

THÈSE

Présentée
en vue de l'obtention du grade de
docteur de

L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

École doctorale : Systèmes
Spécialité : Systèmes Industriels

par

Franck DARRAS

Proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré

Soutenue le 28 octobre 2004, devant le jury composé de :

MM.	Jean-Pierre CAMPAGNE Michel SCHNEIDER	Rapporteur & Président Rapporteur
MM.	Christian BRAESCH Jean-Marie VIGROUX Hervé PINGAUD Paul GABORIT	Examineur Examineur Examineur (Directeur de thèse) Examineur (Encadrant de thèse)
MM.	Lionel DUPONT Frédéric BENABEN	Invité Invité

Thèse préparée au Centre de Génie Industriel de l'École des Mines d'Albi-Carmaux

Table des matières

Remerciements	xi
Introduction	1
I Étude du Système d'Information des PME/PMI	5
1 Système d'Information	7
1.1 Fruit de l'approche systémique	7
1.2 Définitions	8
1.3 Les rapports difficiles du SI et de l'organisation	11
1.4 Le Système d'Information et le génie industriel	12
2 Progiciels de gestion industrielle	17
2.1 Évolutions des modes de gestion et des outils informatiques	17
2.1.1 Modes de gestion	17
2.1.1.1 MRP	17
2.1.1.2 MRP à boucle fermée	18
2.1.1.3 MRP II	18
2.1.1.4 Vers une gestion poussée des ressources : l'ERP	18
2.1.1.5 SCM	19
2.1.1.6 Évolution des méthodes	20
2.1.2 Couverture logicielle	20
2.1.2.1 De la GPAO à l'ERP	21
2.1.2.2 Du CIM à l'entreprise intégrée	21
2.2 L'ERP	23

2.2.1	Définitions	23
2.2.2	Le système nerveux de l'entreprise	25
2.2.3	Le marché des ERP	26
2.2.3.1	Intégrateurs et Éditeurs d'ERP	26
2.2.3.2	Situation du marché français des ERP	27
2.2.3.3	Situation par secteurs d'activité clients	29
2.2.3.4	Situation par taille d'entreprise cliente	29
2.2.3.5	Nouvelle tendance du marché avec l'Open-Source	30
3	Mutation du paysage industriel des PME/PMI	33
3.1	Les PME/PMI en France	33
3.1.1	Situation économique générale des PME/PMI	33
3.1.1.1	Situation 2001	33
3.1.1.2	Situation 2002	33
3.1.1.3	Prévisions 2003 / 2004	34
3.1.1.4	Taux d'investissement informatique	34
3.1.2	Niveau d'informatisation des PME/PMI	34
3.1.3	Tendance du marché à long terme	35
3.2	De nouvelles formes de partenariat entre entreprises	36
3.2.1	Évolution organisationnelle interne	36
3.2.2	Évolution organisationnelle externe	37
3.2.3	Synthèse	38
3.3	Une pression forte sur la maîtrise des coûts	39
3.4	La relation d'une PME/PMI à son Système d'Information	39
3.5	Quels ERP pour ce type d'entreprise?	41
4	Gestion des projets ERP	45
4.1	Définition du projet ERP	45
4.2	Management des projets	45
4.2.1	La décomposition d'un projet ERP en tâches	46
4.2.2	L'organisation des équipes du projet	48

4.3	Les méthodes utilisées pendant la sélection du progiciel	49
4.4	Les méthodes utilisées pendant le déploiement	49
4.4.1	Méthodes	49
4.4.1.1	Conception détaillée	49
4.4.1.2	Paramétrage	50
4.4.2	Outils des éditeurs ERP	52
4.5	Analyse critique	52
4.5.1	Sélection de l'ERP	52
4.5.2	Déploiement de l'ERP	53
II	Conception d'un ERP	55
5	Architecture d'ERP	57
5.1	Vers une vision large de l'« Architecture »	57
5.2	Quelques caractéristiques architecturales d'un ERP	60
5.2.1	Point de vue logique de l'architecture	60
5.2.2	Point de vue physique de l'architecture	62
5.2.2.1	Exigences de performance et d'exploitation	62
5.2.2.2	Choix architecturaux	62
5.2.3	Représentations architecturales	63
5.2.3.1	Point de vue logique de l'architecture	63
5.2.3.2	Point de vue physique de l'architecture	65
5.3	Nouvelles tendances en architecture logique	67
5.3.1	EAI	68
5.3.2	Urbanisation des Systèmes d'Information	68
5.3.3	Services WEB	69
5.4	Nouvelles tendances en architecture physique	69
5.4.1	Un exemple d'architecture n-tiers avec la technologie J2EE	70
5.5	Synthèse	71

