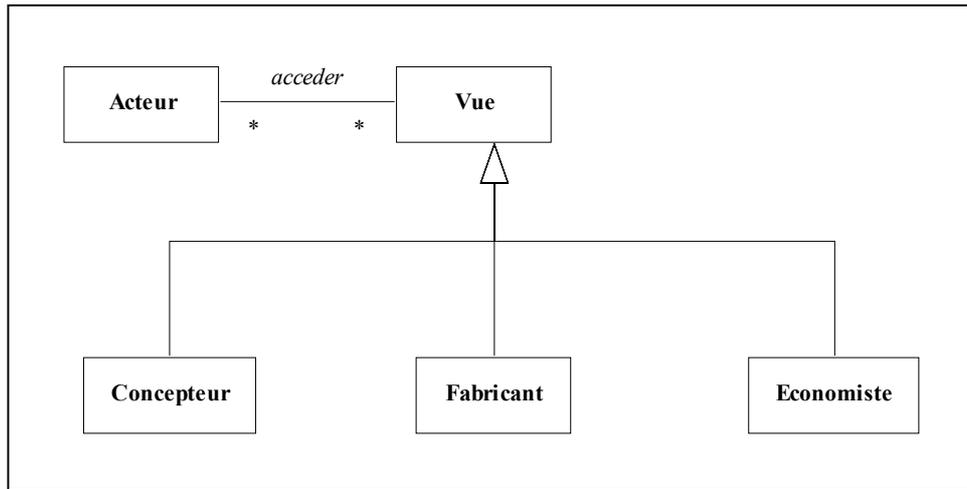


Figure 74 : Diagramme de classes de description de Point de Vue (Version 2).

D'après [Mossé, 2002] et grâce à la multiplicité 0..* du côté de la classe *Vue*, il est possible d'attribuer maintenant plusieurs vues à un acteur. Cependant le nombre de vues prise en compte est limité par les différentes sous classes de la classe *vue*. En effet, si un acteur peut accéder à plusieurs vues il ne pourra accéder qu'aux vues définies statiquement par les sous classes, et il ne sera pas possible d'ajouter dynamiquement de nouvelles vues.

Version 3 : Nous allons présenter le diagramme de la version 3 en deux étapes ; cette version vise à améliorer la solution version 2.

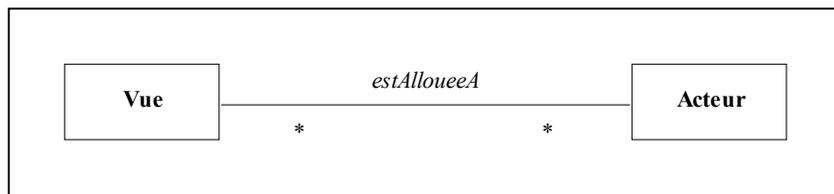


Figure 75 : Diagramme de classes de description de Point de Vue (Version 3-1).

La Figure 76 illustre la possibilité d'allouer un nombre de vues qui est devenu illimité par acteur. Une vue est allouée à un ou plusieurs acteurs. Un acteur peut avoir plusieurs vues. Une classe

attribut est ajouté afin de gérer dynamiquement les vues. Le changement dynamique des vues n'affecte pas les modèles existants.

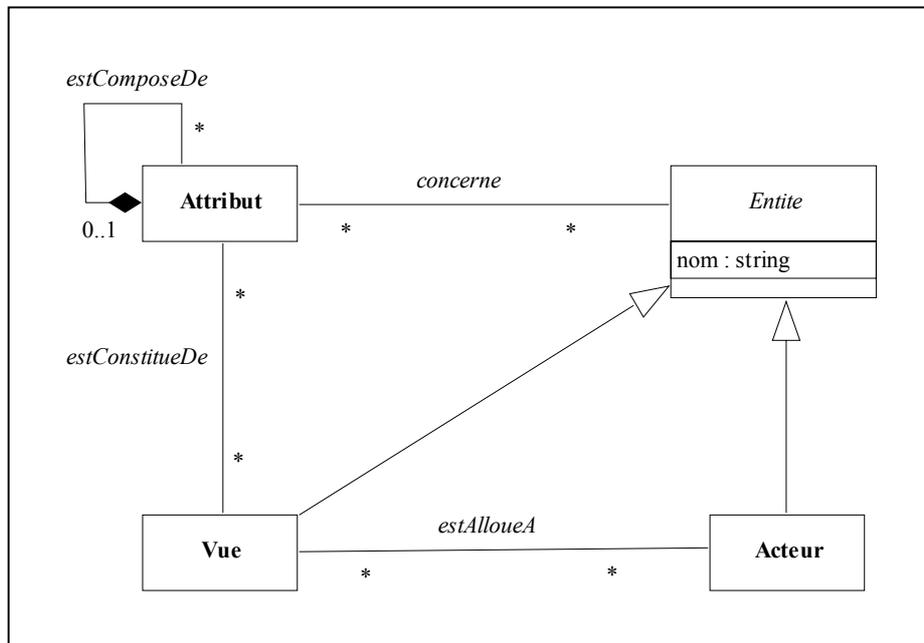


Figure 76 : Diagramme de classes de description de Point de Vue (Version 3-2).

Dans une telle version on alloue une ou plusieurs vues à chaque Acteur. Le nombre de vues est évolutif. Une gestion dynamique des informations visibles est possible. Il est possible d'avoir aussi bien des attributs simples ou composés. D'où la relation du type *estComposeDe*. Un élément de la mémoire de projet peut être du type *Vue*. Une vue peut concerner un ou plusieurs élément de la mémoire de projet. L'ensemble des attributs des différents modèles de la mémoire de projet s peuvent être mis dans la classe *Attribut*.

C'est une solution qui répond à nos attentes où la notion de vue dans sa construction est indépendante des éléments de la mémoire de projet et du domaine d'application de la mémoire de projet.

2) Diagramme d'objets

Considérons un rotor défini par deux attributs : le prix cumulé et la masse totale. Ces deux attributs font partie de deux points de vue la gestion des masses et la gestion des prix défini pour deux acteurs.

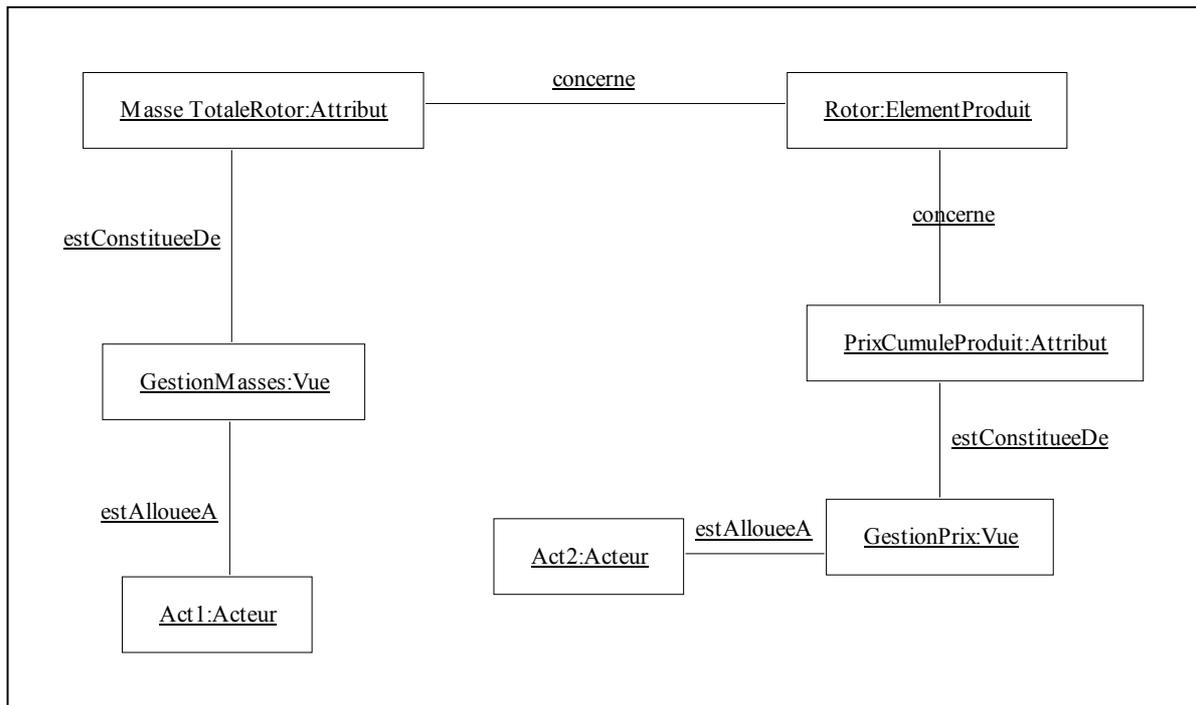


Figure 77 : Diagramme d'objets de description de Point de Vue (Version 3).

A un élément Act1 du type *Acteur* est alloué l'élément GestionMasses élément du type *Vue*. Cet élément est constitué de l'élément MasseTotalRotor élément du type *Attribut*. A l'élément Act2 du type *Acteur* est alloué l'élément GestionPrix du type *Vue*. Cet élément est constitué de l'élément PrixCumuléProduit élément du type *Attribut*. L'élément Rotor du type *ElementProduit* est constitué des deux attributs MasseTotalRotor et PrixCumuléProduit.

III.3.15. Le Modèle des Objectifs

1) Contexte

Un objectif est un but à atteindre. C'est le résultat d'une prévision, d'un projet, d'une phase, d'un processus de conception, de la composition d'éléments d'un produit, etc. Il explique le résultat à atteindre plutôt que le moyen d'y parvenir.

La formalisation d'objectifs est importante dans une mémoire de projet. La définition d'un objectif se doit d'être claire et précise. Il ne mesure qu'une seule chose à la fois et à chaque objectif correspond une action ou plusieurs. Chaque objectif voit sa finalité concrétisée par une date et un ou plusieurs buts atteints. Il sera accompagné des moyens adéquats (ressources, du savoir-faire, etc.) pour être concrétisé.

Les objectifs qu'on définit dans une mémoire de projet s'inscrivent dans le cadre d'une politique générale de l'entreprise. C'est la manifestation concrète de la réalisation de la politique choisie. Cependant l'atteinte de l'objectif ne peut se réaliser que par la participation de tous les participants à la construction de la mémoire de projet. Aussi, les utilisateurs doivent être associés

à la réflexion et la construction. Ils comprendront d'autant mieux l'intérêt et la nécessité d'une définition d'objectifs. Leur adhésion aux objectifs reposera sur cette connaissance claire et la possibilité d'avoir des moyens pour les atteindre.

2) Diagramme de classes

Un objectif permet, pour un projet de conception donné, de fixer des buts qualitatifs, des buts quantitatifs, des échéances, des résultats, etc. Par exemple, un produit peut avoir pour objectif de ne pas dépasser un coût fixé, un concepteur de terminer la conception d'un produit avant une date fixée, etc. L'objectif peut donc caractériser les éléments de la mémoire de projet. C'est pour cela qu'on s'intéresse à sa modélisation comme étant un élément de la mémoire de projet.

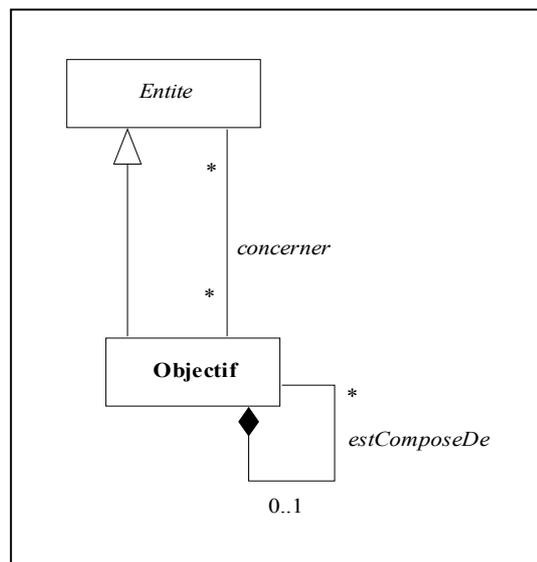


Figure 78 : Diagramme de classes de description des objectifs.

Un élément de la mémoire de projet peut être du type *Objectif*. Un objectif peut concerner un ou plusieurs éléments de la mémoire de projet. Un objectif peut être décomposé en plusieurs objectifs.

2) Diagramme d'objets

L'objectif du diagramme d'objet suivant est de décrire quels sont les objectifs tracés du produit divertissement.

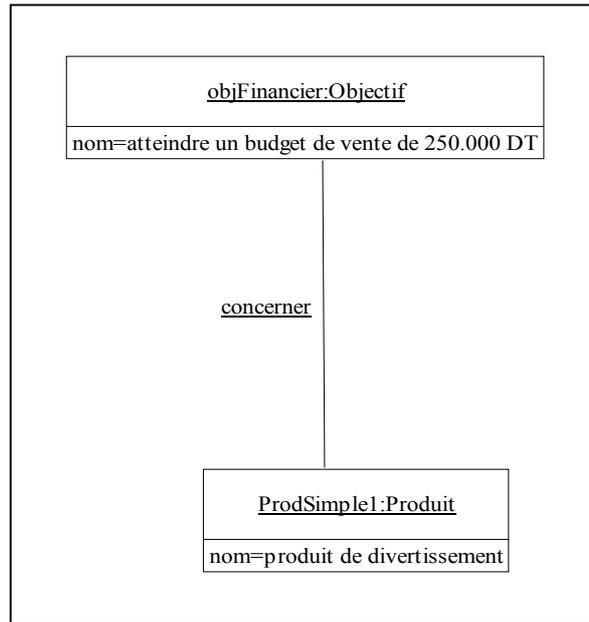


Figure 79 : Diagramme d’objets de description des Objectifs.

Un élément «produit de divertissement» (ProdSimple1) de type *Produit* est concerné par l’objectif financier «atteindre un budget de vente de 250.000 DT» (objFinancier) élément du type *Objectifs*.

III.3.16. Le Modèle de Projet

1) Contexte

La norme X50-150 de l’AFNOR [AFNOR, 1991], définit le projet comme « une démarche spécifique qui permet de structurer méthodiquement et progressivement une réalité à venir » et ajoute « un projet est défini et mis en œuvre pour élaborer une réponse au besoin d’un utilisateur, d’un client ou d’une clientèle et implique un objectif et des actions à entreprendre avec des ressources données ». Un projet, c’est aussi un ensemble fini comportant un début et une fin, un caractère unique, une aventure mêlant des expériences positives et négatives.

2) Diagramme de classes

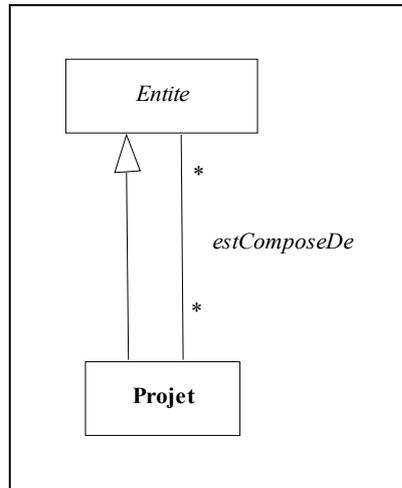


Figure 80 : Diagramme de classes de Projets.

Un élément de la mémoire de projet peut être du type *Projet*. Il peut être en relation avec d'autres éléments de la mémoire de projet.

2) Lien avec d'autres modèles

L'objectif du diagramme d'objets suivant est de décrire partiellement un projet de type « conception Turbopompe ».

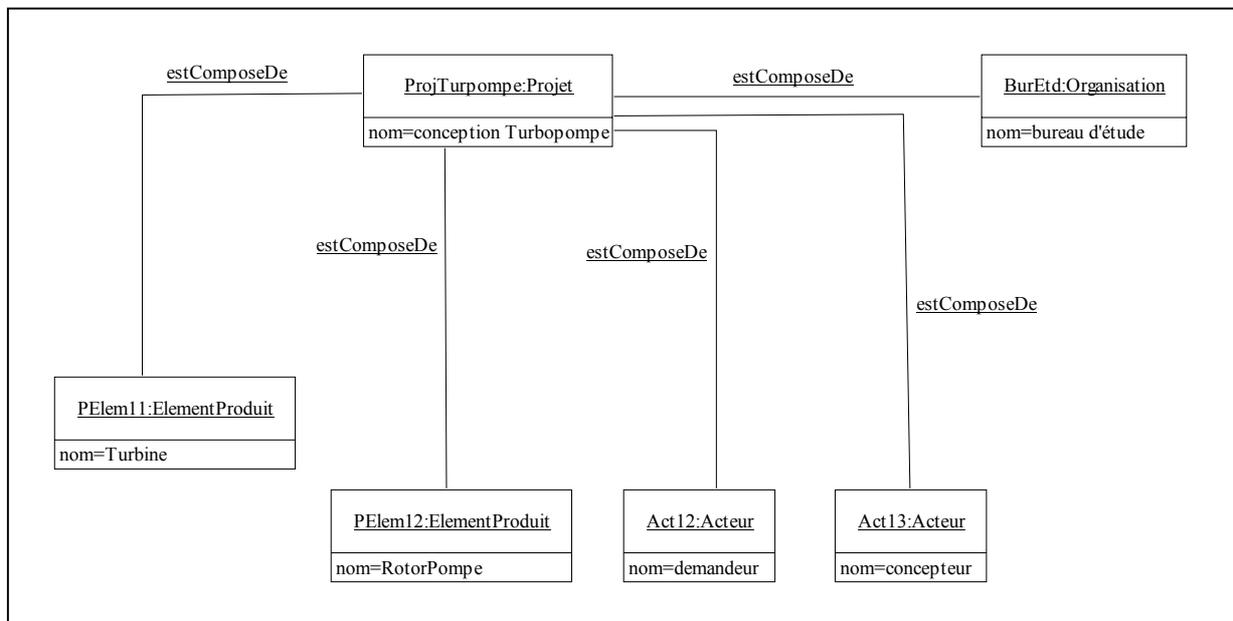


Figure 81 : Diagramme d'objets partiel de description d'un Projet.

L'élément « conception Turbopompe » (Proj) du type *Projet* est en relation avec l'élément «bureau d'étude» (BurEtd) du type *Organisation*. Ce dernier est lié aux éléments «demandeur» (Act12) et « concepteur » (Act13) du type *Acteur*. De même l'élément Proj est en relation avec les éléments « Turbine » (PElem11) et « RotorPompe » du type *ElementProduit*.

III.4. Conclusion

Dans ce chapitre, nous avons présenté une architecture générale de la mémoire de projet et les différents modèles la constituant en utilisant le langage UML. Ces modèles ont été construits sur la base d'un patron de modélisation qui nous a servi à la représentation des structures arborescentes rencontrées fréquemment dans le domaine. Nous l'avons utilisé pour la description des produits, des processus, des documents, et adapté aux organisations.

Afin de factoriser au maximum les propriétés, une classe abstraite entité a été introduite comme super classe de tous les éléments de la mémoire de projet. L'ajout de cette super classe a permis aussi de reporter au niveau de généralisation le plus élevé une relation à sémantique simple « *estLiéA* ». De ce fait il est alors possible de relier un élément de la mémoire de projet avec n'importe quel autre indépendamment de leur type.

Cependant il est nécessaire de pouvoir également mettre en place des relations à sémantique plus riche entre les éléments de la mémoire de projet. Nous avons donc introduit des relations prenant en compte les trois aspects classiques de la systémique : fonctionnelle, structurelle et temporelle.

Pour ce dernier type de relation, nous avons défini plus particulièrement la relation du type *Evolution* qui nous permet de représenter l'évolution d'un élément de la mémoire de projet vers un autre élément. Si l'on ajoute à ce type de relation la possibilité de mettre des attributs de type «date de création» au niveau de la classe la plus générale, il est alors possible de tracer l'historique complet de tout élément de la mémoire de projet.

Comme il n'est pas possible de garder tout l'historique de tous les éléments de la mémoire de projet et de manière à aider le concepteur au choix de la relation la mieux adaptée nous avons proposé un guide de choix des relations.

La classe entité et l'ensemble des relations ont été regroupés dans un paquetage *Corps* qui constitue la colonne vertébrale de la mémoire de projet. Ce paquetage peut être considéré comme une structure d'accueil dans laquelle viennent s'insérer les différents constituants de la mémoire de projet : processus, produit, organisation. L'architecture de domaine proposée constitue ainsi une base qu'il convient d'adapter pour aboutir à un modèle d'un niveau de détail plus fin.

Dans le chapitre suivant nous présenterons l'architecture logique de notre système d'information, le type de plateforme sur laquelle il va être déployé, les langages de développement ainsi que l'entreprise d'accueil qui nous a servi de support pour l'élaboration d'une méthode de mise en place d'une mémoire de projet.

CHAPITRE IV : VERS UN OUTIL DE DEFINITION ET DE REUTILISATION DE LA MEMOIRE DE PROJET : P2M2

IV.1. Introduction

L'objectif de ce chapitre est de présenter les aspects applicatifs des modèles développés pour la mémoire de projet de manière à valider par l'expérimentation les modèles proposés. Il s'agit d'une application permettant de saisir les informations structurées d'un projet donné et de consulter le contenu de la mémoire constituée par l'ensemble des informations stockées.

Un tel outil informatique est une adaptation à un contexte particulier d'entreprise des modèles génériques proposés au Chapitre 3.

Dans ce chapitre, nous présentons l'outil P2M2 (Project Product Memory Management system), outil élaboré dans le cadre de ce travail. Nous présentons les choix technologiques adoptés pour le développement d'un tel prototype. Nous parlons des modèles choisis pour le développement et des différentes techniques de transformation. Un guide utilisateur de transformation de modèles est présenté. Des exemples et des illustrations sont présentés dans ce chapitre montrant les différentes fonctionnalités et interfaces de l'outil utilisé pour la mémoire de projet.