6	Ingénierie des Systèmes d'Information	73
6.1	La gestion des projets en génie logiciel	73
6.1.1	Les critères de qualité du logiciel	73
6.2	Méthodes de conception	75
6.2.1	Cycle de base des projets	75
6.2.1.1	Cycle en V	76
6.2.1.2	Cycle en spirale	76
6.2.1.3	Cycle émergeant	77
6.2.2	La modélisation	79
6.2.2.1	Les cadres de référence pour la modélisation de Système d'In- formation	79
6.2.2.2	SADT	80
6.2.2.3	Les travaux de Zachman	81
6.2.2.4	MERISE	82
6.2.2.4.1	Origine	82
6.2.2.4.2	Trois composantes de la méthode	83
6.2.2.4.3	Transition vers l'objet	84
6.3	Synthèse	86
7	Modélisation d'entreprise	89
7.1	Introduction	89
7.2	Processus de l'entreprise	90
7.3	Normes de la modélisation d'entreprise	91
7.4	Cadre conceptuel	92
7.4.1	Dimension d'abstraction	93
7.4.2	Dimension de généricité	93
7.4.2.1	Constructs	93
7.4.3	Points de vue	94
7.5	Présentation des méthodes	95
7.5.1	La suite IDEF	96
7.5.1.1	IDEF0	96

7.5.1.2	IDEF1	96
7.5.1.3	IDEF2	97
7.5.1.4	IDEF3	97
7.5.2	La méthode GRAI	97
7.5.3	OLYMPIOS	98
7.5.4	MECI	99
7.5.5	Synthèse	99
7.6	Les travaux sur UEML	100
7.7	Synthèse et conclusions	101
III	Utilisation de la modélisation	105
8	Utilisation d'UML pour la modélisation d'entreprise	107
8.1	Proposition d'un méta-modèle	107
8.1.1	Les apports du méta-modèle de CIM-OSA	107
8.1.2	Les apports du méta-modèle de UEML	109
8.1.3	Notre proposition	109
8.2	Mise en application du méta-modèle sur un cas d'école	111
8.2.1	Mise en œuvre du langage UML sur ce cas d'étude	112
8.2.1.1	Point de vue fonctionnel	112
8.2.1.2	Point de vue informationnel	117
8.2.1.3	Point de vue des ressources	119
8.2.1.4	Point de vue organisationnel	122
8.2.2	Synthèse	124
8.3	La grille GRAI comme élément fédérateur	124
8.3.1	La grille des cas d'utilisation	126
8.3.2	La grille des activités	127
8.3.3	Bilan comparatif	127
8.3.4	Enchaînement des diagrammes UML pour la modélisation d'un système	128
8.4	Conclusions	132

9	Cadre de référence	135
9.1	Mise en adéquation des modèles	135
9.1.1	Analyse et spécification	136
9.1.1.1	Côté client	136
9.1.1.2	Côté éditeur	137
9.1.2	Développement	137
9.1.2.1	Côté client	137
9.1.2.2	Côté éditeur	138
9.1.3	Exploitation et maintenance	138
9.1.3.1	Côté client	138
9.1.3.2	Côté éditeur	138
9.1.4	Synthèse	138
9.2	Adéquation dans le cycle de vie	139
9.2.1	Décomposition d'un SI et utilisation d'UML	139
9.2.2	Synthèse	141
9.3	Démarche projet	141
9.3.1	Présentation de l'existant	142
9.3.1.1	Démarche d'analyse des Systèmes d'Information	142
9.3.1.2	Démarche développée lors des travaux de thèse de Boutin	143
9.3.1.3	Projection entre deux modèles	144
9.3.1.4	Synthèse	145
9.3.2	Cadre de référence	145
9.3.2.1	La base du cadre	145
9.3.2.2	Le cadre	146
9.3.2.3	Nouvelle proposition de cadre de référence	147
9.4	Exemple	148
9.4.1	Démarche suivant la taille de l'entreprise	148
9.4.1.1	Démarche pour une petite structure	148
9.4.1.2	Démarche pour une structure moyenne	149
9.4.1.3	Démarche pour une grande structure	150

9.4.2	Gestion de projet dans le déploiement d'un ERP	150
9.4.2.1	Sélection de l'ERP	151
9.4.2.1.1	Première phase	152
9.4.2.1.2	Deuxième phase	152
9.4.2.2	Déploiement	154
9.5	Mise en œuvre des mécanismes et mise en correspondance des modèles	154
9.5.0.3	Mise en correspondance	161
9.5.1	Mécanisme d'extension	164
9.5.2	Autres mécanismes	166
9.6	La capitalisation	167
9.7	Conclusions	170
	Conclusions générales et perspectives	173
	Annexes	181
	A Grille de sélection	181
	B Unified Modeling Language	183
	C Méthodes de modélisation d'entreprise	205
	D MDA - Model Driven Architecture	221
	Bibliographie	225

Table des figures

1.1	Décomposition d'une entité en systèmes	8
1.2	Système de traitement de l'information et système informatique	10
1.3	Cadre conceptuel suivant les niveaux de management	11
1.4	Positionnement des outils de gestion	14
2.1	Du MRP au SCM	17
2.2	L'évolution des modes de gestion et des courants de pensée	21
2.3	CIM et outils complémentaires	22
2.4	Principaux acteurs du marché des ERP	26
4.1	WBS d'un projet ERP	48
4.2	La spirale adéquation/configuration/prototypage	50
5.1	Représentation macroscopique du processus sur l'architecture logique	64
5.2	Enchaînement d'activités - Processus de « Ventes »	66
5.3	Architecture physique	67
5.4	Modèle 3-tiers	70
5.5	Architecture logique d'un système basé sur J2EE	71
6.1	Critères de jugement de la qualité du logiciel	74
6.2	Cycle en V	76
6.3	Cycle de vie en spirale	77
6.4	Démarche projet : méthode USDP	78
6.5	Décomposition hiérarchique d'un système	80
6.6	Enchaînement d'activités avec le formalisme SADT	81
6.7	Le cadre de référence d'une architecture d'entreprise selon Zachman	82

6.8	Repérage dans les trois dimensions	83
7.1	Décomposition d'un processus d'entreprise	90
7.2	Classification des normes par domaine	92
7.3	Évolutions des normes et des méthodes	92
7.4	Relation entre « constructs »	94
7.5	Cadre de conception de la norme ENV 40003	95
7.6	Le modèle SADT/IDEF0	96
7.7	Exemple de grille GRAI	98
7.8	Réseau d'activités	98
8.1	Méta-modèle de CIM-OSA	108
8.2	Méta-modèle de UEML	109
8.3	Proposition de méta-modèle	110
8.4	Méta-modèle partiel - point de vue fonctionnel	113
8.5	Diagramme de classes du processus domaine « Vente »	113
8.6	Cadre de description des processus domaine	114
8.7	Représentation des processus domaine avec les cas d'utilisation	114
8.8	Représentation comportementale du processus domaine « Vente » par un diagramme d'activités	115
8.9	Enchaînement des activités pour le processus métier « Commande de vente »	116
8.10	État d'un processus ou d'une activité	116
8.11	Méta-modèle partiel : point de vue informationnel	117
8.12	Diagramme de classes des objets de l'entreprise	117
8.13	Diagramme de classe détaillé pour la commande	118
8.14	État de l'objet « commande » de l'entreprise - diagramme d'états transitions	118
8.15	Méta-modèle partiel : point de vue des ressources	119
8.16	Hiérarchie de rôle	120
8.17	Relation entre les rôles et les processus domaine	120
8.18	Relation entre les acteurs et les processus à l'aide du diagramme d'activité	121
8.19	Hiérarchie des machines de l'entreprise	122
8.20	État d'une ressource	122