Une adaptation de nos modèles sur un cas réel d'entreprise d'ingénierie est développée : il s'agit de la société STUDI (Société Tunisienne de Développement et d'Ingénierie). Ce cas d'entreprise nous a permis de tester des modèles et de rétro-agir en apportant les corrections nécessaires. Un bilan d'adaptation de ces modèles sur le cas de la société STUDI est présenté avec une approche critique ; des propositions d'évolutions sont aussi faites.

IV.2. Outils utilisés pour le développement de P2M2 :

Dans le présent paragraphe nous allons présenter l'ensemble des outils de développement et le type de la plateforme de déploiement. L'argumentation des choix faits sera présentée au fur et à mesure de la présentation des outils. Nous présentons les éléments de la mémoire de projet qui ont été choisis pour le développement. Un guide de transformation des modèles de classes en objet relationnel sera présenté. Nous parlerons de la maquette de l'application et nous présentons ses fonctionnalités avec des illustrations.

IV.2.1. Les choix technologiques

Notre choix s'est basé sur l'architecture J2EE du fait de son caractère standard. Une telle architecture est spécifiquement tournée vers le développement et le déploiement d'applications d'entreprise orientées web utilisant le langage de programmation Java. Une architecture J2EE a pour objectif essentiel de standardiser le fonctionnement des serveurs d'applications Java et d'homogénéiser leurs architectures dans un but de faciliter le déploiement d'applications métier à base de composants. Ce type d'architecture, répond à nos besoins de développement d'applications et permet essentiellement une conception et une réalisation par composants. Ces besoins peuvent se résumer principalement en l'indépendance vis-à-vis des plateformes de déploiement (les spécificités des produits utilisés n'interviennent pas dans le code des composants mais uniquement dans les descripteurs de déploiement).

D'une manière générale, l'architecture standard définie par J2EE comporte les éléments suivants :

- Un modèle d'applications standard pour le développement d'applications multi-niveaux ;
- Une plate-forme standard hébergeant les applications ;
- La définition d'un ensemble de tests de compatibilité servant à vérifier que les produits J2EE répondent bien à la norme J2EE ;
- Une implémentation de référence apportant une définition du fonctionnement de la plate-forme J2EE.

Le gestionnaire de la base de données qui est choisi est Oracle 10g du type SGBDOR³⁷. L'avantage de tels systèmes par rapport aux SGBDOO³⁸ est la rapidité d'accès et la compatibilité ascendante avec les SGBDR. Le modèle objet relationnel est basé sur l'extension du modèle relationnel aux concepts objets. Le système dans sa forme générale reste relationnel mais les concepts clés de l'objet lui sont ajoutés dans une forme particulière afin d'intégrer les deux modèles. De plus, leur langage de requêtes est basé sur la norme SQL3 [Lentzler, 2004] qui est une extension du langage SQL à des concepts objet. En effet, ce système offre dès à présent l'héritage de types, facilitant la représentation de l'héritage au sens de l'approche objet, largement utilisé dans nos modèles. Ce choix nous permet également de profiter des performances du modèle relationnel qui est plus mature et plus efficace. De plus, la place que

³⁷ SGBDOR : Systèmes de Gestion des Bases de Données Relationnelles

³⁸ SGBDOO : Systèmes de Gestion des Bases de Données Orientées Objets

commencent à occuper les SGBDOR permet au développement de notre mémoire de projet d'aboutir à la réalisation d'un prototype de système opérationnel et crédible. De plus, pour construire les types de données, le développeur peut définir de nouveaux types d'objets qui peuvent modéliser des structures complexes tels que les types hiérarchiques, les références d'objets, etc. Comme les classes, les objets rendent facile la modélisation des entités du monde réel et la réutilisabilité des objets rend le développement des applications orientées bases de données plus rapides et plus efficaces.

IV.2.2. Choix des modèles implémentés

Les modèles développés dans le chapitre 3 constituent une base pour l'application à la mémoire de projet. Afin de tester la faisabilité de ces modèles on se propose de développer une maquette logicielle pouvant nous garantir le déploiement des modèles. Nous nous sommes fixés sur un certain nombre de modèles, les plus significatifs, afin de les valider expérimentalement. L'objectif n'est pas de tout implémenter mais d'avoir un prototype de la mémoire de projet qui soit valide. Ce choix s'est fixé sur les modèles de Produit, Evolution, Variante et Document.

Ainsi le produit représente l'élément essentiel du cadre de conception de la mémoire de projet. Notre mémoire est une mémoire qui doit permettre l'accès à un certain historique de contexte de projet et de la logique de conception. Cet historique est composé d'éléments qui ont subi des évolutions lors de la vie de la mémoire. D'où le choix d'implémenter la relation Evolution. Le concept d'évolution est souvent appliqué à des éléments de variante qui n'est autre qu'une évolution d'éléments composants la variante.

En conception de produit, il est souvent nécessaire de formaliser les choix et leurs justifications afin de garder une trace des responsabilités, mais également pour permettre une réutilisation dans les projets ultérieurs. La plupart du temps, ces choix et leurs justifications sont répertoriés dans des éléments de type documents issus du paquetage Documentation.

IV.2.3. Règles de transformation des modèles : des modèles de classes UML en objets persistants Oracle10g

Certaines transformations de modèles sont nécessaires pour leur déploiement sous le SGDBOR Oracle 10g. L'idée est d'utiliser un AGL tel qu'Objecteering [Objecteering, 2005] pour effectuer les transformations, outil déjà utilisé tout le long de cette thèse. La contrainte rencontrée est qu'il n'existe pas sur le marché un AGL qui permet de transformer des modèles métiers en modèles du type objet-relationnel.

L'objectif de ce paragraphe est d'élaborer un guide de transformation des différents types de modèles de classes présentés dans le chapitre 3 en objet Oracle 10g persistant. Avant de présenter les règles de transformation, rappelons les principes de transformation des modèles d'après [Gardarin, 1999] dans le cas d'une base de données objet-relationnel. Ces règles sont au nombre de trois:

- **Règle 1** : Pour chaque classe, générer le type SQL3 correspondant par la commande CREATE TYPE. Utiliser l'héritage de type pour traduire les spécialisations et l'agrégation de type pour les agrégations composites.
- **Règle 2** : Si une classe n'est pas cible d'une agrégation composite ou d'une généralisation, alors l'implémenter comme une table d'objets du type associé. Cela conduit :

- A implémenter toutes les classes feuilles des hiérarchies d'héritage en incluant les attributs hérités ;
- A respecter les agrégations composites en ce sens que les instances de la classe cible figureront dans la table associée à l'autre classe.

- **Règle 3** : Implémenter toutes les associations par des attributs références mono ou multivalués (en respectant les multiplicités maximales de l'association) d'une table vers une autre, éventuellement dans les deux sens. L'utilisation de référence dans les deux sens permet les parcours de chemins dans les deux sens. C'est utile si les requêtes le nécessitent.

L'ensemble de ces règles nous a permis d'énoncer les différents cas de figure de transformation qu'on peut rencontrer à savoir le passage d'un diagramme de classes UML vers un schéma Oracle 10g. Les classes et tables utilisées seront des exemples de nos modèles énoncés dans le chapitre 3. Pour chacun des cas on présentera le principe de transformation, un modèle de classes, sa transformation en un schéma relationnel et le script SQL correspondant.

1. Cas d'une association du type 1..* :

- **Principe de base** : Le passage d'un diagramme de classes UML vers un schéma Oracle 10g nécessite la création d'un type pour chaque classe. Par la suite pour chaque type créé sous Oracle, on crée les tables correspondantes. Les liens d'associations sont traduits par des références en utilisant le mot clé REF.

- **Diagramme de classes** : Le diagramme de classes de la Figure 83 illustre l'association entre la classe Document et la classe Produit. Les deux sont liées par la relation estLieA.

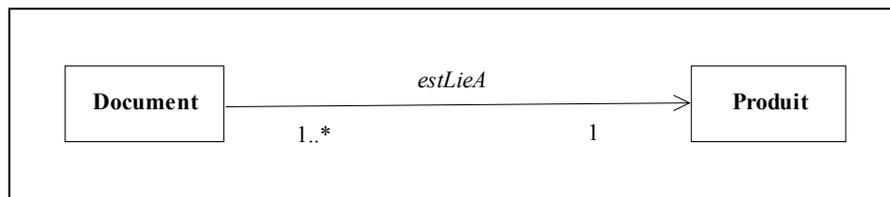


Figure 82 : Exemple d'un diagramme de classes utilisant l'association du type 1..*

- **Schéma de transformation**

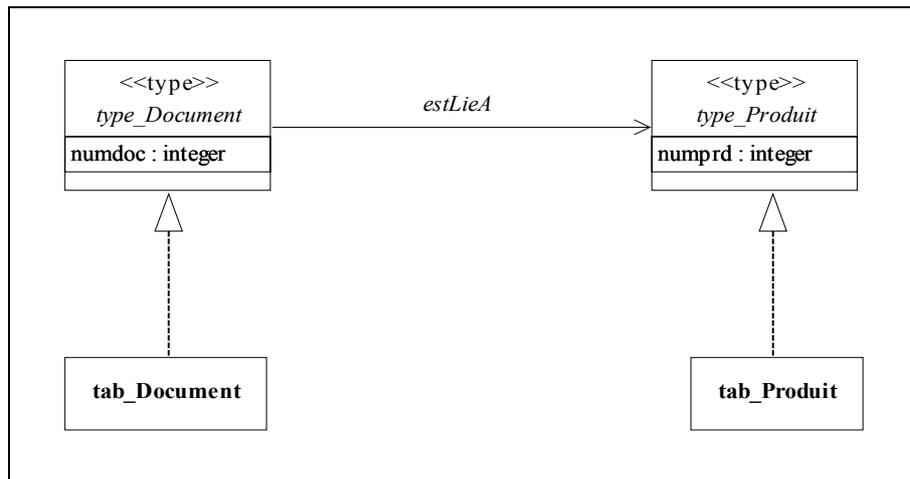


Figure 83 : Schéma de transformation d'un diagramme de classe utilisant l'association du type 1..*

- **Scripts SQL**

- **Création des types sous Oracle :** Pour améliorer les performances en réduisant les jointures, nous pouvons conserver dans les tables une référence à une valeur (pointeur), plutôt qu'une clé étrangère. Nous pouvons ainsi manipuler des objets de structure complexe avec des performances correctes. Ainsi et dans un but de référencer les tuples d'une table dans une autre table, il faut créer cette table à partir d'un type structuré. Ainsi nous pouvons indiquer dans la définition d'un type qu'un attribut contient des références (pointeurs) à des données d'un autre type grâce à l'opérateur REF. Le lien en pointillé entre par exemple tab_Document et type-Document est une relation d'implémentation classique telle que définie dans UML. Il permet d'avoir les attributs qui existent dans le type type_Produit.

La syntaxe de création des types est la suivante :

```
CREATE TYPE type_Produit AS OBJECT (
  numprd      NUMBER(6,0)
);
CREATE TYPE type_Document AS OBJECT (
  numdoc      NUMBER(6,0) ,
  estLieA     REF type_Produit
);
```

- **Création des tables :** Une table n'est une table objet-relationnelle que si elle est créée à partir d'un type. Tous les types sont issus du même type type_Entity. De même il n'existe qu'une seule table TAB_ENTITY issue du type type_Entity. Le script SQL correspondant est le suivant :

```
CREATE TABLE tab_Produit OF type_Produit
  (Constraint pk_numprd PRIMARY KEY(numprd))
```

```
CREATE TABLE tab_Document OF type_Document
  (Constraint pk_numdoc PRIMARY KEY(numdoc))
```

2. Transformation d'un lien de composition monovaluée

- **Principe :** Dans le cas d'une association de type composition monovaluée, nous associons pour chaque classe un type. Le lien de composition sera établi entre les types créés. Par la suite pour chaque type créé sous Oracle, on crée les tables correspondantes.

- **Diagramme de classes :**

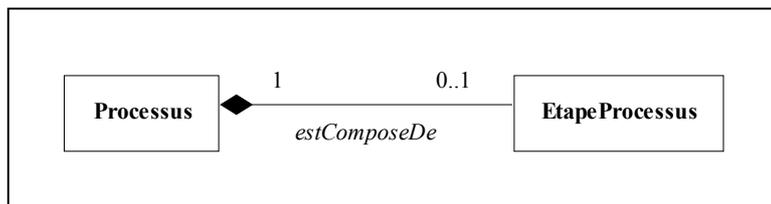


Figure 84 : Diagramme de classes utilisant un lien de composition monovaluée.

- **Schéma de transformation :**

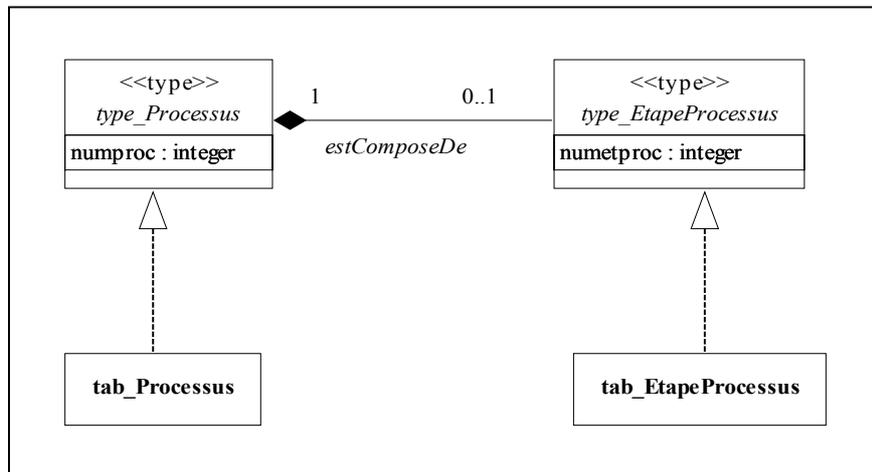


Figure 85 : Schéma de transformation d'un diagramme de classes utilisant l'association du type composition monovaluée.

- Scripts SQL
 - création des types sous Oracle

```
CREATE TYPE type_EtapeProcessus AS OBJECT
(numetproc NUMBER(6,0)
);
```

```
CREATE TYPE type_Processus AS OBJECT
(numproc NUMBER (6,0) ,
  estComposeDe type_EtapeProcessus
);
```

- création des tables

```
CREATE TABLE tab_Processus OF type_Processus
CREATE TABLE tab_EtapeProcessus OF type_EtapeProcessus
```

3. Transformation d'un lien de composition multivaluée

- **Principe :** Dans le cas d'une association de type composition multivaluée, nous associons pour chaque classe un type. On crée un type imbriqué relatif à l'élément de type composant intervenant dans la composition multivaluée. Par suite on crée la table correspondante à l'élément du type composé avec sa table imbriquée.
- **Diagramme de classes**

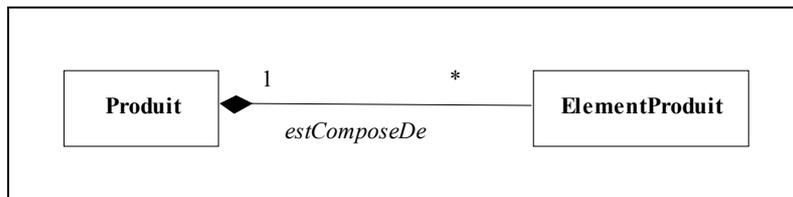


Figure 86 : Diagramme de classes utilisant un lien de composition multivaluée.

- **Schéma de transformation**

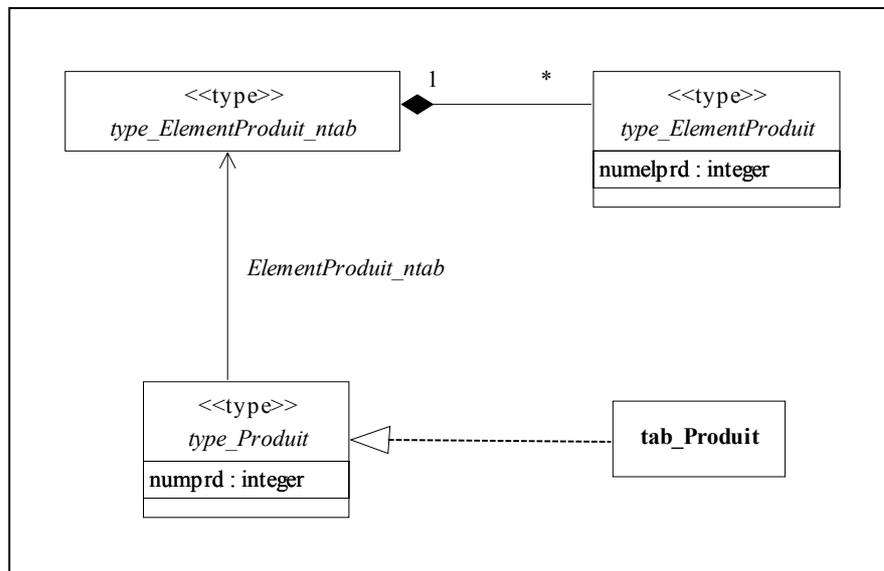


Figure 87 : Schéma de transformation d'un diagramme de classe utilisant l'association du type composition multivaluée.

- **Scripts SQL**

- **création des types sous Oracle** : Pour assurer ce genre de situations de composition multivaluée nous allons utiliser la notion de tables imbriquées (nested table). La table imbriquée est une collection non ordonnée et non limitée d'éléments de même type. La syntaxe de création des types est la suivante :

Création du type type_ElementProduit stockant chaque tuple

```
CREATE TYPE type_ElementProduit AS OBJECT (
  numelprd NUMBER(6,0)
);
```

Création d'une table type_ElementProduit_ntab contenant ces tuples

```
CREATE TYPE type_ElementProduit_ntab AS TABLE OF type_ElementProduit;
```

Création d'un type type_Produit avec un attribut ElementProduit_ntab de type type_ElementProduit_ntab

```
CREATE TYPE type_Produit AS OBJECT (
  numprd NUMBER(4),
```

```

ElementProduit_ntab type_ElementProduit_ntab
);

```

- **Création des tables :** Une table imbriquée est une table à l'intérieur d'une autre table, elle possède toutes les caractéristiques d'une table normale. La syntaxe pour sa création est la suivante :

Création de la table tab_Produit qui contient une colonne de type NESTED TABLE:

```

CREATE TABLE tab_Produit OF type_Produit (numprd PRIMARY KEY)
NESTED TABLE ElementProduit_ntab STORE AS Copie_ElementProduit_ntab;

```

4. Transformation d'associations réflexives

- **Principe :** Dans le cas d'une association de type réflexif, nous associons pour la classe un type. On crée la table correspondante au type.
- **Diagramme de classes**

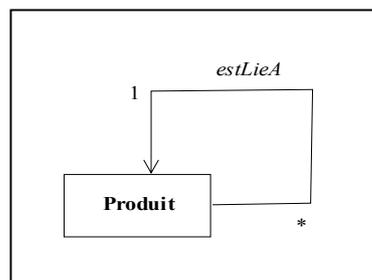


Figure 88 : Diagramme de classes utilisant une association réflexive.

- **Schéma de transformation**

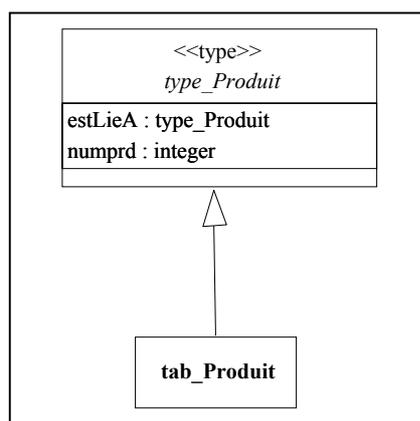


Figure 89 : Schéma de transformation d'un diagramme de classes utilisant l'association réflexive.

- **Scripts SQL**
 - **création des types**

```
CREATE TYPE type_Produit AS OBJECT (
  numprd      NUMBER(6,0),
  estLieA     REF type_Produit
);
```

- **création des tables**

```
CREATE TABLE tab_Produit OF type_Produit
(Constraint pk_Produit PRIMARY KEY (numprd));
```

5. Transformation d'associations réflexives (Composition multivaluée)

- **Principe :** Dans le cas d'une association de type réflexive multivaluée, nous associons pour la classe un type. On crée la table correspondant au type.
- **Diagramme de classes**

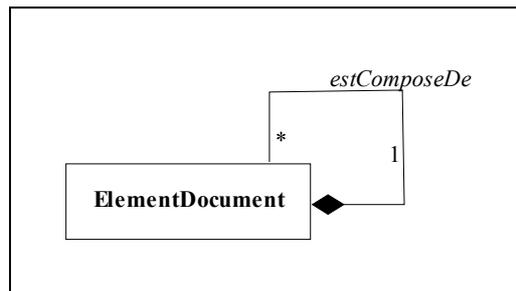


Figure 90 : Diagramme de classes utilisant une association réflexive (Composition multivaluée)

- **Schéma de transformation**

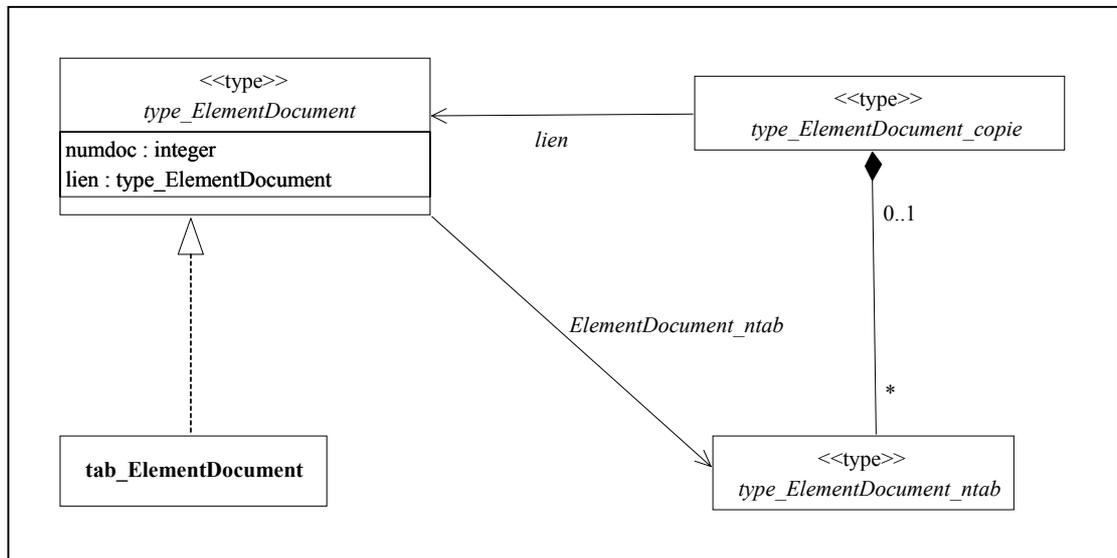


Figure 91 : Schéma de transformation d'un diagramme de classes utilisant l'association réflexive (Composition multivaluée)

- **Scripts SQL**

- **création des types**

Création du type type_ElementDocument

```
CREATE TYPE type_ElementDocument;
/
```

Création d'une copie de ce type avec un lien de référence vers le type type_ElementDocument

```
CREATE TYPE type_ElementDocument_copie AS OBJECT
(lien REF type_ElementDocument);
/
```

Création d'une table type_ElementDocument_ntab contenant les tuples de l'objet type_ElementDocument_copie

```
CREATE TYPE type_ElementDocument_ntab AS TABLE
OF type_ElementDocument_copie;
/
```

Création d'un type type_ElementDocument avec un attribut ElementDocument_ntab de type type_ElementDocument_ntab

```
CREATE TYPE type_ElementDocument AS OBJECT
(numdoc number(6,0),
ElementDocument_ntab type_ElementDocument_ntab
);
```

- **création des tables**

Création de la table tab_ElementDocument qui contient une colonne de type NESTED TABLE :

```

CREATE TABLE tab_ElementDocument OF type_ElementDocument (PRIMARY KEY
(numdoc))
NESTED TABLE ElementDocument_ntab STORE AS
Copie_ElementDocument_ntab;

```

6. Transformation dans le cas de l'héritage

- **Principe :** Dans le cas d'une association de type héritage, nous associons pour chaque classe un type. La relation d'héritage est conservée entre les types créés.
- **Diagramme de classes :**

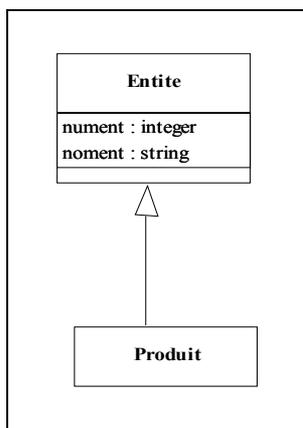


Figure 92 : Diagramme de classes utilisant un lien d'héritage.

- **Schéma de transformation**

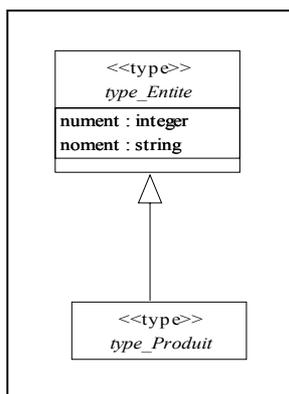


Figure 93 : Schéma de transformation d'un diagramme de classe utilisant l'héritage

- **Scripts SQL :**
 - **création des types :** Oracle 10g supporte la notion d'héritage entre types. Le mot clé UNDER désigne un type qui hérite d'un autre type. Le mot clé

FINAL désigne qu'aucune autre classe ne peut hériter de cette classe. La syntaxe de création est la suivante :

```
CREATE TYPE type_Entite AS OBJECT
  (nument NUMBER (6,0) ,
   noment varchar2(20))
  NOT FINAL;
```

```
CREATE TYPE type_Produit UNDER type_Entite
  (nomprd varchar2(20))
  FINAL;
```

- **création des tables** : Il existe deux méthodes pour la création de tables dans le cas de l'héritage :

- **Méthode 1** : Le modèle horizontal : à chaque type nous associons une table Objet-Relationnelle. La syntaxe de création des tables est la suivante :

```
CREATE TABLE nom_table OF nom_type;
```

- **Méthode 2** : Le modèle plat : Une seule table qui se base sur tous les types de données utilisateur. La syntaxe de création de la table est la suivante :

```
CREATE TABLE nom_table OF nom_super_type;
```

Le nom_super_type désigne le type le plus élevé dans la hiérarchie des tables.

Comparaison des deux méthodes :

	Avantages	Inconvénients
Le modèle horizontal	l'accès aux données est facile et direct.	L'inconvénient de ce modèle est que les requêtes de type « SELECT * from View » nécessitent la représentation de l'opérateur UNION à travers toutes les tables en question et la projection des rangées à travers seulement les colonnes spécifiées.
Le modèle plat	<ul style="list-style-type: none"> - Une vision globale de la base de données. - Simple et ne pose aucun obstacle au support d'index et contraintes. - Le modèle est clair et extensible. 	<ul style="list-style-type: none"> - Manipulation pénible de données avec le langage de script SQL lors de l'accès aux champs masqués. - Une seule table ne peut pas contenir plus que 1000 colonnes : c'est le nombre limite de colonnes par table.

Figure 94 : Comparaison entre le modèle horizontal et le modèle plat.

D'après la Figure 95 et bien que le modèle plat présente quelques inconvénients par rapport au modèle horizontal, il permet d'avoir une représentation meilleure de la base de données donc une vision globale ; il est caractérisé surtout par sa clarté et son extensibilité. Ainsi c'est ce type de modèle plat qui a été adopté dans notre travail. Une fois nos modèles objets relationnel obtenus, il suffit de les écrire sous Oracle10g afin d'obtenir du code SQL3.

IV.2.4. Architecture de P2M2

Après avoir présenté dans les paragraphes précédents, d'une part, les choix technologiques de développement et de déploiement et, d'autre part, les différentes techniques de transformation de nos modèles en Oracle 10g, nous présentons dans le présent paragraphe l'architecture de notre système d'information dédié à la mémoire de projet. Cette architecture multi-tiers est composée de quatre niveaux. Un tel choix permet d'atteindre différents objectifs à savoir :

- Déploiement simple, mise à jour aisée donc les coûts sont diminués ;
- Factorisation de la logique de l'entreprise : la logique qui existe dans une application est séparée du code de gestion des événements utilisateur et celui d'affichage des éléments graphiques. Ceci favorisera le pouvoir de réutiliser cette logique dans d'autres applications. Cette logique est appelée logique métier.
- Délégation des aspects techniques de transformation (mapping objet-relationnel, gestion des transactions) à des composants spécialement développés.

Par conséquent l'architecture multi-niveaux est en parfaite adéquation avec nos besoins et les nouveaux environnements d'exécution d'applications, à savoir les serveurs d'applications. Néanmoins on a voulu ajouter à ces objectifs l'expérimentation et l'enrichissement de nos connaissances dans ces nouvelles architectures. Les niveaux sont :

- **la base de données** qui est supportée par le SGBDOR Oracle10g. Naturellement Oracle10g supporte les types objets, il permet aux développeurs d'application d'accéder directement aux structures à travers leurs applications. L'abstraction d'objet et l'encapsulation des comportements d'objet permettent d'avoir des applications plus faciles à comprendre et à maintenir [Abbey, 2005]. Ce premier niveau est responsable de :

- La définition du type de hiérarchie de la base en utilisant les caractéristiques du serveur de type objet-relationnel (type d'hiérarchie, type de références, nested tables...);
- La sauvegarde des données dans une seule table d'objet relationnel appelée table « entity_tab »;
- Le filtrage des données en utilisant des vues d'objets associées aux déclencheurs de la base de données « Instead Of ... ».

- **le conteneur EJB**³⁹ qui fournit l'environnement d'exécution pour les EJB qui encapsulent la logique métier. Un conteneur J2EE offre un support d'exécution aux composants applicatifs. Il fournit des API qui sont à utiliser pour accéder aux services. Il peut aussi s'occuper de la sécurité, du groupement des ressources, de la gestion de cycle de vie de composants et des services de nommage et de gestion de transactions. Il existe un EJB par vue d'objet et un EJB de type session.

- **Le conteneur web** qui fournit les services de présentation. à travers des JSP (Java Server Page).

- **le client** composé d'un simple navigateur web.

³⁹ EJB : Entreprise Java Beans

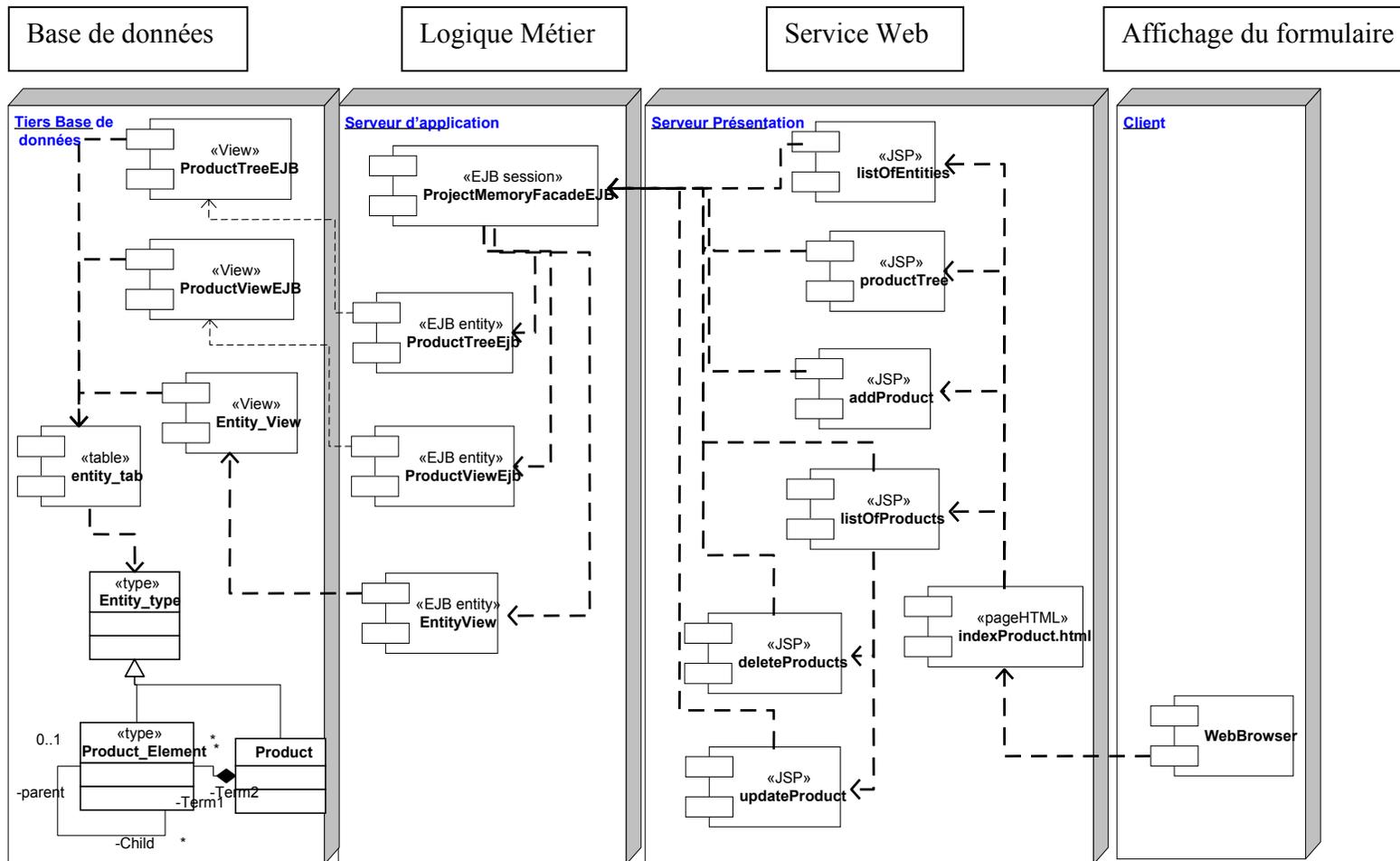


Figure 95 : Modèle d'architecture multi-niveaux d'application J2EE de la mémoire de projet.

Du point de vue opérationnel, cette architecture va permettre à tout utilisateur de formuler à travers un navigateur (Client léger) tout besoin lié au produit, document, justification, évolution et variante. Les données ainsi saisies s'adresseront au serveur de présentation qui les reliera au serveur d'application afin d'invoquer la logique métier spécifique à son besoin. A terme, ceci mettra en œuvre la couche de données afin de répercuter les différentes opérations sur le système d'information.

IV.2.5. P2M2 : Fonctionnalités et interfaces

P2M2 (Project Product Memory Management system) est un outil dédié à la saisie et la sauvegarde des données et des informations relatives à des projets de conception ainsi qu'à la consultation et la réutilisation de l'historique de projet. Notre outil P2M2 dispose de deux catégories de fonctionnalités:

- Mémorisation de projets ;
- Réutilisation de la mémoire de projet.

1) Fonctionnalité de description de produit

A travers une interface appropriée, l'utilisateur introduit l'ensemble des données et informations relatives au produit. Notre outil permet d'ajouter, supprimer et mettre à jour des produits. On peut afficher les produits par liste (Figure 97) ou par détail d'arborescence d'un produit (Figure 98).

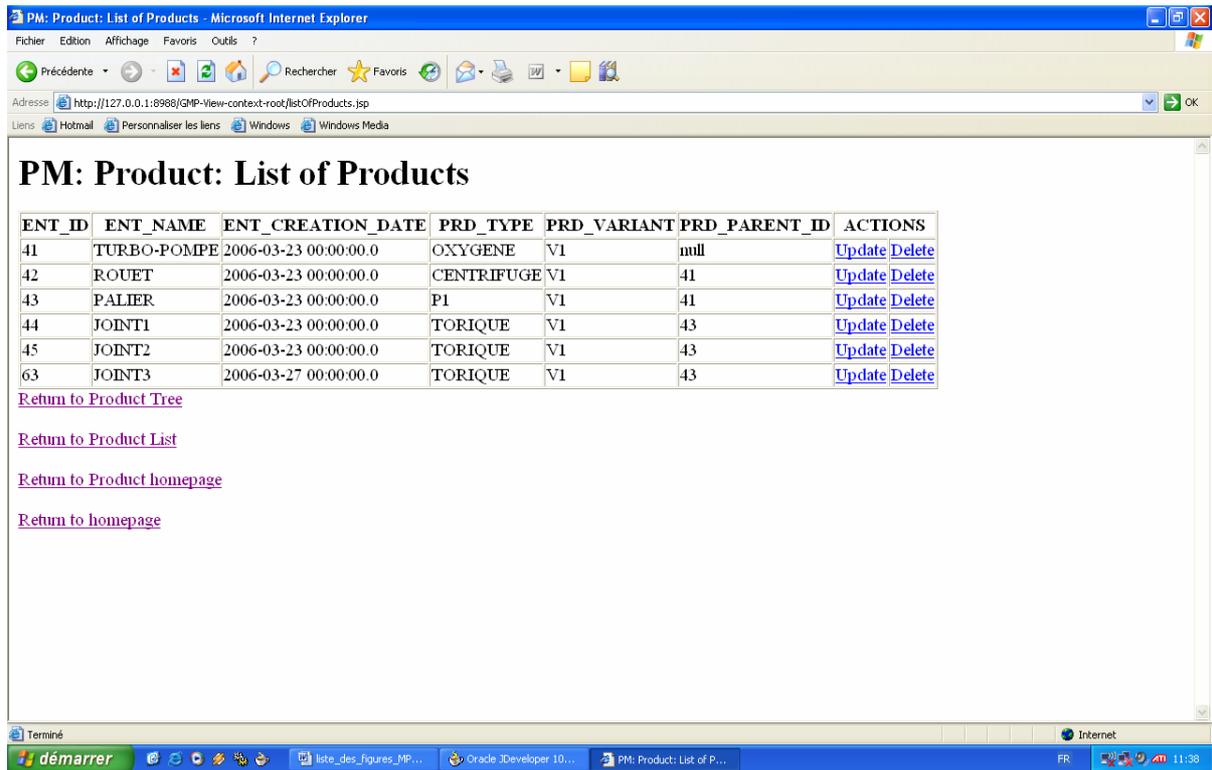


Figure 96 : Interface de description de la liste des produits

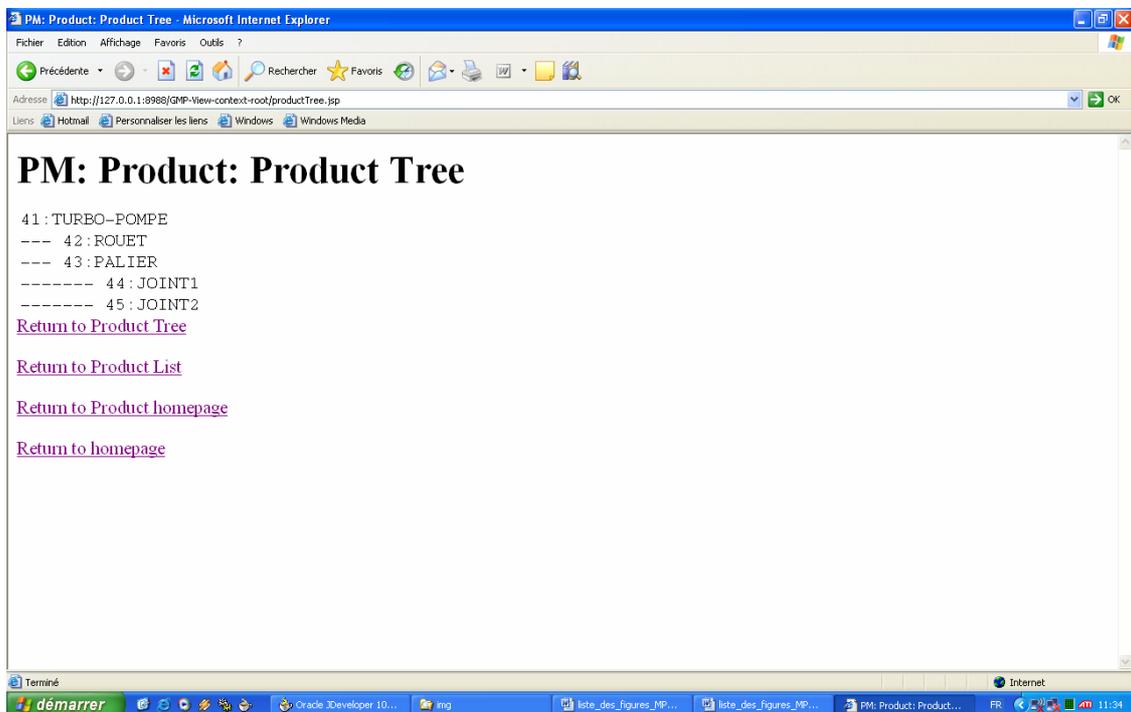


Figure 97 : Interface de description de l'arborescence du produit « Turbopompe ».

La Figure 97 présente la liste des produits archivés dans la base de la mémoire de projet. Chaque produit dispose d'un numéro (ENT_ID) attribut hérité à partir de la classe Entité, nom du produit (ENT_NAME), date de création (ENT_CREATION_DATE), type de produit (PRD_TYPE), PRD-VERSION (variante d'un produit), le numéro du produit parent s'il existe (PRD_PARENT_ID) et le type d'opération qu'on peut faire (suppression, mise à jour). La Figure 98 illustre la structure arborescente d'un produit « Turbopompe ».