8.21	Méta-modèle partiel : point de vue organisationnel	122
8.22	Structure de l'organisation	123
8.23	Exemple d'organisation tiré de la structure précédente	123
8.24	Exemple de grille de pilotage de la méthode GRAI	125
8.25	Ordre de consultation des diagrammes UML	129
8.26	Grille GRAI à l'aide du diagramme des cas d'utilisation organisé en paquetages	130
8.27	Grille GRAI à l'aide du diagramme d'activités organisé en macro-activités . . .	131
8.28	Représentation du modèle d'architecture 4+1 vues	132
8.29	Intersection entre le modèle 4+1 vue et les quatre points de vue de la norme . .	132
9.1	Mise en relation du modèle d'entreprise et modèle de référence	139
9.2	Organisation des différentes étapes de représentation via UML	140
9.3	Mise en relation et utilisation des différents diagrammes	141
9.4	Démarche dans la mise en œuvre d'un Système d'Information	142
9.5	Démarche de mise en œuvre proposée par Pascal Boutin	144
9.6	Convergence de modèles	145
9.7	Positionnement de notre démarche par rapport à l'existant	145
9.8	Notre premier cadre de conception	146
9.9	Notre second cadre de conception	148
9.10	Démarche d'utilisation du cadre de référence pour une petite structure	149
9.11	Macro tâche d'un projet ERP	150
9.12	Conduite de projet « Première phase »	152
9.13	Conduite de projet « Deuxième phase »	153
9.14	Grille GRAI à l'aide des cas d'utilisation	156
9.15	Grille GRAI particulière à l'aide des cas d'utilisation	157
9.16	Réseau d'activité de « enregistrer commande » à l'aide du diagramme d'activité	158
9.17	Liste des machines-outils et description de leurs capacités	159
9.18	Affectation des rôles du système à des personnes physiques	159
9.19	Relation entre les classes « Rôle », « Personne » et « Unité »	160
9.20	Instanciation d'un rôle, d'une personne et d'une unité	160
9.21	Description des codifications des objets d'entreprise	161

9.22	Domaine fonctionnel de la solution logicielle	161
9.23	Enchaînement d'activités pour « enregistrer commande » (modèle d'entreprise)	162
9.24	Enchaînement d'activités pour « enregistrer commande » (modèle de référence)	163
9.25	Enchaînement d'activités pour « enregistrer commande » (niveau conception)	163
9.26	Effort de maintenance par rapport aux développements de spécifiques	164
9.27	Modèle conceptuel de l'application (sans spécifique)	165
9.28	Évolution du modèle avec le même comportement (sans spécifique)	165
9.29	Évolution du modèle avec un nouveau comportement (avec spécifique)	166
9.30	Exemple de gestion de spécifique (sans)	167
9.31	Exemple de gestion de spécifique (avec)	167
9.32	Cadre de référence - Extension particulière	168
9.33	Cadre de référence - Extension partielle	169
9.34	Cadre de référence - Extension générique	169
B.1	Décomposition fonctionnelle et hiérarchique d'un système	183
B.2	Quatre niveaux de modélisation	188
B.3	Pattern de conception d'une vision abstraite à un cas concret	190
B.4	Pattern de conception « composite »	190
B.5	Pattern de conception « composite » dans l'assemblage de produits	191
B.6	Évolution d'UML	193
B.7	Exemple de mise en œuvre des relations entre cas d'utilisation	196
B.8	Extrait du méta-modèle de la représentation des classes	197
B.9	Relation d'association	198
B.10	Relation d'agrégation	198
B.11	Relation de composition	198
B.12	Relation de généralisation	198
B.13	Association porteuse d'attribut	198
B.14	Extrait de la représentation des diagrammes de séquence	200
B.15	Diagramme d'états-transitions	201
B.16	Diagramme d'activités	202
B.17	Représentation d'une architecture via les diagrammes UML	203

C.1	Activité vue par IDEF0	206
C.2	Activité vue par CIMOSA	210
C.3	Réseau d'activités	215
C.4	Grille GRAI - Business Plan	216
C.5	Grille GRAI - Gestion à l'affaire	217
C.6	Activité vue par MECI	218
D.1	Le modèle d'architecture MDA	222
D.2	Vision du MDA	223

Liste des tableaux

2.1	Deux styles différents pour l'intégration	23
2.2	Revenus des grands éditeurs (en milliers de dollars)	27
2.3	Parts de marché des éditeurs en France	28
2.4	Part de marché par taille de l'entreprise (Vente licences)	29
3.1	Répartition des PME/PMI par leur taille	33
3.2	Dynamique des projets sur le marché français des éditeurs d'ERP	34
3.3	Durée de vie estimée d'un ERP	35
4.1	Dépenses engagées	46
5.1	Représentation comparée des démarches d'architecte et d'ingénieur	58
6.1	Les modèles du Système d'Information pour MERISE	85
7.1	Normes européennes	92
7.2	Positionnement des différentes méthodes de modélisation d'entreprise	100
8.1	Spécification du processus Commercial	111
8.2	Relation entre processus métiers et activités du processus domaine de « Vente »	111
8.3	Relation entre processus métiers et activités du processus domaine d'« Achat »	111
8.4	Relation entre les activités et les objets d'entreprise	112
8.5	Table de correspondance entre notre méta-modèle et les éléments du langage UML	124
8.6	Bilan des choix de modélisation	128
9.1	Correspondance Modèle d'Entreprise - Modèle de Référence	136
9.2	Répartition des neuf diagrammes en classes de représentation du SI	140

9.3	Apports d'UML aux différentes étapes, phases et tâches d'un projet ERP . . .	143
A.1	Exemple de questions de la grille de sélection.	182
C.1	Liste des méthodes IDEF	206
C.2	Description du vocabulaire de la suite IDEF	209
C.3	Description du vocabulaire de CIMOSA	213
C.4	Description du vocabulaire de MECI	220

Liste des acronymes

ADL.....	Architecture Description Language
AFITEP.....	Association Francophone de Management de Projet
API.....	Application Program Interface
APICS.....	American Production and Inventory Control Society
BPR.....	Business Process Reengineering
CIM.....	Computer-Integrated Manufacturing
CIMOSA.....	CIM Open System Architecture
CNIL.....	Commission Nationale de l'Informatique et des Libertés
CWM.....	Common Warehouse Metamodel
EAI.....	Enterprise Application Integration
EPC.....	Event Process Chain
ERP.....	Enterprise Resource Planning
GIM.....	GRAI Integrated Methodology
GPAO.....	Gestion de Production Assistée par Ordinateur
GRAI.....	Graphes à Résultats et Activités Interreliés
GRC.....	Gestion de la Relation Client
ICAM.....	Integrated Computer Aided Manufacturing
IDEF.....	Icam DEFinition
IHM.....	Interface Homme-Machine
J2EE.....	Java 2 Entreprise Edition
MDA.....	Model-Driven Architecture
MERISE.....	Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entre- prise
MES.....	Manufacturing Execution System
MOA.....	Maîtrise d'OuvrAge
MOE.....	Maîtrise d'Œuvre
MOF.....	Meta-Object Facility
MRP.....	Material Requirements Planning
MRP II.....	Manufacturing Resource Planning
MVC.....	Modèle Vue Controleur
OCL.....	Object Constraint Language
OMG.....	Open Management Group

OOM	Oriented-Object MERISE
PGI	Progiciel de Gestion Intégré
PME	Petite et Moyenne Entreprise
PMI	Petite et Moyenne Industrie
PQL	Plan Qualité Logiciel
RAD	Rapid Application Development
ROI	Return On Investment
SADT	Structured Analysis Design Technique
SCE	Supply Chain Execution
SCM	Supply Chain Management
SCP	Supply Chain Planner
SGBD	Système de Gestion de Bases de Données
SI	Système d'Information
TFP	Trous Fonctionnels Potentiels
TIC	Technologies de l'Information et de la Communication
TPE	Très Petites Entreprises
UEML	Unified Enterprise Modeling Language
UML	Unified Modeling Language
USDP	Unified Software Development Process
WBS	Work Break-down Structure
XMI	XML Metadata Interchange
XML	eXtended Markup Language

Il faut beaucoup de talent pour faire rire avec des mots. Mais il faut du génie pour amuser avec des points de suspension... (Frédéric Dard)

Remerciements

Les travaux de thèse présentés dans ce mémoire ont été réalisés dans le cadre d'une convention CIFRE entre le laboratoire de génie industriel de l'École des Mines d'Albi-Carmaux et la société SYLOB, éditrice et intégrateur d'un ERP.