2) Fonctionnalité de description de document

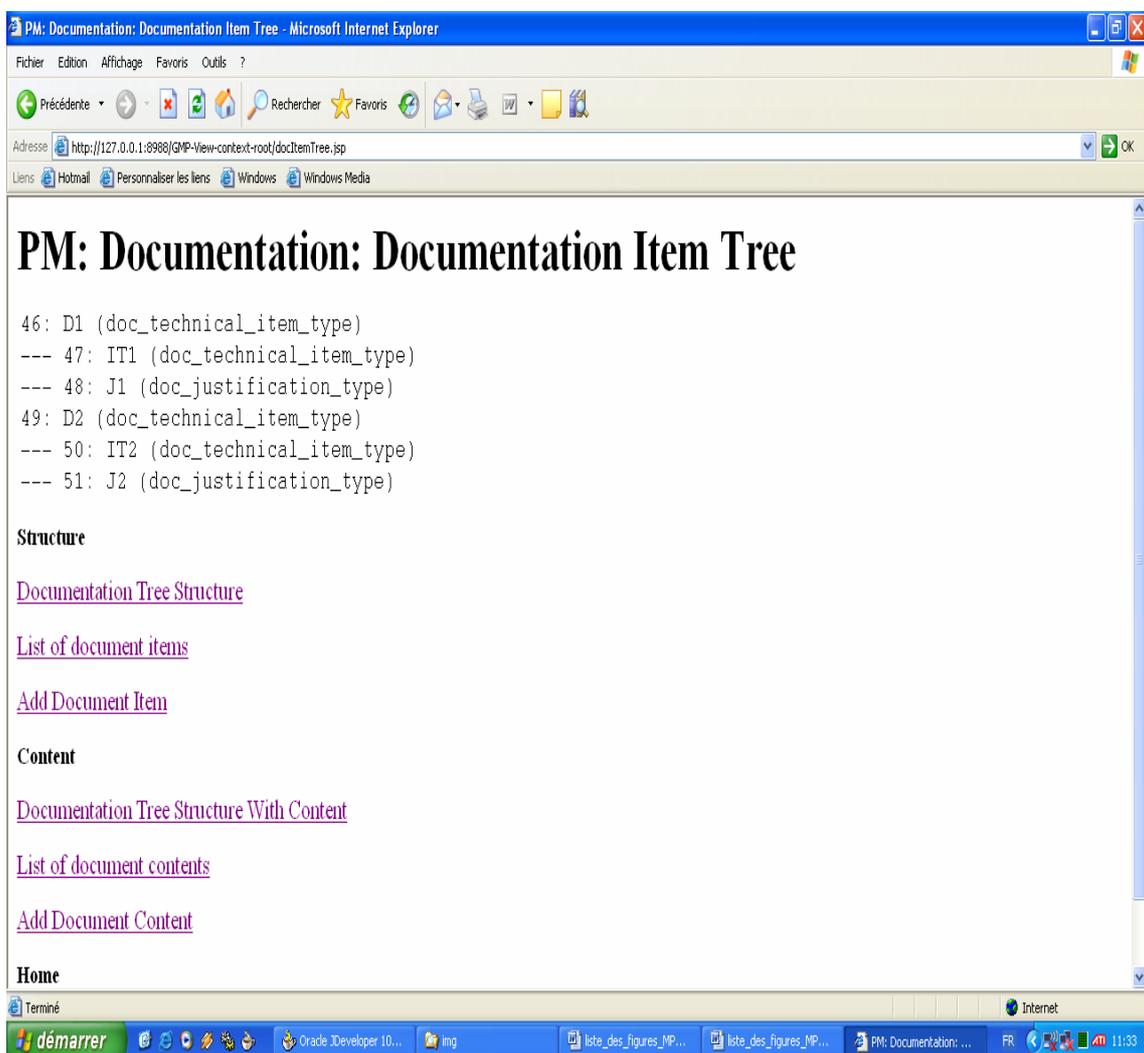


Figure 98 : Interface de description de l'arborescence d'un document.

A travers l'interface de la Figure 99, il est possible avec notre outil P2M2 d'afficher d'une manière structurée les éléments constituant un document, ceci après avoir saisi l'ensemble des données et informations relatives aux documents de projets à mémoriser. Ainsi sur la Figure 100 on remarque la description arborescente de deux documents D1 et D2. Le document D1 (resp. D2) est composé d'un document technique IT1 (resp. IT2) et d'un document de justification J1 (resp. J2).

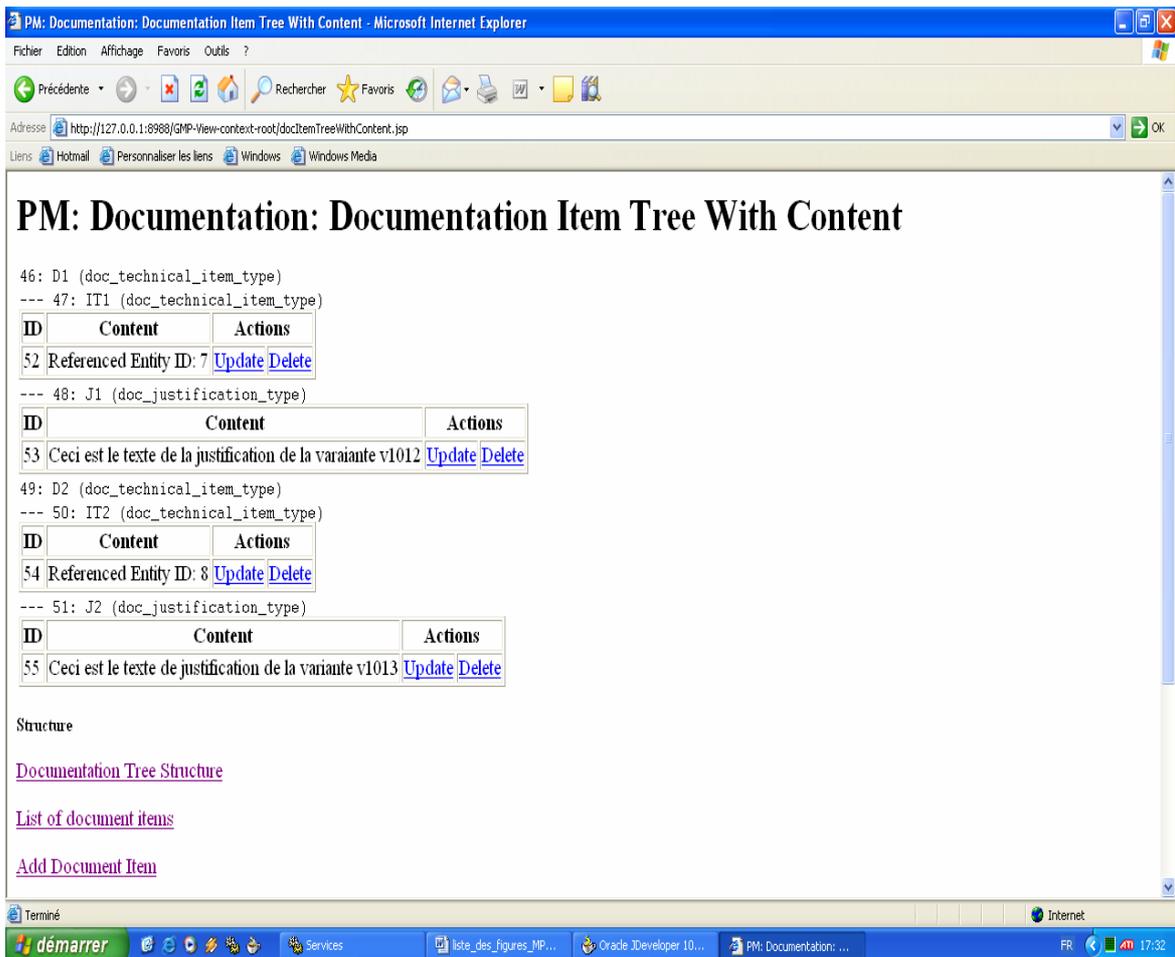


Figure 99 : Interface de gestion de l'arborescence d'un document.

Il est aussi possible avec l'outil P2M2 de gérer le contenu des documents (Figure 100) sachant que leur structure peut être modifiée à travers les variantes.

D'après la Figure 100, le document D1 (resp. D2) est composé d'une part d'un élément technique IT1 (resp. IT2) ayant comme numéro 10 (resp. 11) référencé par l'entité de numéro 7 (resp. 8) et d'autre part d'un élément de Justification J1 (resp. J2) dont le contenu est un texte de justification de la variante V1012 (resp. V1013).

L'interface de la Figure 100 nous permet d'ajouter un document, d'afficher les constituants d'un document ainsi que d'afficher la structure d'un document et de ses contenus.

Chaque document dispose d'un numéro (ENT_ID) attribut hérité à partir de la classe Entité, nom du document (ENT_NAME). Un contenu d'un document est décrit aussi par un numéro (ENT_ID) et le nom du contenu (ENT_NAME).

3) Fonctionnalité de gestion de variante

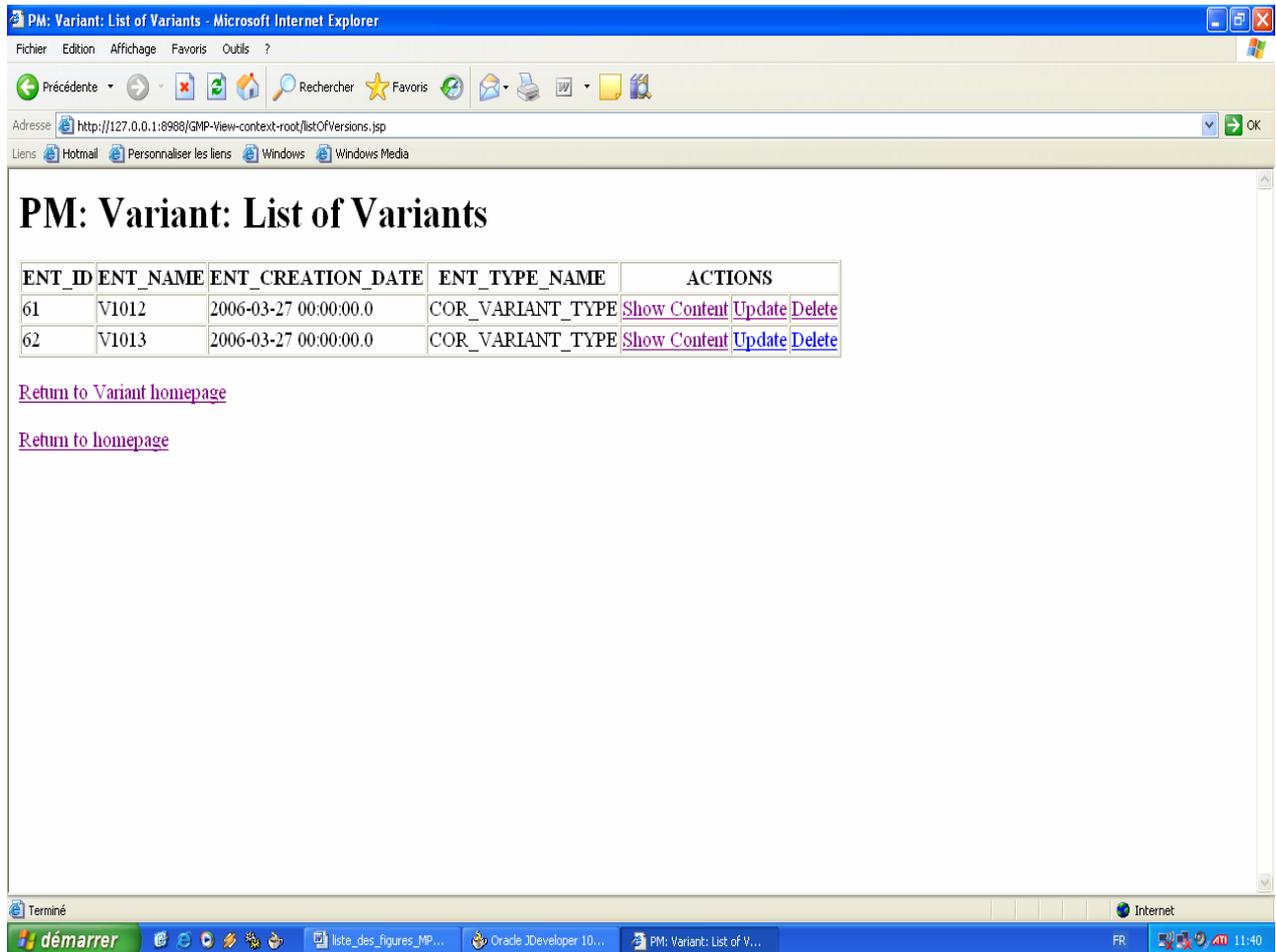


Figure 100 : Interface de description des variantes.

P2M2 permet de construire et de gérer des variantes à partir des éléments de la mémoire de projet. Il permet dans une variante de référencer n'importe quel élément constituant de la variante.

La Figure 101 montre la possibilité de P2M2 d'afficher la liste des variantes existant dans la mémoire de projet. Dans ce cas de figure, on est en présence de deux variantes à savoir la variante V1012 et V1013. Ainsi la variante V1012 (resp. V1013) est identifiée par le numéro 7 (ENT_ID) (resp. 8), un nom (ENT_NAME), une date de création (ENT_CREATION-DATE), un nom de type (ENT_TYPE_NAME) et une action qui permet d'afficher le contenu de la variante, de la supprimer ou de la mettre à jour.

P2M2 permet d'afficher la structure d'une variante. La Figure 102 (resp. 103) illustre l'affichage du contenu de la variante V1012 (resp. V1013). Elle est constituée d'un certain

nombre de produits. Chaque produit est identifié par un numéro (ENT_ID), d'un nom (ENT_TYPE_NAME) et d'une action qui permet de supprimer un élément de la constitution de la variante.

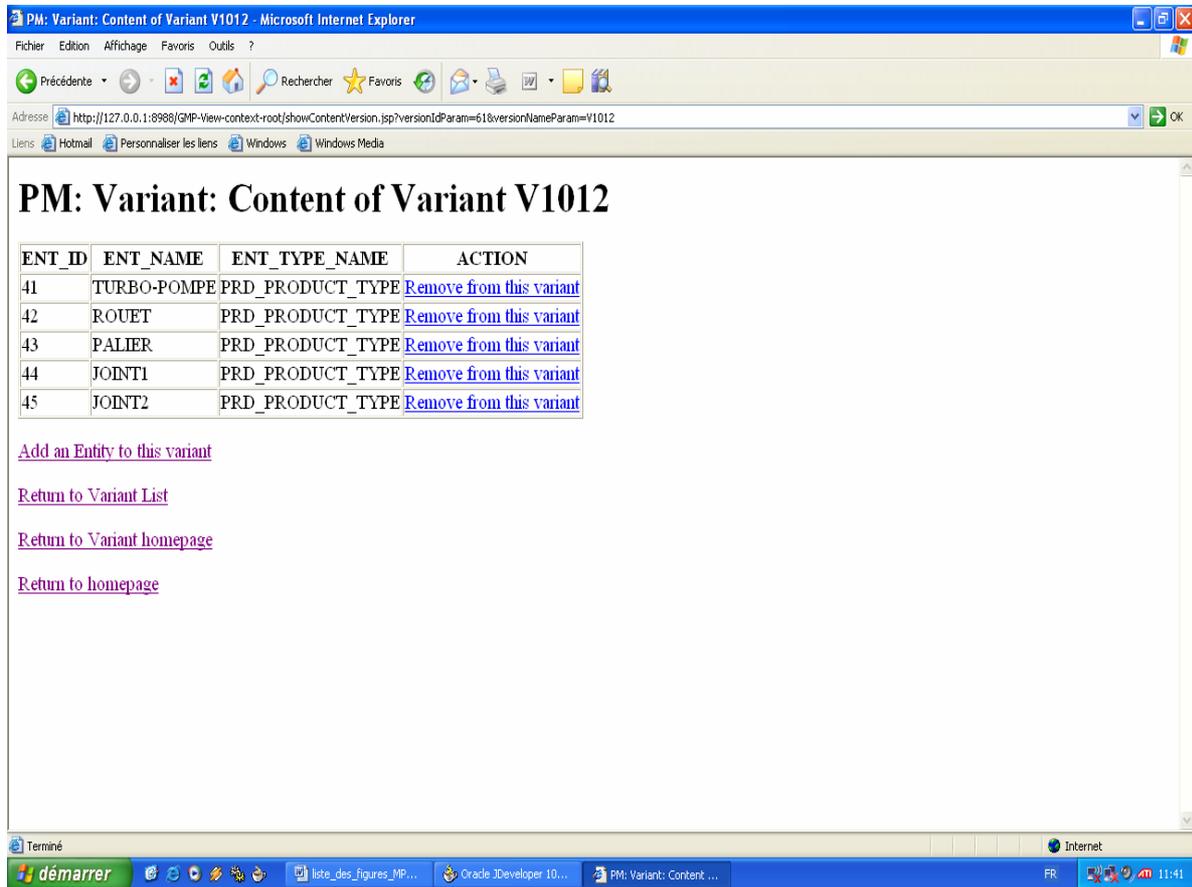


Figure 101 : Interface d'affichage de la variante V1012.

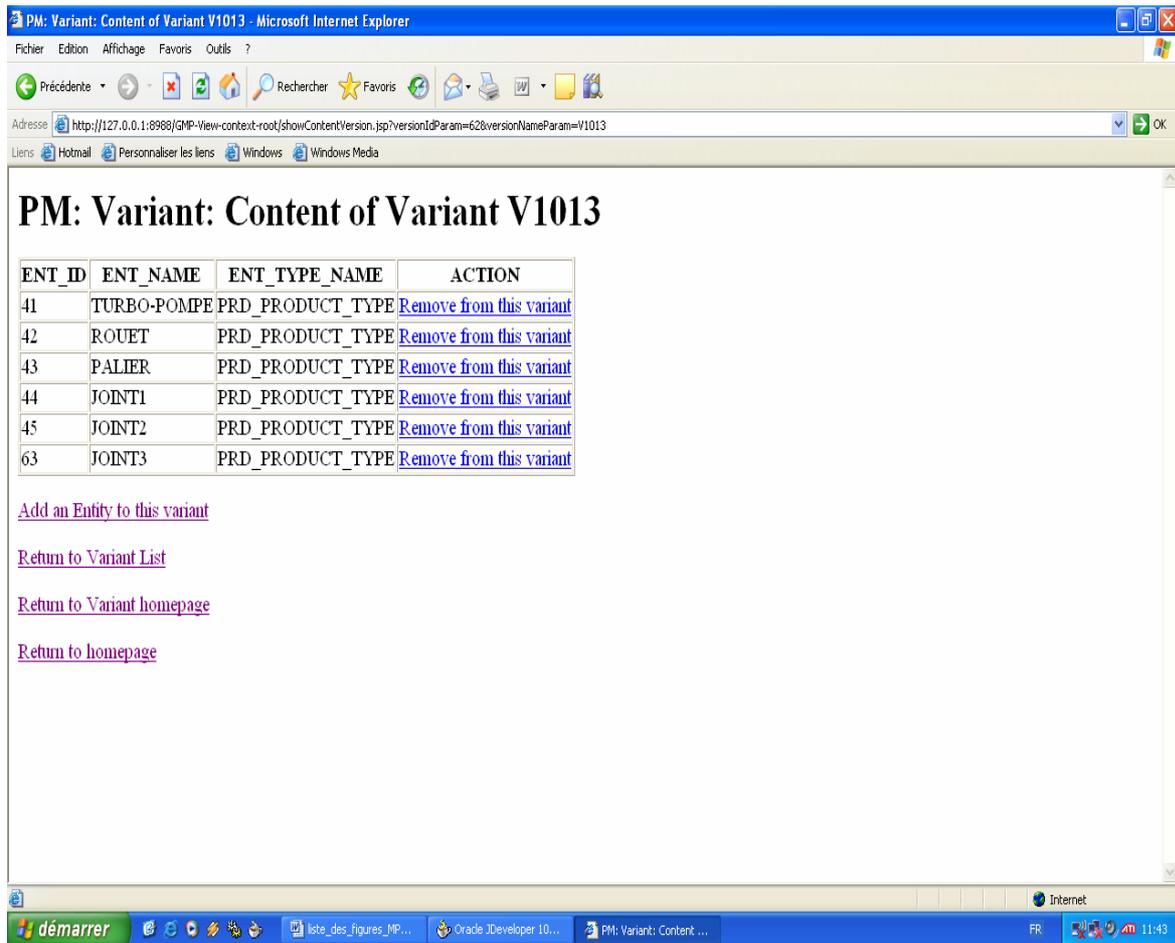


Figure 102 : Interface d’Affichage de la Variante V1013.

4) Fonctionnalité de gestion des Evolutions

P2M2 permet de construire et de gérer des évolutions des différents éléments de la mémoire de projet. Ainsi les variantes peuvent faire évoluer leur historique en référant des éléments dans l’évolution. Dans la Figure 108 on peut voir comment une variante ou un élément de document peut être déclaré comme une évolution.

La Figure 104 illustre les trois évolutions qui existent dans la mémoire de projet. Chaque évolution est caractérisée par un identifiant (ENT_ID), nom de l’évolution (ENT_NAME), date de création (ENT_CREATION_DATE), l’identifiant de la source dans l’évolution (SOURCE_ID), nom de l’élément source (SOURCE_NAME), l’identifiant de la cible (TARGET_ID), nom de la cible (TARGET_NAME) un ensemble d’action sur l’évolution du type mise à jour ou suppression.

Ainsi l’évolution E1 (resp. E2, E3) identifiée par le numéro 19 (resp. 20, 21) a comme élément source V1012 (resp. J1, D1) et comme élément cible V1013 (resp. J2, D2). Cette évolution a été créée le 06-06-2005 (resp. 06-06-2005, 06-06-2005).

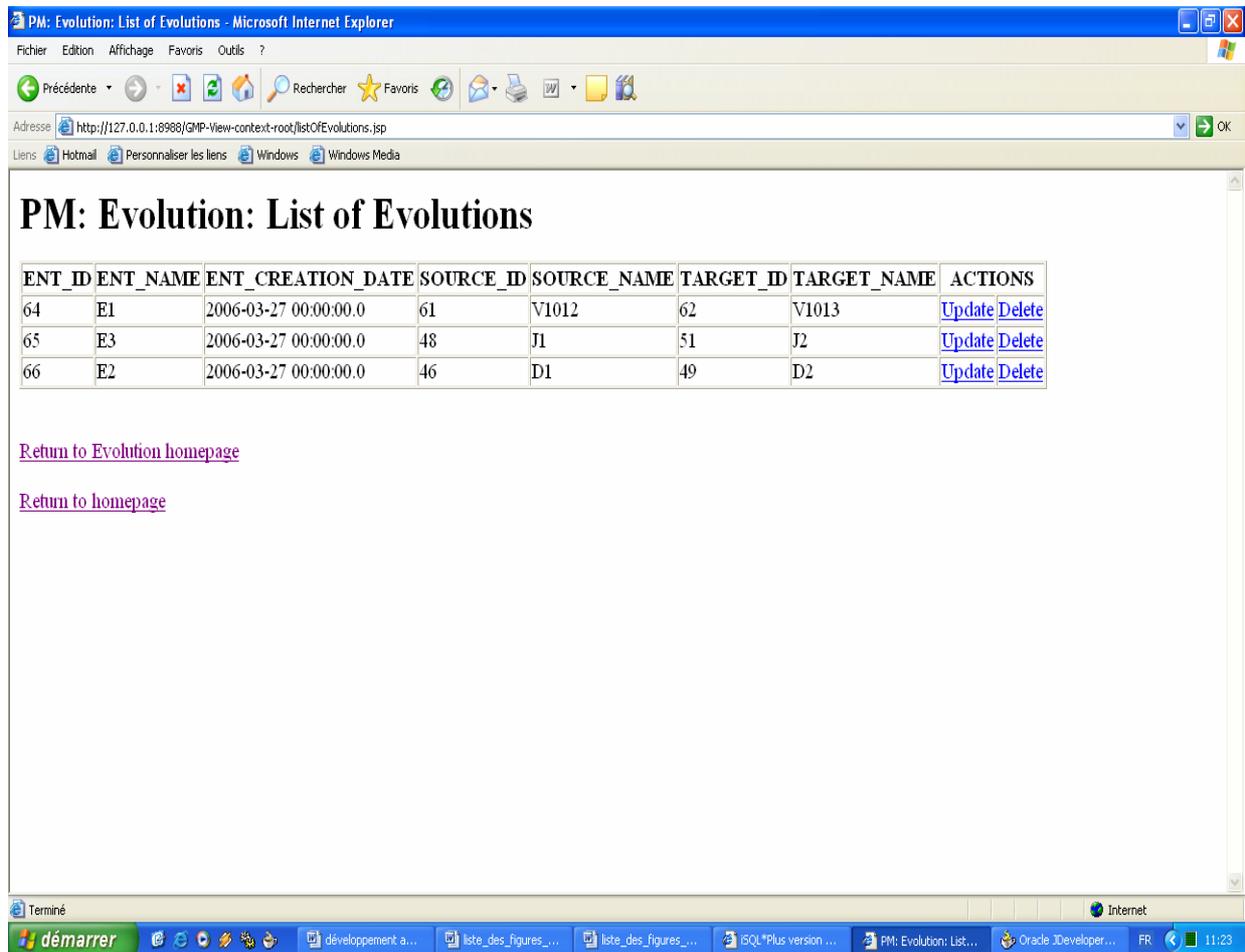


Figure 103 : Interface d’affichage de la liste des évolutions existant dans la mémoire de projet avec une description de chacune des évolutions.

IV.3. Validation des modèles auprès d’une entreprise d’ingénierie : Cas de la STUDI

IV.3.1. Présentation de la société STUDI

STUDI, Société Tunisienne d’Ingénierie, fondée en 1970 sous forme de société anonyme, est un bureau d’Ingénieurs Conseils indépendant. Grâce à une volonté de diversification, STUDI, devenue Groupe STUDI, intervient dans tous les domaines du développement économique et social et notamment : l’eau, l’environnement, le développement agricole, les infrastructures de transport, les aménagements côtiers, le bâtiment, l’économie et l’évaluation de projets et l’informatique. STUDI exerce son activité dans plus de trente pays et ceci pour le compte d’une clientèle diversifiée : les administrations, les services publics, les collectivités locales, les sociétés privées, les organismes internationaux d’aide au développement. Elle mène différents types de prestations :

- Conception, études techniques et maîtrise d’œuvre ;

- Pilotage et maîtrise d'ouvrage délégué;
- Etudes générales : programmation, schémas directeurs, études de faisabilité;
- Etudes institutionnelles ;
- Etudes d'impact ;
- Diagnostics-expertises ;
- Organisation technique, administrative et informatique des organismes et/ou exploitations ;
- Formation de personnel ;
- Assistance technique.

STUDI regroupe un certain nombre de sociétés. C'est la société STUDI, Ingénieurs Conseils au sein de laquelle s'est déroulée l'expérience d'adaptation des modèles de la mémoire de projet. STUDI, Ingénieurs Conseils a comme principale activité la conception et les études techniques dans les domaines du bâtiment, des ouvrages d'art, des infrastructures, du transport, de l'hydraulique urbaine et rurale, des ressources en eau, de l'environnement, des énergies conventionnelles et renouvelables, du développement agricole et rural, du développement urbain et de l'aménagement du territoire, études générales et économiques, assistance technique et supervision des travaux.

IV.3.2. Démarche pour la mise en place de la mémoire de projet chez STUDI

Afin de pouvoir mettre en place la mémoire de projet, une étude préliminaire au sein de la société STUDI a été menée qui a consisté à :

- Comprendre les activités du bureau d'étude : construction de bâtiments, de ponts, de routes etc. ;
- Identifier un projet cible : le projet retenu consiste à réaliser l'extension du siège de la banque centrale de Tunis ;
- Assister aux différentes réunions organisées par l'équipe qui mène le projet ;
- Identifier les éléments utilisés dans le projet pour l'expérience d'adaptation ;
- Faire un rapprochement entre ces éléments identifiés et les éléments qui existent dans les modèles de la mémoire de projet ;
- Dégager la différence et proposer les modifications nécessaires aux modèles dans un but de les adapter.

IV.3.2.1. Notion de Projet-Processus chez STUDI

Tout projet au sein de la STUDI passe par les étapes suivantes :

- Phase d'étude
- Phase travaux

- Phase cloture

Chaque phase d'étude est composée des différentes étapes suivantes :

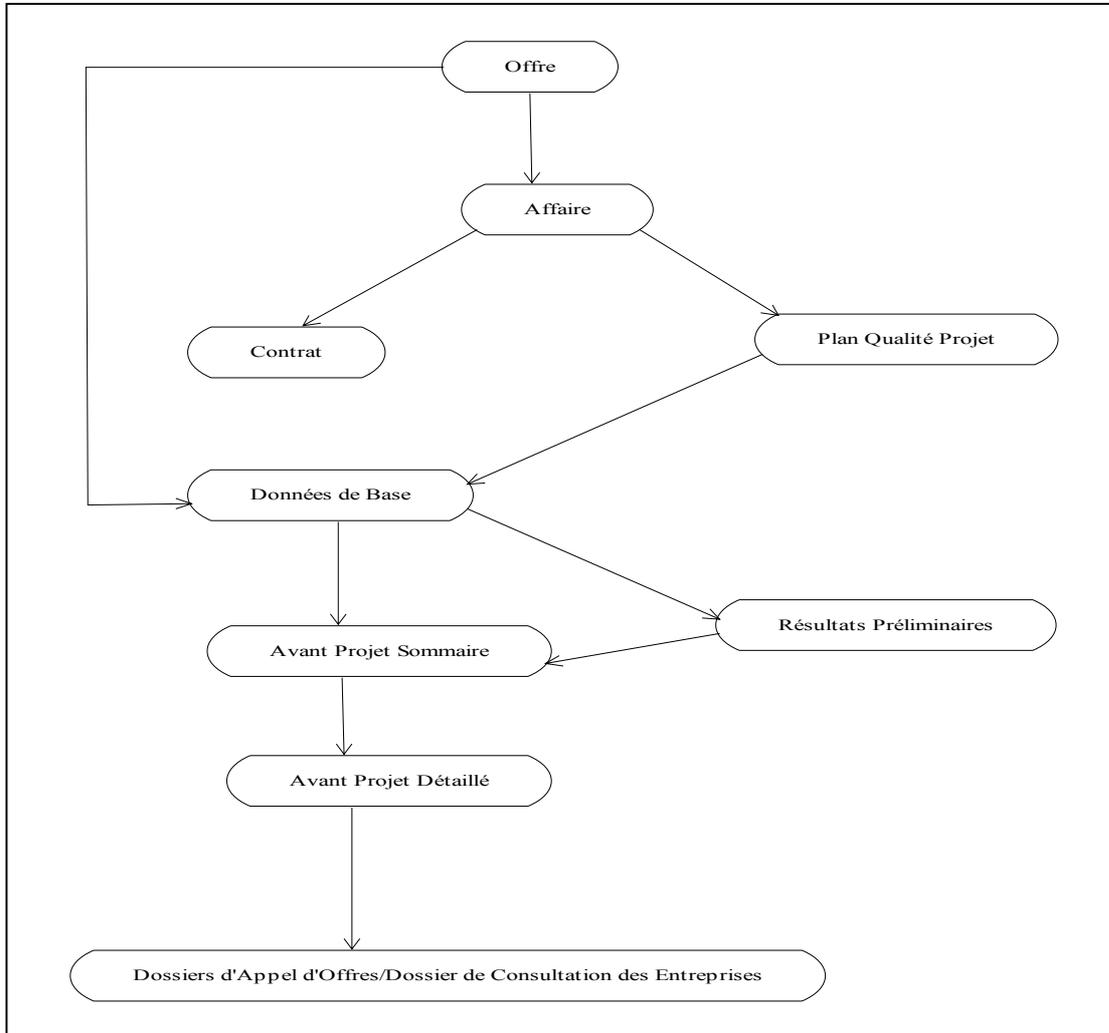


Figure 104 : Diagramme d'activités d'un Projet chez STUDI (Etape phase d'étude).

Ainsi on remarque que tout projet au sein de la STUDI est composé d'un certain nombre de phases.

PHASE 1 : Description de la phase Etude

- L'offre : contient le cahier de charges du produit, les services demandés et les conditions de déroulement de la consultation (contenu attendu de la réponse, délai de fourniture de la réponse, modalités de fourniture d'informations complémentaires, etc.). Les consultations sont faites après

l'étude préalable du projet en fournissant aux candidats la définition de la solution attendue obtenue dans cette étude.

- **Affaire** : définie sur le plan contractuel (gestion du contrat avec le client, gestion des ressources financières et matérielles) et qui est garante des coûts et des délais annoncés au client. Cette affaire consiste à l'extension du siège de la banque centrale de Tunis.

- **Contrat** : a pour objectif de définir les conditions et les modalités d'intervention du bureau d'étude (STUDI) dans l'opération d'extension du siège de la banque centrale de Tunis pour ce qui concerne les études la direction et le contrôle technique des travaux. Ce document contient :

- Les missions du bureau d'études,
- Le programme fonctionnel,
- La mise en forme du programme technique,
- Les délais,
- La rémunération du bureau d'études,
- Les responsabilités du bureau d'études,
-

- **PQP** : Plan Qualité Projet : Ce document définit l'ensemble des actions que la STUDI veille à respecter pour obtenir le niveau de qualité requis pour le système à construire ainsi que les procédures d'interface entre la maîtrise d'ouvrage et le maître d'œuvre. Ce plan qualité est validé par la maîtrise d'ouvrage pour tout ce qui concerne ses relations avec le maître d'œuvre, il comporte :

- Fiche des tâches,
- Check-list documents à produire,
- Planning des tâches,
- Devis interne,
- Check-list données d'entrées et interfaces,
- Fiche de suivi des décomptes des entreprises des lots spéciaux et
- Fiche de suivi des travaux des entreprises.

- **Données de base** comprennent :

- Plans d'architecture,
- Levés topographiques du terrain,
- Rapports géotechniques,
- Programme fonctionnel (climatisation, ventilation, groupe de secours...) et

- Budget prévisionnel.

- Résultats préliminaires : après analyse des données de base, un premier résultat est obtenu. Ces résultats servent de paramètres pour l'APS.

- APS (Avant Projet Sommaire) ; il contient :
 - Les grandes lignes de conception : exemple construction d'un bâtiment de huit étages avec un système de climatisation et de sécurité.
 - Les estimations sommaires des travaux : exemple besoin en quantité d'une matière, etc.

- APD (Avant Projet Détaillé) ; il contient :
 - Détails de conception
 - Pièces écrites : CCAP (Cahier des Clauses Administratives Particulières), CCTP (Cahier des Clauses Techniques Particulières)
- DAO (Dossier d'Appel d'Offres) / DCE (Dossier de Consultation des Entreprises)

Ce dossier contient:

 - Plans détaillés
 - Pièces écrites : CCAP (Cahier des Clauses Administratives Particulières), CCTP (Cahier des Clauses Techniques Particulières)

PHASE 2 : Travaux

La phase travaux consiste en :

- Réception du matériel nécessaire (planches, fondations, etc.)
- Réception des installations (électriques ...)
- Réception provisoire des travaux
- Réception définitive des travaux

PHASE 3: Clôture du projet

C'est la phase finale du projet qui consiste à définir :

- Les conditions de clôture du projet
- Les visas autorisant la clôture du projet

- Le bilan
- Les actions de suivi commercial recommandé
- La diffusion finale

Modèle de classes des concepts Projet-Processus chez STUDI

L'étude du concept de projet et la façon dont il est géré au sein de la société STUDI nous permettent d'élaborer le diagramme UML suivant :

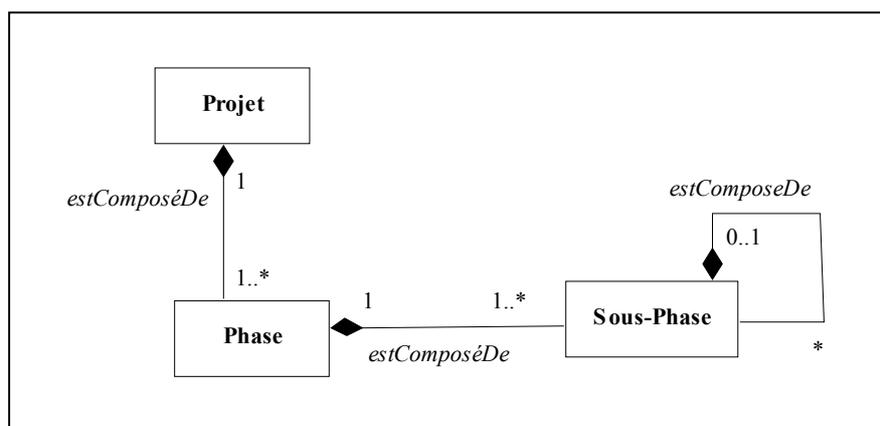


Figure 105 : Diagramme de description des notions Projet-Processus chez STUDI.

Ainsi en faisant une comparaison entre nos modèles présentés dans le Chapitre 3 et le modèle de la Figure 106, on remarque qu'il existe une grande similitude entre les modèles : nous gérons aussi la notion de projet mais la notion de phase et sous phase chez STUDI n'est autre que la notion de processus et étapes de processus dans nos modèles.

Un projet chez STUDI est composé d'un certain nombre de phases, par exemple les phases Etude, Travaux. Chaque phase est à son tour décomposée en un certain nombre de sous phases, exemple la phase Etude est composée des sous phases Offre, Affaire, Contrat etc.

Exemple : Diagramme objet des concepts Projet-Processus

Le diagramme d'objets de la Figure 107 illustre un exemple de projet d'extension de la banque centrale de Tunis mené par STUDI. Ce projet est décomposé en phases et en sous-phases.

Ainsi le projet « extension de la BCT » est décomposé en certain nombre de phases à savoir la phase « Travaux » (PH2), la phase « étude » (PHA1) et la phase « clôture » (PHA3). Cette dernière est à son tour décomposée en sous phases « offre » (SP1), « affaire » (SP2), « contrat » (SP3), et « avant projet sommaire » (SP4).

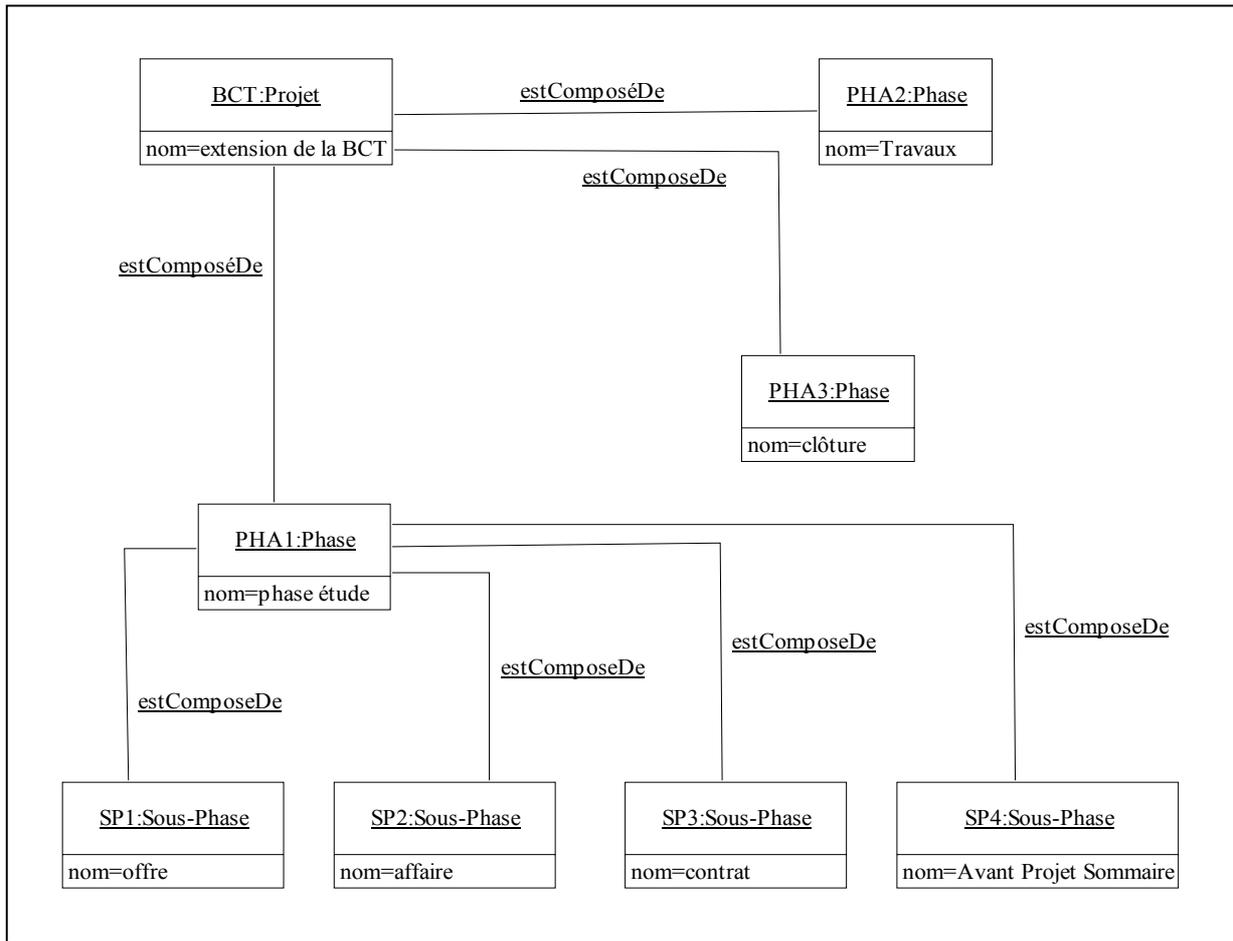


Figure 106 : Diagramme objet du concept Projet chez STUDI.

Le diagramme d'objets de la Figure 107 présente l'exemple du projet choisi au sein de la STUDI pour mener notre adaptation de modèles à savoir l'extension de la banque centrale de Tunis.

IV.3.2.2. Notion de Produit chez STUDI

Dans le cas de notre expérience d'adaptation des modèles au sein de la STUDI et du projet « extension de la banque centrale », le produit identifié est « la banque centrale ». Ce produit est constitué d'un certain nombre de lots à savoir le lot fluide, lot fondation profonde, lot ascenseur, lot structure, lot équipements cuisines, lot structure, lot réseaux divers, lot électricité, et lot sécurité incendie. Une telle étude nous permet d'élaborer le diagramme UML suivant :

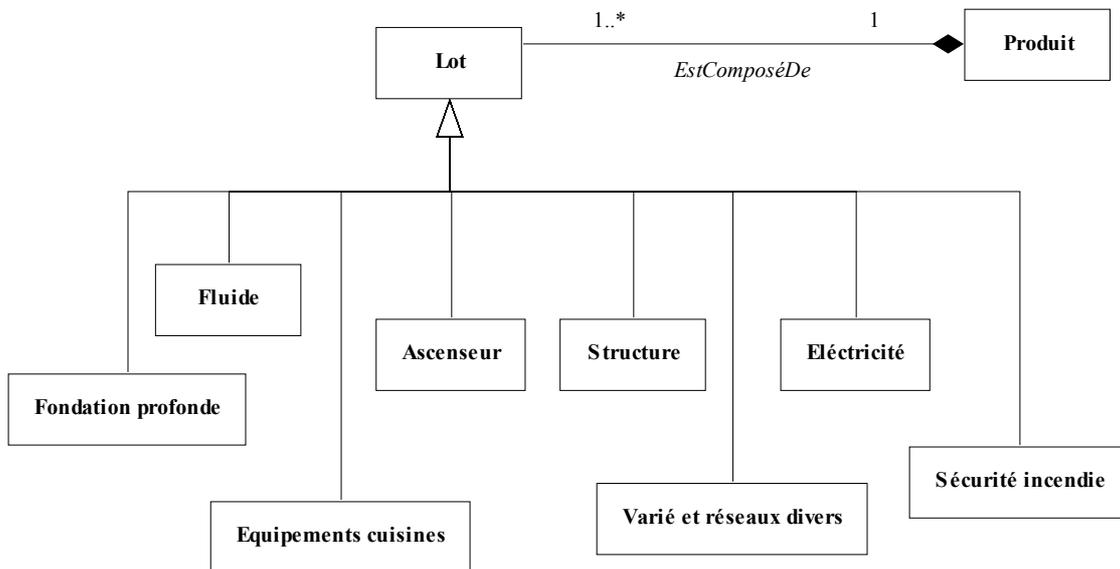


Figure 107 : Diagramme de description du concept Produit chez STUDI.