Tout d'abord, je souhaite remercier mes encadrants : M. Hervé PINGAUD, directeur de thèse, M. Paul GABORIT, encadrant de thèse et M. Jean-Marie VIGROUX, PDG de la société SYLOB. Durant ces quelques années de collaboration, ils m'ont permis d'atteindre l'objectif en réglant périodiquement le viseur à partir de leurs savoirs, leurs expériences, leurs qualités humaines et leurs encouragements afin de toucher la cible dans le dix.

Je tiens à exprimer ma gratitude à toutes les personnes ayant participé de près ou de loin à ces travaux de thèse. Je pense plus particulièrement à François DOLIDON pour ses connaissances sur le houblon et Frédérick BENABEN pour ses connaissances sur la théorie des jeux.

J'adresse également mes remerciements aux personnes qui ont accepté de participer à mon jury de thèse :

M. Jean-Pierre CAMPAGNE, professeur à PRISMA de l'INSA de Lyon, qui m'a fait l'honneur d'être rapporteur de ma thèse et de présider ce jury.

M. Michel SCHNEIDER, professeur au LIMOS de l'ISIMA de Clermont-Ferrand, qui m'a fait l'honneur d'être rapporteur de ma thèse.

M. Christian BRAESCH, maître de conférence au LISTIC d'Annecy, pour l'intérêt qu'il a porté à ce travail et pour avoir accepté de participer à ce jury.

MM. Lionel DUPONT et Frédérick BENABEN, respectivement directeur et maître-assistant du laboratoire de génie industriel de l'EMAC, pour avoir accepté de participer à ce jury en tant que membres invités.

Je tiens à remercier toutes les personnes qui m'ont soutenu durant cette période : collaborateurs de la société SYLOB, personnels de l'EMAC et l'équipe de MASC. Je pense, plus particulièrement, à l'équipe du centre génie industriel : Didier, Franck F., Jacqueline, Jacques, Marc-André et Michel. Je n'oublie pas les thésards : David, Guillaume, Khaled, Lionel, Marc, Matthieu L. et Schéhérazade que j'ai pu côtoyer, et la génération de l'avenir : Carmen, Elise, Jahouer, Jihed, Matthieu D., Naly et Thomas à qui je souhaite la plus grande réussite. Je ne saurais oublier Isabelle sans qui [...] je n'aurais pu découvrir le cinéma d'art et d'essai de l'Athonor.

Je tiens à remercier aussi les banlieusards d'Albi, tels qu'Alain, Angélique, Caroline, Claude, Jean-Pierre et Nicolas pour les discussions ou les collaborations scientifiques, mais aussi leur soutien jusqu'à la dernière minute.

Je pense aussi aux personnes qui m'ont accompagné dans cette vie albigeoise : Delphine et Sophie pour m'avoir supporté comme « supporter », Catherine, Elise, Isabelle, Karine, Laetitia, Mariem, Nadia, Pascale, Petra et Souhila pour leur gentillesse et bien sûr Sandrine pour avoir choisi la mauvaise personne [...] en tant que témoin.

Enfin, une pensée particulière pour Aurélie, Delphine, François et Noah qui m'ont supporté avant d'être docteur et me supporteront encore, enfin je l'espère.

À ma sœur

À mes parents

À ma famille

Introduction

La gestion industrielle a notablement progressé au cours des dix dernières années sous le triple effet :

- d’une modification du paysage industriel fortement marquée par une révision des relations de l’entreprise avec ses fournisseurs et ses clients (fonctionnement en réseau, chaîne logistique, entreprise étendue) d’une part, et par un élargissement des marchés jusqu’aux limites extrêmes de la mondialisation, d’autre part ;
- d’une volonté très forte des entreprises de mieux maîtriser leurs outils de production et les processus de gestion qui guident leur pilotage ;
- d’une évolution formidable de l’informatique, tant au niveau des architectures (réseau, système d’exploitation, client/serveur, intranet, Internet et Web) que des applications (SGBDR, IHM, AGL, progiciels) qui a servi de courroie de transmission pour concrétiser ces volontés de progrès.