Ainsi en faisant une comparaison entre nos modèles présentés dans le Chapitre 3 et le modèle de la Figure 108, on remarque qu'il n'existe pas une grande différence entre les modèles. Par exemple et au niveau de la classe ElementProduit élément de nos modèles développés dans le Chapitre 3, elle est appelée la classe Lot dans le cas des modèles de la STUDI. Il ya eu aussi l'ajout d'une généralisation dans le modèle de description du concept Produit chez STUDI.

Exemple : Diagramme d'objet

Le diagramme d'objet de la Figure 109 a comme objectif de montrer la décomposition du produit « extension de la banque centrale » en éléments du produit « alimentation en eaux », « évacuation des eaux », « climatisation » etc.

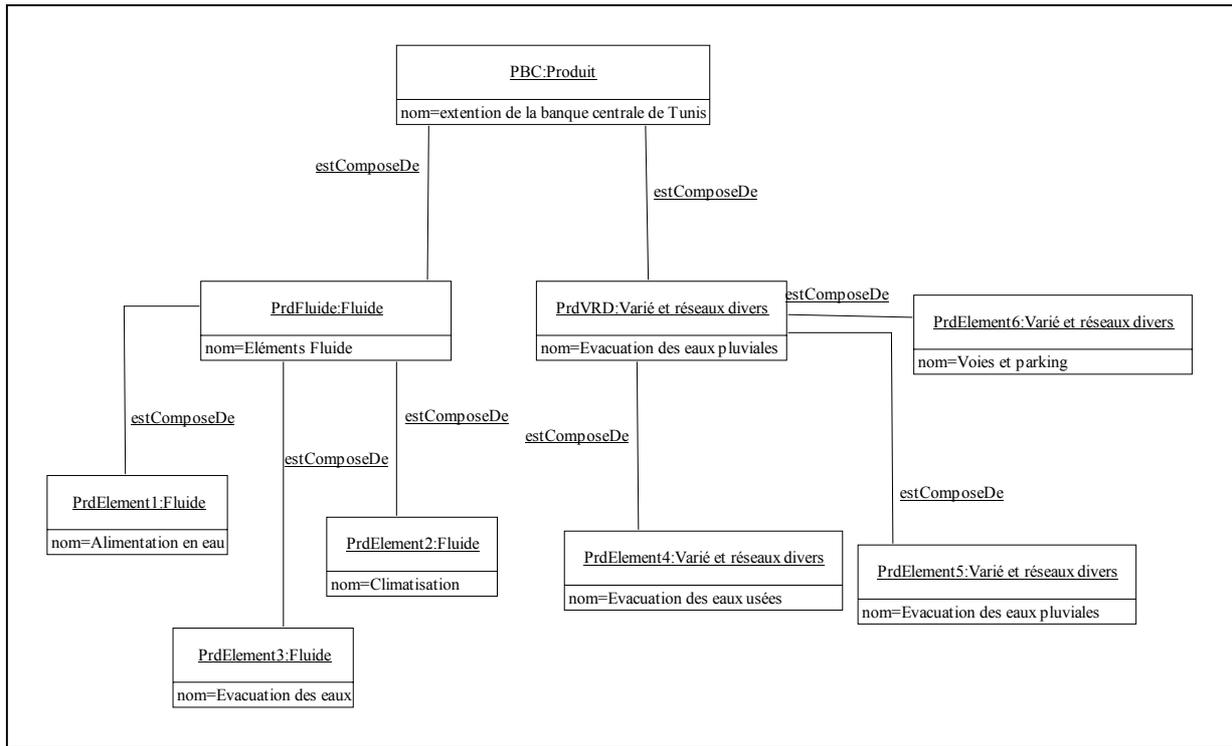


Figure 108 : Diagramme objet du concept Projet chez STUDI.

IV.3.2.3. Notion d'Organisation chez STUDI

Parmi les modèles d'adaptation chez STUDI, nous avons considéré aussi le modèle d'organisation. Comme exemple d'organisation, on avait pris l'exemple du projet « extension de la banque centrale », dont le maître d'œuvre est la Banque Centrale. Les éléments d'organisation qui ont été identifiés sont le bureau d'étude, le bureau d'architecte, le bureau de contrôle, le bureau du décorateur, le maître d'ouvrage délégué et l'entreprise(s) de travaux (Génie civil, Electricité, Fluide etc.). Ces bureaux d'étude et d'entreprise constituent les éléments d'organisation par rapport à notre modèle d'organisation présenté dans le chapitre 3.

Dans ce cas d'adaptation nous avons remarqué que notre modèle d'organisation s'adapte parfaitement au cas de la STUDI.

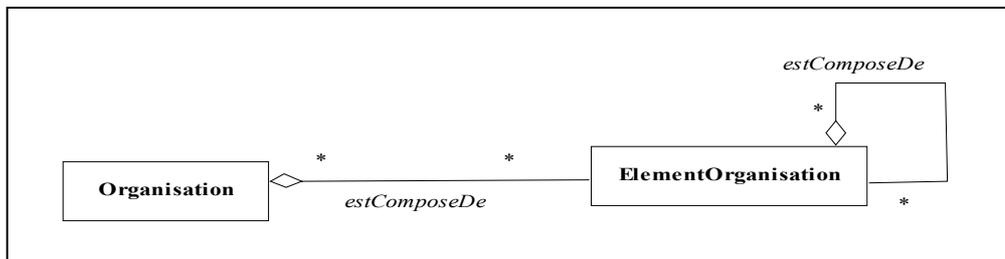


Figure 109 : Diagramme de description du concept Organisation chez STUDI.

Exemple : Diagramme d'objet

Le diagramme d'objet de la Figure 111 illustre l'exemple du projet « extension de la banque centrale » et les bureaux d'étude intervenant dans la réalisation de ce projet.

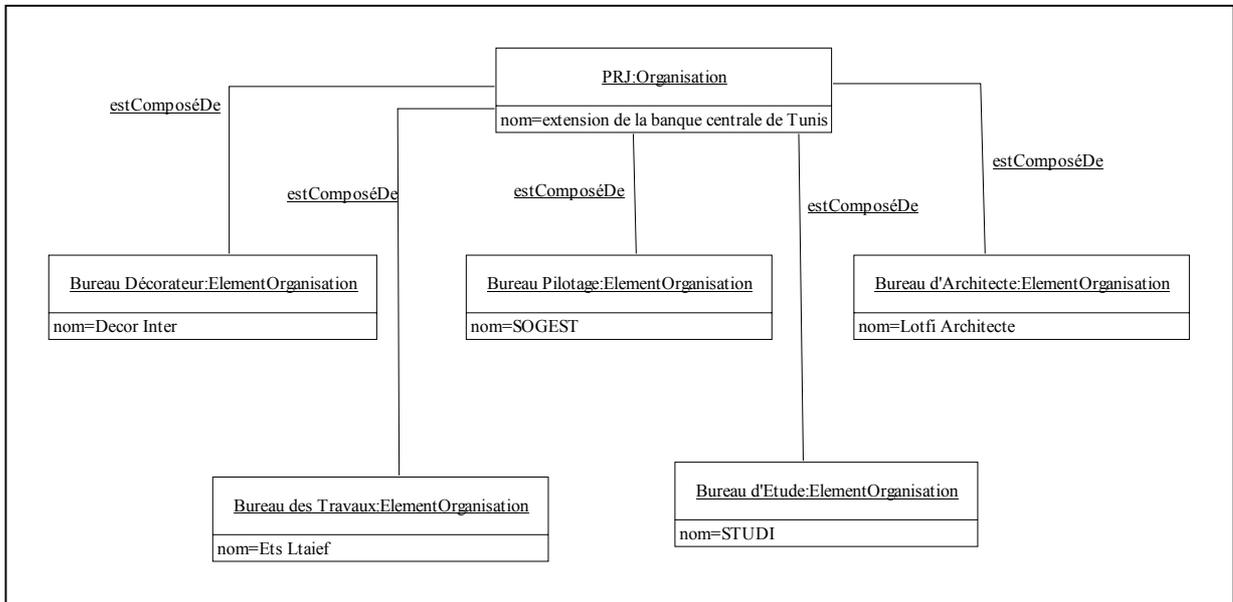


Figure 110 : Diagramme objet du concept Projet chez STUDI.

IV.3.2.4. Bilan d'adaptation des modèles chez STUDI

Ainsi comme nous venons de le voir, la procédure d'adaptation des modèles sur le projet d'extension de la banque centre de Tunis, mené par au sein de la société d'ingénierie STUDI n'a pas révélé beaucoup de changements au niveau de nos modèles. Ces changements consistent principalement en des appellations différentes de classes ayant le même objectif telle que la classe Phase chez STUDI qui est appelée Processus dans nos modèles, l'ajout de relation telle que la généralisation dans le modèle de produit, ajout de classe telle que la classe Lot. Nous avons aussi remarqué que la notion d'archivage de projets dans des documents n'est pas dans un but de réutiliser ces documents dans des projets futurs mais juste des documents prouvant la réalisation de projets. La réutilisation dépend de l'expérience acquise de chacun des acteurs intervenant dans un projet après avoir vécu et participé à un certain nombre de projets similaires dans le passé. Les intervenants dans un projet se basent sur leur savoir-faire et leur expérience dans le passé.

A remarquer, que la procédure d'adaptation n'avait concerné que les modèles, quant à l'adaptation du prototype développé n'a pas été traité. Ainsi une telle adaptation de l'application sur l'exemple de la STUDI, pourra éventuellement poser des problèmes qui peuvent aller jusqu'à la réécriture du code source des différents modules de l'application.

Suite à ses remarques d'adaptation des modèles, une perspective s'ouvre dans l'élaboration des modèles de la mémoire de projet. Elle consiste à faire un modèle beaucoup plus général de point de vue abstraction. Ce qui revient à dire l'élaboration d'un métamodèle pouvant répondre a une

telle procédure d'adaptation des modèles au sein de la STUDI moyennant des techniques d'instanciation du métamodèle afin d'obtenir les modèles souhaités.

La technologie MDA (Model Driven architecture) [OMG,2005] pourra être une réponse à une telle perspective de métamodélisation du fait que dans de telle technologie nous gérons les modèles, les métamodèles et les métamétamodèles avec des techniques d'instanciation et d'abstraction qui permettent de passer d'un élément de modélisation à un autre.

IV.4. Conclusion

L'objectif principal de ce chapitre est la présentation de la maquette d'application et l'application de nos modèles sur un cas d'entreprise.

Concernant le premier objectif et avant de présenter la maquette, des choix de type technologiques et des modèles à implémenter ont été faits. Nos choix se sont basés sur une plateforme du type J2EE avec comme gestionnaire de bases de données Oracle 10g. Ils ont prouvé la faisabilité de la démarche proposée. Nous avons choisi quelques modèles les plus significatifs qui montrent quelques propriétés que doit satisfaire une mémoire de projet à savoir la structure des éléments, le versionnement, la justification et les évolutions des éléments constituant la mémoire de projet. Le déploiement d'une telle architecture a nécessité des transformations de nos modèles UML en objet relationnel. Quelques captures d'écran du prototype développé ont aussi été présentées qui montrent les fonctionnalités du système développée.

L'adaptation des modèles auprès d'une entreprise d'ingénierie nous a permis d'expérimenter et de valider nos modèles. Nous avons constaté une grande similitude entre nos modèles et un cas réel tel que celui de la STUDI. Les modifications dans nos modèles n'étaient pas aussi importantes. Elles étaient du type ajout d'une généralisation, le changement de nom d'une classe par un autre nom. Dans un objectif de ne pas réécrire les modèles pour un domaine particulier, l'idée est de remonter en niveau d'abstraction. Cela consiste à développer un métamodèle dédié à la mémoire de projet; L'architecture MDA pourra être d'un apport important pour l'élaboration de tel métamodèle. Pour se faire, il faudra concevoir un profil du type mémoire de projet dont les modèles seront situés au niveau M2 de l'architecture. L'élaboration de profil dédié à la mémoire de projet d'une part et moyennant les techniques de transformation à concevoir et à mettre en œuvre d'autre part pourront lors d'une étape d'instanciation du profil à générer les modèles spécifiques souhaités de la mémoire de projet du type par exemple Génie civil, Mécanique etc. au niveau M1 de l'architecture MDA.

CONCLUSION

Dans cette thèse nous avons présenté une démarche pour la capitalisation de la connaissance liée à la conception de produit avec comme principal objectif la réutilisation des expériences passées. Pour ce faire, nous avons défini la mémoire de projet pour la capitalisation des connaissances de projet en conception. Son architecture n'est pas dédiée à un domaine particulier, mais elle a été élaborée plus généralement pour permettre l'archivage des informations et la traçabilité des historiques de projet de conception de produits. Nous résumons ici les résultats que nous avons obtenus. Nous présenterons par la suite les possibilités de développements futurs.

Dans l'état de l'art des concepts du KM présenté dans le Chapitre 1 nous avons étudié les différents concepts liés au KM, les difficultés et l'intérêt d'avoir une politique de KM dans une entreprise. D'autre part et dans une entreprise, pour réussir la capitalisation des connaissances il faut tenir compte de tous les aspects organisationnels, humains, technique etc. et de ne pas réduire le problème de la capitalisation à un problème humain seulement par exemple (ou technique). Les techniques d'ingénierie des connaissances nous ont montré qu'ils sont d'une grande utilité pour la formalisation des connaissances dans une activité de conception. Ainsi la capitalisation des connaissances dans le domaine de la conception ne pouvait se faire sans l'intégration d'un modèle de processus de conception.

Dans un but d'atteindre notre objectif, à savoir la modélisation et la mise en œuvre d'une mémoire de projet dans le domaine de conception, nous avons présenté dans le chapitre 2 un état de l'art sur les méthodes de formalisation des connaissances afin de retenir la méthode la plus adéquate et qui répond au mieux à nos besoins de modélisation. Une réponse à nos besoins en mémoire de projet a été les SGDT (Système de Gestion des Données Techniques), solution qu'on avait démontré insuffisante du point de vue objectifs de la mémoire de projet du fait qu'elle ne couvre pas toutes les fonctionnalités que doit avoir une mémoire de projet. A titre d'exemple les processus gérés dans de telle solution ne sont que des processus du type fabrication. Par contre dans le cas de la mémoire de projet s'on s'intéresse au processus du type conception.

Un outil de formalisation des connaissances dans une mémoire de projet est nécessaire. On avait démontré qu'aucune des méthodes de formalisation de la connaissance ne couvrait nos besoins. Pour ce faire on a choisi UML comme outil de structuration de la connaissance langage qui a fait ses preuves dans beaucoup de domaines de structuration des connaissances. Ainsi les différents modèles constituant la mémoire de projet ont été formalisés avec le langage UML.

Nous avons poursuivi dans le Chapitre 3 à présenter l'architecture générale de la mémoire de projet et les différents modèles la constituant en utilisant le langage UML. Ces modèles ont été construits sur la base d'un patron de modélisation qui nous a servi à la représentation des structures arborescentes rencontrées fréquemment dans le domaine. Nous l'avons utilisé pour la description des produits, des processus, des documents, et adapté aux organisations. Pour chaque modèle on avait présenté le contexte, le modèle de classes et un ou plusieurs modèles d'objets comme application.

Afin de factoriser au maximum les propriétés, une classe abstraite *Entité* a été introduite comme super classe de tous les éléments de la mémoire de projet. L'ajout de cette super classe a permis aussi de reporter au niveau le plus élevé de généralisation une relation à sémantique simple « estLiéA ». De ce fait, il est alors possible de relier un élément de la mémoire de projet avec

n'importe quel autre indépendamment de leur type. D'autres relations à sémantique plus riche ont été introduites.

Dans le Chapitre 4 nous avons détaillé l'architecture du système d'information à mettre en œuvre. Nous avons présenté l'architecture logique de notre système, le type de plate-forme sur laquelle elle a été déployée et les langages de développement.

Enfin on a décrit notre expérience pratique au sein d'une entreprise d'ingénierie, à savoir la STUDI, qui nous a permis de tester et de proposer des améliorations de notre approche.

Par ailleurs nous proposons en perspective plusieurs axes de développement.

- Le premier axe consiste à développer un métamodèle dédié à la mémoire de projet suite à l'expérience menée au sein de la STUDI. Dans un but de rendre nos modèles à la fois plus généralisés et perfectionnés. L'architecture MDA pourra être un élément au développement de cet axe. C'est-à-dire ceci consiste à développer un profil dédié à la mémoire de projet.
- Le deuxième axe consiste à établir un lien entre notre outil « P2M2 » et les outils de type SGDT. Ainsi notre travail se situe en amont. En effet notre outil permet la formalisation et la structuration des données dans un langage compréhensible par le concepteur qui sont stockées dans une base de données évoluée (SGBROR). Il sera possible d'établir des liens entre notre outil (haut niveau) et les données techniques (bas niveau).
- Le troisième axe consiste à aider le concepteur à une utilisation simplifiée de la chaîne de conception. Pour ce faire, une étude du couplage entre notre outil et d'autres outils de conception (outils CAO, GED, Workflow, etc.) doit être menée.

REFERENCES BIBLIOGRAPHIQUES

[Abbey, 2005] Abbey Michael Oracle 10g-Notions Fondamentales Campus press Juillet 2004.

[Adb-Ali et al, 2005] Jamal Abd-Ali, Karim El Guemhioui, « Transformation horizontale de PSMs: de EJB à .NET », 1ère Journée sur l'Ingénierie dirigée par les modèles Paris 30-1 juillet 2005.

[Abdemouleh, 2004] Abdelmouleh A. « Composants pour la modélisation des processus métier en Productique, basés sur CIMOSA » Thèse de doctorat (LGIPM/AGIP), Université de Metz Septembre 2004.

[AFNOR, 1990] AFNOR, NF X50-150- Analyse de la valeur, caractéristiques fondamentales, 1990.

[AFNOR, 1991] AFNOR (1991) X50 50-105, "Le management de projet" : concepts, Paris, août 1991.

[AIT, 1997] Projet E.J. WAITE, AIT – Advanced Information Technology for Design and Manufacture – WP 4.1 Terminology, 1997.

[Alexander et al, 1977] Alexander C., Ishikawa S., et al., A Pattern Language, Oxford University Press, new York, NY, 1977.

[Anderl, 1995] Anderl, P. « KADS TOOL : un atelier pour la méthode CommonKADS », in Génie logiciel et genie cognitive, N° 31, Juin 1993, pp. 42-50.

[ATL, 2005] Projet Atlas, INRIA, « ATL: Atlas Transformation Language ». <http://www.sciences.univnantes.fr/lina/atl/,2005>.

[Bahrami et al, 1993] Bahrami, A., Dagli, C .H., (1993). *From fuzzy input requirement to crisp design*. International journal of Advanced manufacturing Technology, 8, pp.52-60.

[Balmisse, 2002] Gilles Balmisse « Gestion des connaissances : Outils et application du Knowledge Management », Edition Vuilbert, Septembre 2002.

[Barros et al., 2000] B. Barros, F. Verdejo « DEGREE : A system for carryingout and evaluating collaborative learnig experiments in distance learning ». Artificial Intelligence. Winter 2000. N°9.pp 27-37.

[Basque, 2004] Richard Basque, « CMMI : un itinéraire fléché vers le Capability Maturity Model Integration », Dunod, octobre 2004.

[Bazjanac, 1974] Bazjanac, V. Architectural Design Theory: Models of the Design Process, in Spillers, 1974.

[Beck et al, 1987] Beck K., Cunningham W., Using Pattern Languages for Object-Oriented Programs, Proceedings of ECOOP'94, 1994.

[Bekhti, 2003] Bekhti S. (2003), « DYPKM : Un Processus Dynamique de Définition et de Réutilisation de Mémoire de Projet », Thèse de l'UTT, spécialité Réseaux, Connaissances et organisations, 17 Décembre 2003.

[Benjamins, 2000] V. Benjamins “project Presentation IBROW. An Intelligent Brokering Service for Knowledge Component reuse on the World-Wide- Web”. University of Amsterdam, March 2000. <http://www.swi.psy.uva.nl/projects/ibrow/docs/deliverables/project-presentation.pdf>

[Bensta, 2005a] Hatem Ben Sta, Lamjed Ben Said, Khaled Ghédira, Michel Bigand, Jean Pierre Bourey « Cartographies of ontology concepts », International Conference on Enterprise Information System à Miami du 24 au 28 mai 2005.

[Bensta, 2005b] Hatem Ben Sta, Michel Bigand, Khaled Ghédira, Jean Pierre Bourey « Une approche de développement de mémoire de projet en utilisant un schéma de conception », AFIA2005, Association Française d'Intelligence Artificielle, à Nice, France du 31 Mai au 03 Juin 2005.

[Bentley, 2001] Principe élémentaire de la PRINCE2 méthode structure de gestion de projet Colin Bentley, Hampshire Training Consultants, Avril 2001.

[Bézivin, 1996] Bézivin, J., The OSMOSIS path to method unification, 1996, Rapport, Université de Nantes, 1996.

[Blackler, 1995] Blackler, F. Knowledge, Knowledge Work and Organisations: An Overview and Interpretation. Organisation Studies, Vol. 16 N° 6, pp. 1021-1046, 1995.

[BNAE, 1992] BNAE, NF L00 007 B – Industrie aéronautique et spatiale- Vocabulaire – Termes Généraux, 1992.

[Bohner et al, 1996] Bohner, S.A, Arnold, R.S., Software Change Impact Analysis. IEEE Computer Society Press, 1996.

[Bonnardel, 1992] Bonnardel N., *Le rôle de l'évaluation dans les activités de conception*, Thèse de Doctorat en Psychologie Cognitive, Université d'Aix en Provence, Aix en Provence, 1992.

[Boy et al, 2000] Boy J., Dudek C., Kuschel S. (2000). *Management de projet : fondements, méthodes et techniques*. De Boeck Université, Bruxelles.

[Breuker et al, 1993] Breuker, J., De Greef, P. (1993) *Modelling system-user cooperation in KADS*. KADS: A principled approach to knowledge-based System Development. Ed. Schreiber, G., Wielinga, B, Breuker, J. Academic Press Publ.

[Brice]: BRICE A. - Design Rationale Management (DRAMA),
<http://www.quantisci.co.uk/drama>.

[Brooking, 1999] Brooking A. (1999). Capturing Knowledge within the Organization. In: *Corporate Memory: Strategies for Knowledge Management*, International Thomson Business Press, London, p. 61-75.

[Brown et al, 1985] Brown, D.C., Chandrasekaran, B. "Expert systems for a class of mechanical design activities", In *Knowledge Engineering in Computer-Aided Design* (J . S. Gero, ed.), Elsevier, Amsterdam, 1985, pp. 259-289.

[Brun, 1991] Brun, J. M. Towards integrated product modelling. The contribution of an ESPRIT program, Proc. MICAD, Paris, 12-15 Février 1991, pp. 14-30.

[Bucciarelli, 1988] L. Buciarelli, « *An ethnographic perspective on engineering design* » 1988. Design studies, Vol. 9, n°3, pp 159-168., juillet.

[Buckingham Shum, 1997] BUCKINGHAM SHUM S., "Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge". Proceedings of AAI Spring Symposium on Artificial Intelligence in Knowledge Management, P.9-16 (1997);
<http://ksi.cpsc.ucalgary.ca/AIKM97/AIKM97Proc.html>

[Caldeira, 1993] Caldiera, G., Impact of ISO9000 on Software Maintenance, IEEE, 1993.

[CaldWell et al, 2000] N. CaldWell, J. Clarkson, "Web-Based Knowledge Management for Distributed Desing". IEEE Intelligent Systems, May-June 2000, pp 40-47

[Calvez, 1990] CALVEZ J.P., Spécification et conception des systèmes, une méthodologie, Ed.Masson, Paris, 1990.

[Cantzler, 1997] Olivier CANTZLER, Thèse de Doctorat « Une architecture conceptuelle pour la pérennisation d'historiques globaux de conception de produits industriels complexes », Ecole Centrale de Paris, Avril 1997.

[Carlier, 2005] Management de la qualité pour la maîtrise des systèmes d'information Carlier A. Editions Hermes 2005.

[Carratt et al. 2000] B. Carratt, T. Alaux, and B. Eynard, (2000), "Application d'un système de gestion de données techniques chez un motoriste avion", in Proceedings of Conférence Internationale sur la CFAO, la Simulation et les Nouvelles Technologies de Conception et de Fabrication (MICAD'00), Paris, France, pp. 279-286.

[Castellani, 1993] Castellani, X., (1993). *Méthodologie Générale d'Analyse et de Conception des Systèmes d'Objets Tome 1*; Masson, Paris.

[Chandrasekaran, 1989] Chandrasekaran, B. "A framework for design problem solving", *Research in Engineering Design*, Vol. 1, No. 2, 1989, pp. 75-86.

[Chandrasekaran, 1990] Chandrasekaran, B. "Design Problem Solving: A Task Analysis"; *AI Magazine*, winter 1990.

[Chandrasekaran et al, 1993] Chandrasekaran, B., Goel, A.K., Iwasaki, Y. "Functional representation as design rationale", *IEEE Computer*, January 1993, pp. 48-56.

[Charlet et al, 1999] Jean Charlet (DSI/AP-HP), Nathalie Aussenac-Gilles (IRIT), www.irit.fr/GRACQ/COURS/CoursJCNA/IC-OntoBCT99.pdf

[CMM, 1991] Paulk, M.C, Curtis, B., Chrissis, M.B. et al., *Capability Maturity Model for Software (CMU/SEI-91-TR-24, ADA240603)*. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon, University, August 1991.

[Coad, 1992] Coad P., *Object-Oriented Patterns*, *Communications of the ACM*, Vol 35, N°9, September 1992.

[Coad, 1996] Coad P., *Object Models-Strategies, Patterns and Applications*, Yourdon Press Computing Series, 1996.

[Cobos et al, 2002] R. Cobos, J.A.Esquivel, X. Alaman, "IT Tools for knowledge Management: A study of the current Situation. *European Online Magazine for IT professional*, vol.3, N°1, February 2002. <http://www.upgrade-cepis.org/issues/2002//1/upgrade-vIII-1.html>.

[Cochoy et al. 1998] Cochoy, F. et Terssac (de), G. (1998) Traçabilité et mappabilité : deux enjeux qui travaillent l'organisation. *Actes du Second colloque Langage et Travail*, Centre de Recherche en Gestion, Ecole Polytechnique, Paris, 24-26 septembre.

[Concklin et al. 1991] Concklin, E. J., & Burgess-Yakemovic, K. C. (1991). A Process-Oriented Approach to Design Rationale. *Human-Computer Interaction*, Vol.6, 357-391, 1991.

[Conklin et al., 1998] CONKLIN J.E. et BEGEMAN M.L. – gIBIS: "A Hypertext Tool for exploratory Policy Discussion", *ACM Transactions on Office Informations Systems*, 6,303-331, 1998.

[Cook et al, 1999] Cook S. and Brown J. *Bridging Epistemologies: the Generative Dance between Organisational Knowledge and Organisational Knowing* .*Organisation Science*, Vol. 10 No 4, pp. 381-400, 1999.

[Cooper, 1979] Cooper, R.G., (1979). *The dimensions of industrial new products success or failure* *Journal of Marketing*, vol. 14, p93-103.

[Coplien, 1992] Coplien J.O., *Advanced C++ Programming Styles and Idioms*, Addison-Wesley, 1992.

[Courtois, 1985] Courtois, P.J. "On time and space decomposition of complex structures", Communications of the ACM,-Vol :-2i-No, 6, 1985; pp.-590-603 .

[Cranefield et al., 1999] Cranefield, S. et Purvis, M. (1999), UML as an ontology modelling language. Dans: Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm, Sweden. 84,132. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-23/cranefield-ijcai99-iii.pdf>.

[Daimler, 2000] Daimler Chrysler (2000).methodology and tools Oriented to Knowledge-based engineering Applications, MOKA project ESPRIT 25418, deliverable D4.3, final version, 26 juin.

[Darses, 1997] Darses F., « L'ingénierie concourante: un modèle en meilleure adéquation avec les processus cognitifs de conception », Bossard P., Changevrièr C. and Leclair P. (Eds), *Ingénierie concourante : de la technique au social*, Economica, 1997.

[Dart, 1993] Dart, S., The past, Present, and Future of Configuration Management, Technical Report, Software Engineering Institute, Caarnegie-Mellon University, CMU/SEI-92-TR-9, May 1993.

[Davenport et al., 1998] Davenport T.H., Prusak L. (1998). *Working Knowledge : How Organizations Manage What They Know*. Harvard Business School Press, Boston.

[Davies, 2001] J. Davies, "Supporting Virtual Communities of Practice". Industrial Knowledge Management, Springer-Verlag, London, 2001.

[Delattre, 1985] Delattre, P., (1985). *Système, structure, fonction, évolution. Essai d'analyse épistémologique*. Collection recherches interdisciplinaires. Ed. Maloine, Paris.

[Deneux, 2002] Dominique DENEUX HdR Méthodes et modèles pour la conception concourante, l'Université de Valenciennes et du Hainaut Cambrésis 2002.

[Dickey, 1997] Dickey, R.« The need for speed », Computer Integrated Manufacture & Engineering : above and Beyond PDM, Vol. 1, N°. 2, 1997, pp. 16-20.

[Dieng et al, 2000] : R. DIENG, O. CORBY, A. GIBOIN, J. GOLEBIEWSKA, N. MATTA, M. RIBIERE, "Méthodes et outils pour la gestion des connaissances", Dunod, Paris, 2000.

[Dieng et al, 1998] DIENG R., CORBY O., GIBOIN A. & RIBIÈRE M.: «Methods and Tools for Corporate Knowledge Management», Proceedings of the Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98), Banff, Alberta, Canada, 1998. <http://ksi.cpsc.ucalgary.ca/KAW98S/dieng>.

[Dietinger et al., 1998] T. Dietinger, C. Gütl, H. Maurer, K. Schmaranz, « Intelligent Knowledge Gathering and Management as New Ways of an Improved Learning Process ». Proceedings of AACE WebNet'98 Conference, Orlando, Florida USA. 7-12 Novembre, 1998. pp 244-249. www2.iicm.edu/cguetl/papers/webnet98/webnet98_final.pdf.

[Dinsdale, 1991] Dinsdale J., (1991). *Engineering design education*, CIRP annals manufacturing technology, vol. 40/2 pp 595- 601.

[Djeraba, 1993] Djeraba, C., « Quelques liens sémantiques dans un système à Base de Connaissances » Thèse de doctorat Université Claude Bernard de Lyon, 1993.

[Doumeingts, 1984] Doumeingts, G. (1984), méthode GRAI, méthode de conception des systèmes en productique, Thèse de doctorat d'état, Université Bordeaux 1.

[Drucker, 1993] Peter Drucker. Au-delà du capitalisme, la métaphore de cette fin du siècle. Dunod Paris, 1993.

[Dumont, 2005] ITIL pour un service optimal par Christian Dumont, Novembre 2005, EYROLLES.

[Durkheim, 1982] Emilie Durkheim, The rules of the Sociological Method, S.Lukes Editor; traduit par W.D.Halls, Free Press, 1982.

[Durruvu et al., 1989] Durruvu, S., et al., (1989) Knowledge based systems applications in engineering design : research at MIT. AI Magazine, 10(3), 79-96.

[Eastman, 1981] Eastman, C. M. « Recent developments in representation in the science of design », 18th IEEE Design Automation Conference, 1981, pp. 13-21.

[Ermine et al., 1996] Ermine J.L., Chaillot M., Bigeon P., Charenton B. et Malavielle D., MKSM a method for knowledge management, Knowledge management : Organization, Competence and Methodology, Proc. Of ISMICK'96, Rotterdam, 21-22 1996, Advances in Knowledge Management, vol. 1, Würzburg, Ergon Verlag, pp. 288-302.

[Ermine, 1996] Ermine J.L Les systèmes de connaissances. Hermès, 1996.

[Ermine, 2000] Ermine, J.L, Capitaliser et partager les connaissances avec la méthode MKSM. A parître in Traité IC2 (Information, Communication, Commande), Volume Capitalisation des Connaissances, Hermès, 2000.

[Ermine, 2003] J.-L. Ermine. La gestion des connaissances, Hermes, 2003. Science Publications; Paris (FRA). 166 p. *Localisation : MYCSA*

[Escande, 1996] Escande, E. «Modélisation objet du processus de conception dans le domaine du génie électrique : application au cas de la machine asynchrone», Thèse en génie Electrique de L'institut National Polytechnique de Grenoble, France, Décembre 1996.

[Eschenbächer et al, 1999] Eschenbächer, J., Cocquebert, E., (1999). *Managing extended enterprise manufacturing networks by using the electronic assistant (GENCMM)*. IMS.99, Leuven, Belgium, pp 393-403.

[Euzenat, 1996] Euzenat J.: «Corporate memory through cooperative creation of knowledge bases and hyper-documents», Proceedings of the 10th Knowledge Acquisition for Knowledge-based Systems Workshop (KAW'96), pp. 36/1-36/18, 1996.

[Eynard, 1999] Eynard B., « Modélisation du produit et des activités de conception Contribution à la conduite et à la traçabilité du processus d'ingénierie », Thèse de Doctorat de l'Université Bordeaux1, 1999.

[Fenves et al, 1985] Fenves, S. J., Rasdorf, W. J. « Treatment of Engineering Design Constraints in a Rational Database », *Engineering With Computers*, Vol. 1, N° 1, pp. 27-37, Spring 1985.

[Finkelstein et al, 1995] Finkelstein, A.C., Finkelstein, L., (1995). *Review of Design Methodology* Design and Systems: W.W. Gasparski; Transaction Publishers; New Brunswick (U.S.A) and Lono (U.K).

[Firestone, 2001] Joseph M. Firestone. Keys issues in Knowledge Management. Knowledge And innovation, Journal of the KMCI, Volume 1 N°3, 15 Avril 2001. <http://www.DKMS.com>

[Fowler, 1997] Fowler M., *Analysis Patterns-reusable Object Models*, Addison-Wesley, Reading MA, 1997.

[Gamma et al, 1994] Gamma E., Helm R., Johnson R. and Vlissides J., *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley, Publishing Company, 1994.

[Ganascia, 1990] Jean-Gabriel GANASCIA « L'âme machine », Seuil, 1990.

[Gardarin, 1999] Georges GARDARIN, « Bases de Données : Objet et Relationnel », Eyrolles Janvier 1999.

[Gardarin, 2003] Georges GARDARAIN « Bases de données », Eyrolles 5ème tirage 2003.

[Garel, 2003] Gilles Garel « Le management de projet », Poche Novembre 2003.

[Gauss et al, 1995] Gauss, D.C., Minch, E., (1995) *Design Processes: A State Space Perspective*. Design and Systems: W.W. Gasparski; Transaction Publishers; New Brunswick (U.S.A) and Lono (U.K).

[GDT, 2004] Grand Dictionnaire Terminologique, Office Québécois de la Langue Française, <http://www.granddictionnaire.com>

[Gédalge, 2004] Pierre Gédalge, promotion 2004, Thèse professionnelle – Mastère spécialisé Management des systèmes d’information et des nouvelles technologies MSIT « L’externalisation de l’informatique d’une entreprise éléments pour un outil d’aide à la décision dérivé de la théorie des coûts de transaction », Ecole des Mines de Paris.

[Genest et al., 1990] Genest B.-A., Nguyen T.H. (1990). *Principes et techniques de la gestion de projets*. Les éditions Sigma Delta, Laval.

[Georgel, 2005] IT Gouvernance : Maîtrise d'un système d'information de Frédéric Georgel, Thierry Chamfrault DUNOD Juin 2005.

[Gero, 1987] Gero, J.S. "Prototypes: a new schema for knowledge based design", Tech. Rep., Architectural Computing Unit, Dept. Architectural Sci., 1987.

[Giard, 1991] V. GIARD, "Gestion de projets", Edition ECONOMICA, Paris, 1991.

[Girod, 1995] Girod M. (1995). La mémoire organisationnelle. *Revue française de gestion* , 105, p. 30-42.

[Girod, 1996] GIROD-SEVILLE M. : *La mémoire des organisations*, L'Harmattan, 1996.

[Goldberg et al, 1992] D. Goldberg, B. Oki, D. Nichols and D.B. Terry:” Using Collaborative Filtering to Weave and Information Tapestry”. *Communications of the ACM*, December 1992. Vol. 35, N°12, pp 61-70.

[Griethuysen, 1992] Griethuysen, F., (1992). *The constructive-deductive design approach-application to power transmissions*, CIRP annals manufacturing technology, vol. 41/1 pp 169-172.

[Gruber 1993] Thomas R. Gruber . Toward principles for the design of ontologies used for knowledge sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993.

[Grundstein et al, 1999] Grundstein, M. Barthès, J-P .A.. An Approach to Enterprise Knowledge Capitalization. In *Knowledge Management. Enterprise, Network and Learning. Advances in Knowledge Management*. Germany, 2,1999, 151-159.

[Guéra, 1992] Guéra, F. “Réutilisation de solutions génériques pour résoudre des problèmes de conception”, Thèse en Informatique de l’Institut Blaise Pascal, Paris VI, France, Janvier 1992.

[Gulliver, 1989] Gulliver F.B. (1989). Post-Project Appraisals Pay. In : *Havard Business Press Review*, Managing Projects and Programs, Havard Business School Press, Boston, p. 279-288.

[Gzara, 2000] Lilia GZARA, « Les patterns pour l’ingénierie des systèmes d’information produit », thèse de l’Institut National Polytechnique de Grenoble, 12 Décembre 2000.

[Hachette, 2001] Hachette Multimédia / Hachette Livre.

[Hansen et al, 1999] M. T. Hansen, N. Nohria, T. Tiemey, “What’s Your Strategy for Managing Knowledge?”, in: Harvard Business Review, Vol. 77, N°.3-4, 1999, 106-116.

www.wiwiss.fu-erlin.de/w3/w3schrey/foalien/mdialog/Aufsatz_Hansen_Nohria_Tierney.pdf.

[Harani, 1997] : HARANI Y., "Une Approche Multi-modèles pour la Capitalisation des Connaissances dans le Domaine de la Conception", Thèse de l'INPG, spécialité en Génie Industriel, 19 Novembre 1997.

[Hermosillo et al., 2002] Hermosillo Worley J., Grabot B; Geneste L., Aguirre O. (2002) “Role, skill and Knowledge introducing human resources” in BPR, 9th International Multi-Conference on Advanced Computer Systems - Conference on Production System Design, Supply Chain management and Logistics, Miedzydroje, Pologne, 23-25 octobre.

[Herniaux, 1993] Herniaux G. (1993). *Organiser la conduite de projet*. INSEP Éditions, Paris.

[Heurtel, 2005] Olivier Heurtel, Oracle 10g Administration, Collection Ressources Informatiques, Edition ENI Mars 2005.

[Hu, 2000] Hu, X., (2000), a survey on design rationale: representation, capture and retrieval, Proceedings of DETC’00, 2000 ASME Design Engineering Technical Conferences, September 10-3, 2000, Baltimore, Maryland, DETC2000/DFM-14008.

[Iffenecker, 1992] Iffenecker, C. “Un système multi-agents pour le support des activités de conception de produits », Thèse en Informatique de l’Institut Blaise Pascal, LAFORIA, Université Paris VI, France, 1992.

[IPDMUG, 1997] International Product Data Management Users Group, IPDMUG Glossary, <http://www.pdmic.com/IPDMUG/members/glossary1.html>

[ISO, 1987] ISO, Quality Systems – Model for Quality Assurance in Design/development, Production, Installation and Servicing, 1987.

[ISO, 1991] ISO, ISO 9000-3, Quality Management and Quality Assurance Standards Guidelines on the application of ISO90001 to the development, supply and maintenance of software, ISO, 1991.

[Jonassen et al, 1992] D. Jonassen, T. Mayer and R. A. McAleese, “Manifiesto for a Constructivist Approach to Uses of Technology in Higher Education”. Designing Environments for Constructive Learning, Duffy, Lowyck & Jonassen, Springer-Verlag. pp 231-247.

[Jones et al, 1963] JONES J.C., THORNLEY D. Conference on Design Methods, Pergamon Press, Oxford, 1963.

[Kadima, 2005] Hubert Kadima MDA, Conception orientée objet guidée par les modèles, DUNOD 2005.

[Karl, 1972] Karl R. Popper. Objective Knowledge. London, England: Oxford University Press, 1972.

[Karsenty, 1994] : Karsenty L. "L'explication d'une solution dans des dialogues de conception". Thèse de doctorat d'Ergonomie Cognitive de l'Université Paris VIII, Juillet 1994.

[Karsenty, 1996] L. Karsenty 1996 "An Empirical Evaluation of Design Rationale Documents", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96), Vancouver, 14-18 Avril 1996, ACM Press.

[Karsenty, 2001] Karsenty L. (2001) Méthodes pour la création de mémoires de projet en conception. Revue Française de Gestion Industrielle, 20(1), 35-51.

[Kerzner, 1998] Kerzner H. (1998). *Project Management: A System Approach to Planning Scheduling and Controlling*. John Wiley and Sons, New York.

[Klein, 1993] Klein, M. "Capturing Design Rationale in Concurrent Engineering Teams", IEEE Computer Support for Concurrent Engineering, January 1993, pp. 39-47.

[Kleiner et al, 1999] Kleiner A., Roth G. (1999). L'expérience comme guide de l'entreprise. In: *Le Knowledge Management*, Harvard Business Review, Editions d'organisation, p. 177-196.

[Kogut et al., 2002] Kogut, P.A., Cranefield, S., Hart, L., Dutra, M., Backlawski, K., Kokar, M. et Smith, J. (2002), UML for ontology development. The knowledge Engineering Review, Vol. 17, N°1, p. 61-64. 84,132.

[Kühn et al, 1997] Kühn O. & ABECKER A.: «Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. Journal of Universal Computer Science, 3(8), pp. 929-954, 1997.

[Kusiak et al, 1993] Kusiak, A., Wang, J. "Decomposition of the Design Process", Journal of Mechanical Design, Vol. 115, pp. 687-695, Dec. 1993.

[Labrousse, 2004] Labrousse M., (2004), « proposition d'un Modèle Conceptuel unifié pour la Gestion Dynamique des Connaissances d'Entreprise », Thèse de l'Ecole Centrale de Nantes, spécialité Génie Mécanique, 13 Juillet 2004.

[Lamonde et al, 2001] La mémoire de projet : véhicule d'intégration de l'ergonomie et de la SST à la Conception. Fernande Lamonde, Anabelle Viau-Guay, Philippe Beaufort, Jean-Guy Richard, PISTES © Vol. 3 No. 2 Octobre 2001.

[Lanzara, 1986] Lanzara, G. F. (1986). La théorie de la conception entre "problem solving" et "problem setting". In A. Demailly, J. L. Lemoigne (eds.). *Sciences de l'intelligence et sciences de l'artificiel*. Lyon: P.U.L.

[Larousse, 2000] Le dictionnaire français Le Petit Larousse, 2000.

[Latombe, 1977] Latombe, J.C. "Une application de l'intelligence artificielle à la conception assistée par ordinateur: le système TROPIC », Thèse de Doctorat d'Etat Es-Sciences en mathématiques, Grenoble, France, 1977.

[Le Bissonnais, 1992] Le Bissonnais J. (1992). *Le management de projet de A à Z*. AFNOR, Paris.

[Le Moigne, 1994] Le Moigne, J.L., (1994). *La théorie du système général. Théorie de la modélisation*. 4ème Ed. PUF.

[Leinen, 1997] Leinen, S., (1997). *Une nouvelle approche pour la modélisation et la gestion des contraintes en CAO*. Thèse de l'Université de Metz (LRIM), 19 décembre.

[Lentzner, 2004] Rémy Lentzler SQL 3 Avec Oracle, MySQL, Microsoft SQL Server et Access, DUNOD 2005.

[Lewkowicz et al, 1999] LEWKOWICZ M., ZACKLAD M. "MEMO-net, un collecticiel utilisant la méthode de résolution de problème DIPA pour la capitalisation et la gestion des connaissances dans les projets de conception", IC'99, Palaiseau, p.119-128. 14-16 juin 1999.

[Longueville et al, 2003] Longueville B., Le Cardinal J.S, Bocquet. J.C, Daneau. P. Toward a project memory for Innovative Product Design, A Decision-Making Process Model. International Conference on Engineering Design, ICED 03 Stockholm, August 1ç-21, 2003

[MacLean et al., 1991] MACLEAN A., Young R.M., BELLOTTI V.M.E., Moran T.P., "Questions, Options, and Criteria: Elements of Design Space Analysis", *Human-Computer Interaction*, Vol.6, 1991.

[Maders, 2000] Maders H.P. (2000). *Conduire un projet d'organisation : guide méthodologique*. Éditions d'Organisation, Paris.

[Maier, 2002] R. Maier, "State-of-Practice of Knowledge management Systems: Results of an Empirical Study". European Online Magazine for IT professional, Vol. 3, N°1, February 2002.

[Malhotra et al, 1980] Malhotra, A., Thomas, J., Carroll, J., & Miller, L. (1980) Cognitive processes in design. *International Journal of Man-machine Studies*, 12, 119-140.

[Malhotra, 2000a] Yogesh Malhotra, Knowledge Management for E-business Performance: "Advancing Information Strategy to Internet Time, Information Strategy, the executive's journal, vol.16 .(4), Summer 2000.

[Malhotra, 2000b] Yogesh Malhotra, From information Management to Knowledge Management; Beyond 1 the 'Hi-Tech Hide bound's Systems, *Knowledge Management Journals*, 2000.

[Malvache et al., 1993]: MALVACHE, P. and PRIEUR, P., "Mastering corporate experience with the Rex method". In J. P. Barthès ed., Proc. of ISMICK'93, Compiègne, October 1993, pp. 33-41.

[Marino et al., 1990] O. Marino, F. Rechenmann, P. Uvietta, *Multiple perspectives and classification mechanism in Object-oriented Representation*, Proc. 9th ECAI, Stockholm, Sweden, p. 425-430, Pitman Publishing, London, August 1990.

[Matta et al, 1999a] Matta N. et al. (1999a), Ribière R., Corby O. Définition d'un modèle de mémoire de projet, Rapport de Recherche INRIA N. 3720, Juin 1999.

[Matta et al, 1999b] Matta N. et al. (1999b). Méthodes de capitalisation de mémoire de projet, Rapport de Recherche INRIA N. 3819, Novembre 1999.

[Matta et al., 2000] N. MATTA, M. RIBIERE, O. CORBY, M. LEWKOWICZ, M. ZACKLAD, "Project Memory in Design", *Industrial Knowledge Management - A Micro Level Approach*, Rajkumar Roy (Eds), Springer-Verlag, 2000

[Maurino, 1993] Maurino M, *La gestion des données techniques – Technologie du concurrent engineering*, ED. Masson, Paris, 1993.

[Maier, 2002] R. Maier, "State of Practice of Knowledge Management Systems : Results of an Empirical Study". *European Online Magazine for IT professional*, vol. 3 N° 1, Février 2002.

[Mayer et al, 1995] Mayer, R. J. Menzel, C. P., Painter, M.K., deWitte, P.S., Blinn, T., Perakath, B., *Information Integration for Concurrent Engineering (IICE) IDEF3 process description capture method report*, <http://www.idef.com>, September 1995.

[McCullough et al., 1998] McCullough D., Korelsky T. et White M. - *Information Management for Release-based Software Evolution Using EMMA, Software Engineering and Knowledge Engineering*, 1998.

[McDermid, 1991] McDermid John. (1991). *Software Engineer's Reference Book*. Oxford: Butterworth-Heinemann. ca. 1200 p.

[McElroy, 2002] Mark. W. McElroy. *The new Knowledge Management: Complexity, Learning and Sustainable Innovation*. Knowledge Management Consortium International Press. November, 2002.

[MacLean et al., 1991] MACLEAN A., Young R.M., BELLOTTI V.M.E., Moran T.P., "Questions, Options, and Criteria: Elements of Design Space Analysis", *Human-Computer Interaction*, Vol.6, 1991.

[Mekhilef et al, 2003] Mekhilef M., Kelleher D, Olesen A, "European Guide to Good Practice in Knowledge Management – Chapter 1 – Terminology", site du Knowledge Board, 2003.

[Meredith et al., 1989] Meredith, Jack R., Mantel, Samuel J. Jr. (1989). *Project Management : A Managerial Approach*. John Wiley and Sons, New York.

[Midler, 2002] Midler C., « L'auto qui n'existait pas—Management des projets et transformation de l'entreprise », ISBN 2 10 004228 9, 2002.

[Mossé, 2002] Francis G. Mossé Modeling Roles A Practical Series of Analysis Patterns JOURNAL OF OBJECT TECHNOLOGY, ©JOT, 2002 Vol. 1, no. 4, September-October 2002

[Muller, 1997] Muller P.A Modélisation objet avec UML. Eyrolles, Avril 1997.

[Muller et al, 2000] Muller, P.-A. et Gaertner, N. (2000), Modélisation objet avec UML. Collection Best of Eyrolles, Paris, France, 2^e éd. 84,233.

[Nadler, 1995] Nadler, G., (1995). *Systems methodology and design*. Design and systems: W.W. Gasparski; Transaction Publishers; New Brunswick (U.S.A) and Lono (U.K).

[Nahapiet et al, 1998] Nahapiet, J and Ghoshal, S. Social Capital, Intellectual Capital and the Organizational J Advantage. Academy of Management Review, Vol. 23 No 2, pp. 242-266, 1998.

[Nonaka et al, 1995] Nonaka, I., Takeuchi, H., The knowledge creating company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, p284, 1995.

[Objecteering, 2005] <http://www.objecteering.com/>

[OMG, 2003] OMG, (2003) Object Management Group Unified modelling language Specification, www.omg.org/docs/formal/03-03--01.pdf, Version 1.5 formal/03-03-012003, March 2003.

[OMG, 2003a] OMG: MOF 2.0, Query / Views / Transformation. ad/2002-04-10, Revised Submission, Version 1.0, 2003/08/18, OpenQVT, disponible à <http://www.omg.org/docs/ad/03-08-05.pdf>

[OMG, 2003b] OMG: OCL Response to the UML 2.0 OCL RfP (ad/2000-09-03). Revised Submission, Version 1.6. January 6, 2003. OMG Document ad/2003-01-07, disponible à <http://www.omg.org/docs/ad/03-01-07.pdf>

[OMG, 2005] OMG: UML Unified Modelling Language Specification version 1.3, <http://www.omg.org/uml>, 2005.

[Ouazzani et al, 1997] Ouazzani, A., Bernard, A., Bocquet, J.C., (1997). *Design Process Management System : SAGEP*, International Conference on Engineering Design, ICED 97, Volume 1, pp 221-226, Tampere, August 19-21.

[Pachulski et al, 2000] Alexandre Pachulski, Michel Grundstein, Camille Rosenthal-Sabroux. GAMETH: A 1 Methodology aimed to locate the companies crucial Knowledge.ECKM 2000, Bled(Slovenie),26-21, Octobre 2000.

[Pahl et al, 1984] Pahl, G., Beitz, W., (1984). *Engineering design*, Ed. Ken Wallace, Publ. The design council, London.

[PDMIC, 1998] Product Data Management Information Center, Glossary of Product Data Management Related Terms, (<http://www.pdmic.com/glossary/index.html>).

[Penalva, 1990] Penalva J.M. - SAGACE: une représentation des connaissances pour la supervision de procédés, Systèmes Experts de deuxième génération, EC2 (Ed.), Avignon, 1990.

[Penalva, 1994] Penalva J.M. - SAGACE, la modélisation des systèmes dont la maîtrise est complexe, ILCE, EC2 (Ed), Montpellier, 1994.

[Perrin, 1999] Perrin, J. (1999). *Diversité des représentations du processus de conception, diversité des modes de pilotage de ces processus*, ouvrage collectif ECOSIP : Pilotage et évaluation des processus de conception, éd. L'Harmattan, Paris.

[PMI, 1998] Project Management Institute (1998). *Management de projet : un référentiel de connaissances*. AFNOR, Paris.

[Polanyi, 1996] Michael Polanyi, The Tacit Dimension. Routledge and Jegan Paul, 1996, London, UK.

[Pomian, 1996] Pomian J. (1996). *Mémoire d'entreprise : techniques et outils de la gestion du savoir*. Éd. Sapiientia.

[Prat et al, 2003] Prat C.,Mira-Bonnardel S., « Influence de la veille stratégique sur le processus de conception », CONFERE, UTBM, 2003.

[Prax, 1997] Prax, Jean-Yves PRAX (1997) Manager la connaissance dans l'entreprise, les nouvelles technologies au service de l'ingénierie de la connaissance. INSEP éditions.

[Prax, 2000] Jean-Yves PRAX. *Le guide du Knowledge Management- concepts et pratiques du management de la connaissance*, DUNOD, 2000.

[Prax, 2002] Jean-Yves PRAX. Le management territorial à l'ère des réseaux – Editions d'Organisation Octobre 2002

[Pressman, 1992] Pressman, R.S, Software Engineering, A practitioner's approach , Third Edition, McGraw Hill, 1992, ISBN 0-07-050814-3, pp. 663-733.

[Prusak, 1999] Larry Prusak. Where did Knowledge Management come from Knowledge Directions, Institute of Knowledge Management, Fall 1999.

[Rivière, 2002] M. Rivière, R. Dieng-Kuntz. A Viewpoint Model for Cooperative Building of an Ontology, Conceptual Structures : Integration and Interfaces, Proceedings of the 10th International Conference in Conceptual Structures (ICCS'2002), Springer-Verlag, LNCS 2393, editeur : U. Priss, D. Corbett, G. Angelova. p. 220-234, Borovetz, Bulgarie, 15-19 juillet, 2002.

[Ramon, 2001] Ramon C. Barquin, what is knowledge management? Journal of the KMCI, Volume one, no.two,january 15,2001.

[Randoing, 1995] Randoing J-M, Les SGDT, Ed. Hermès, Paris, 1995.

[Reuchlin, 1981] REUCHLIN M., Psychologie, PUF, Paris, 1981.

[Rittel, 1972] Rittel, H. "On the planning crisis: Systems analysis for the first and second generations", Brediftsokonomien, N°8, 1972.

[Robert, 1993] Le nouveau petit Robert 1, Dictionnaire de la langue française, Ed. Dictionnaire le Robert, Paris, 1993.

[Robert, 2000] Le dictionnaire français Le Petit Robert, 2000.

[Ryle, 1949] Ryle, G.The Concept of Mind. Chicago, IL: University of Chicago Press, 1949.

[Ryle, 1984] G.Ryle, «The concept of Mind», University of Chicago Press1984.

[Saloff, 2000] Michel Saloff-Coste. Le petit livre blanc du management des connaissances, 2000.

[Sauvagnac et al., 2000] Sauvagnac, C., Falzon, P., & Leblond, R. (2000). La mémoire organisationnelle: reconstruction du passé, construction du futur. In J. Charlet, M. Zacklad, G. Kassel, & D. Bourigault (Eds.), *Ingénierie des connaissances. Evolutions récentes et nouveaux défis*. Paris: Eyrolles.

[Scardamalia et al, 1999] M. Scardamalia et C. Bereiter "Student Communities for the Advancement of Knowledge". Communications of the ACM. Vol 39, N°4. April 1996. pp 36-37.

[Schafer et al, 2002] J.B Schafer, J. Konstan and J. Riedl: Electronic Commerce recommender Application. Journal of Data Mining and Knowledge Discovery. Vol.5, N°.1/2, pp 115-152.

[Schreiber et al, 1999] Schreiber G., Akkermans H., AnjewierdenA., De Hoog R., Shadbolt N. Van der Velde W., Wielinga B., "Knowledge Engineering and Management: The CommonKADS Methodology", The MIT Press, ISBN 0- 262-19300-0, 1999.

[Sénéchal et al., 1998] Sénéchal, O., Tahon, C., (1998). *A methodology for integrating economic criteria in design and production management decisions*. International Journal of Production Economics 56-57, Elsevier, pp 557-574.

[Serré, 2000] Serré, Ph., Rivière A., (2000). *Analyse d'une spécification géométrique*. 7ème Colloque AIP-Primeca sur la conception mécanique intégrée. La Plagne, 2-4 avril.

[Simon, 1969] Simon, H.A. « The Sciences of the artificial », MIT Press, 1969.

[Simon, 1973] Simon, H. A. (1973). The structure of ill structured problems. *Artificial Intelligence*, 4, 181-201.

[Simon, 1994] Simon, A., The Integrated CASE Manifesto. American Programmer, July 1994.

[Simon, 1995] Simon, H.A., (1995). *Problem Forming , Problem Finding, and Problem Solving Design End Systems*: W.W. Gasparski. Transaction Publishers; New Brunswick (U.S.A) and Lono (U.K).

[Simon, 1996] Simon G.: « Knowledge Acquisition and modeling for corporate memory: lessons learnt from experience », Proceedings of the 10th Knowledge Acquisition for Knowledge-based Systems Workshop (KAW'96), pp. 41/1-41/18, 1996.

[Skyrme, 1999] David Skyrme Associates. Knowledge Management: Making sense of an oxymoron. Management Insight No. 2, 1999.

[Snowden, 2000] Dave Snowden. Conference Proceedings. 2nd annual Knowledge Management and Organisational Learning Conference. London: Linkage International,2000.

[Sobiech, 1995] Sobiech, R., (1995). *Design as an Object of Sociological Reflection and Study Design End Systems*: W.W.Gasparski. Transaction Publishers; New Brunswick (U.S.A) and Lono (U.K).

[Sohlenius, 1992] Sohlenius, G., (1992). *Concurrent engineering*, CIRP Annals manufacturing technology, Vol. 41/2 pp 645-655.

[Spender, 1998] Spender, J. Pluralist Epistemology and The Knowledge-based Theory of The Firm. ! Edition d'Organisation, Vol. 5 No 2, pp. 233-256, 1998.

[Sriram, 1989] Sriram, Duvvuru, George Stephanopoulos, Robert Logcher, David Gossard, Nicolas Groleau, David Serrano and Dundee Navinchandra, "Knowledge-Based System Applications in Engineering Design: Research at MIT", *AI Magazine*, Vol. 10, No. 3, Fall 1989, pp. 79-96.

[Steels, 1993] STEELS L. - Corporate Knowledge Management, Management of Industrial and Corporate Memory, Proceedings of ISMICK'93, Compiègne 1993.

[Stephanopoulos, 1989] Stephanopoulos, G. (1989). *Artificial intelligence and symbolic computing in process engineering and design*. Proceedings of the 3rd International Conference on foundation of computer aided design. Snowmass Village, Colorado, USA. pp 21-47. 9-14 Juillet.

[**Studer, 1999**] R. Studer: Informations und Wissensmanagement. Lecture at the Institute AIFB, University of Karlsruhe, 1999.

[**Suh, 1990**] Suh, N.P. "The Principles of Design", Oxford University Press, New York, 1990.

[**Sveiby, 2000**] Karl-Erik Sveiby, What is Knowledge Management, March 1996.
Updated 1 March 1998, 3 June 1998 20, March 1999, April 2000.

[**Swan et al, 1999**] Swan, J., Scarbrough, H., and Preston, J. Knowledge Management The Next Fad to Forget people? Proceedings of the 7th European Conference on Information Systems (ECIS'99), Copenhagen, Denmark. pages 668 -678, 1999.

[**Tiwana, 2001**] Tiwana A., "The essential Guide to Knowledge Management", Englewood Cliffs, NJ: Prentice-Hall, 2001.

[**Tollenaere, 1996**] TOLLENAERE M., " Du modèle produit au modèle de conception : une démarche d'intégration de connaissances ", GI'5 : 5e Congrès International de Génie Industriel, Tome II, pp.277-286, Grenoble - France, 2-4 Avril 1996.

[**Tourtier, 1995**] Tourtier P.-A. (1995) : «Analyse préliminaire des métiers et de leurs interactions», Rapport intermédiaire du projet GENIE, INRIA-Dassault-Aviation.

[**Tsoukas, 2002**] Tsoukas, H. Do We Really Understand Tacit Knowledge? Presented to Knowledge Economy and Society Seminar, LSE Department of Information Systems, 14th June, 2002.

[**van Aalst et al., 1995**] J.W. van Aalst, T.T Carey & D.L. McKerlie (1995) "Design Space Analysis as "Training Wheels" in a framework for learning user interface design", In Proceedings of CHI'95, 1995, pp.154-161, Denver, Colorado.

[**Van Heijst et al, 1996**] Van Heijst G., VAN DER SPEK R. & KUIZINGA E.: «Organizing Corporate memories», Proceedings of the 10th Knowledge Acquisition for Knowledge-based Systems Workshop (KAW'96), pp. 42/1-42/17, Banff, Canada, novembre 1996.
<http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html>

[**Vargas, 1996**] Vargas, C., Yvars, P.A., (1996). *Un langage pour la modélisation de processus de conception en ingénierie des systèmes mécaniques*, First International Conference : IDMMME.96, Nantes, France, 14-17 Avril.

[**Vogel 1988**] Claude VOGEL. *Génie Cognitif*, Masson, 1988.

[**Weiser et al., 1998**] Weiser M., Morrison J. (1998). Project Memory: Information Management for Project Teams. *Journal of Management Information Systems*, vol. 14, no 4, p. 149-166.

[Wenger, 1999] Wenger, E. Communities of Practice: The Key to Knowledge Strategy. Knowledge directions, the Journal of the Institute For Knowledge Management, 1999.

[Wiig, 1999] Karl M.Wiig. Knowledge Management: An Emerging Discipline Rooted in a Long History Knowledge Research Institute Inc. fall 1999.

[Wilson, 1980] Wilson, D., (1980). *An Exploratory Study of Complexity in Axiomatic Design*; PhD Department of Mechanical Engineering, MIT, Cambridge.

[Xuereb, 1991] Xuereb, J., (1991). *Une redéfinition du processus d.innovation*, Revue française de gestion, juin.

[Yoshikawa, 1989] Yoshikawa, H., (1989). Design philosophy : the state of the art , CIRP annals manufacturing technology, Vol. 38/2, pp. 579-586.

[Zamfiroiu, 1998] Michel Madalin ZAMFIROIU Thèse intitulé «Contribution a la Traçabilité du processus de conception en Génie Logiciel» Décembre 1998, Université Paris IX-Dauphine UFR Sciences des Organisations.

Résumé

L'objectif principal de nos travaux est de spécifier et de modéliser un outil de gestion des connaissances pour les projets. Il s'agit de proposer un ensemble de modèles dédiés à la réalisation d'une mémoire de projet. De telles mémoires de projet devraient répondre au moins à deux besoins identifiés des concepteurs en situation de réutilisation : accéder aux résultats des projets passés et à la logique de conception qui a conduit à la conception de produit. Il s'agit de remonter dans l'histoire afin de tracer et comprendre les différents choix et arguments acceptés et les solutions rejetées par les concepteurs lors des étapes de conception. Cette traçabilité devient par ailleurs une exigence des démarches qualité, permettant par exemple d'établir les responsabilités en cas de défaillance du produit en phase d'exploitation. Le standard UML a été utilisé comme langage de modélisation. Dans l'ensemble des modèles, nous avons proposé de prendre en compte à la fois le contexte et la logique de conception. Afin de valider nos modèles, un prototype « P2M2 » a été développé permettant l'archivage et la réutilisation des éléments de la mémoire de projet. Cet outil offre des fonctionnalités permettant l'accès aux données ceci selon le besoin des différents utilisateurs. Nos modèles développés ont été adaptés à une entreprise d'ingénierie en Tunisie. Ceci nous a permis de tester nos modèles et d'apprécier la facilité d'adaptation. Un ensemble de perspectives a été tracé : la définition d'un profil dédié à la mémoire de projet, la finalisation du développement de l'application « P2M2 », l'adaptation de l'application sur d'autres cas d'entreprise.

Mots-clés : Connaissance, mémoire de projet, réutilisation, traçabilité, logique de conception, modèles, UML.

Abstract

The main goal of our tasks is to specify and model a tool of knowledge management for projects. We suggest a set of models dedicated to realize a project memory. Such projects memories should respond at least two needs identified by the designer in reuse situation: to have access to previous results and to design rationale this has led to the creation of the product. To do so, one has to go back in history in order to understand various choices and arguments as well as the rejected solutions while creating the product. This traceability has become a condition of high quality allowing for instance to set up responsibility in case of failure of the product in working phase. The standard UML has been used as a modelling language. For all the models, we have taken into consideration not only the context but also the design rationale. In order to valid our models, a prototype "P2M2" has been developed allowing archiving and the reutilization elements of project memory. This tool offers functionality which permits access to the data according to the needs of various users. Our models have been adapted for an engineering enterprise in Tunisia. Thus, we have been allowed to test our models and appreciate the facility to adapt them. A whole of perspective has been traced: defining a profile dedicated project memory, the finalisation of the application development "P2M2", the adaptation of the application in other case of enterprises.

Key Words: Knowledge, project memory, reutilisation, traceability, design rationale, models, UML.