Cette évolution se mesure sur le marché des applications logicielles de gestion, qui a vécu une période d’expansion fantastique (le chiffre d’affaires mondial des éditeurs d’ERP est passé d’1 milliard de dollars pour l’année 1990 à 84 milliards de dollars en 2002¹ et, en France, le chiffre d’affaires des éditeurs est de l’ordre du milliard d’euros en 2002²). Les habitudes ont effectivement changé. Peu d’industriels ont encore la velléité de développer leur propre solution informatique de gestion. Pour des raisons de qualité technique et de coût de revient, on achète donc une solution chez un éditeur. Le progiciel est devenu une composante du Système d’Information et un support incontournable du fonctionnement de l’entreprise. La transition d’une offre plutôt sectorielle et basée sur des métiers de gestion (gestion de production, gestion financière et comptable, gestion des ressources humaines, gestion de la maintenance...) à une offre plus universelle et surtout plus intégrale (progiciels de gestion intégrés) a donné lieu à une remise en cause profonde de l’organisation des entreprises autour de leur Système d’Information. La solution informatique ne doit plus être évaluée seulement par son degré d’adéquation aux besoins relatifs à chaque métier, mais elle doit posséder une capacité intrinsèque à supporter les flux d’informations entre les métiers pour apporter fluidité, rapidité et qualité aux échanges entre les différents acteurs. L’entreprise peut, dès lors, imaginer optimiser un fonctionnement mieux compris, et s’organiser pour une plus grande satisfaction du client. Mais cet objectif ambitieux ne peut être atteint que par une connaissance profonde du fonctionnement de l’entreprise dans sa totalité, et pas par une somme de connaissances partielles (les métiers). Il appelle également une aptitude à exprimer les besoins et à les transmettre afin de fixer les attendus (les spécifications) du Système d’Information. Enfin, parce que la conception du Système d’Information passe par l’acte d’achat du progiciel, il faut être à même de sélectionner le (ou les bons) éditeur(s) partenaire(s) et d’adapter leur produit.

Les SSII qui animent le marché de l’informatique de gestion se sont inscrites, peu ou prou, dans cette modification de trajectoire. C’est ainsi que sont nées de très grosses sociétés qui

¹<http://www.fortune.com>

²<http://www.idc.fr>

ont su, très tôt, proposer des produits intégrés. Après avoir vécu une expansion forte autour d'affaires avec des grandes entreprises sur des projets souvent supérieurs au million d'euros, ces sociétés ont fait le constat d'une saturation du marché des grands comptes et s'intéressent aujourd'hui à des entreprises de taille plus petite, en particulier aux PME/PMI. Mais cette nouvelle cible s'accompagne d'une augmentation notable de la concurrence puisque les acteurs des marchés plus traditionnels, souvent déjà en place chez ces industriels (éditeur de solutions comptables, de GPAO...), réagissent en évoluant, eux aussi, vers des solutions plus complètes. C'est ainsi que par stratégie d'acquisition d'applications complémentaires, ou par stratégie de fusion avec des partenaires, la dynamique du marché des progiciels de gestion intégrés est importante (Microsoft rachète Navision en 2002 pour 1,45 milliard d'euros, PeopleSoft rachète JD Edwards en 2003 pour 1,68 milliard de dollars tandis qu'Oracle lance une OPA hostile sur PeopleSoft pour 6,38 milliards de dollars), alors que les marchés des solutions spécialisées ont connu, de cause à effet, des difficultés.

Le retour d'expériences montre qu'il est difficile pour les éditeurs qui possèdent les parts les plus importantes du marché, tels que SAP ou ORACLE, de rentrer massivement sur ce nouveau secteur des PME/PMI. Nous essaierons d'analyser les sources de difficulté. Sans entrer dès maintenant dans le contenu de cette analyse, nous pouvons citer deux facteurs importants dans ce type d'entreprise :

- les fortes contraintes imposées par les ressources sur ces projets de SI qui pénalisent l'expression du besoin,
- une réserve vis-à-vis de ce type de progiciels due à la complexité inhérente à leur utilisation au sein de l'organisation.

La définition de l'offre commerciale des éditeurs est donc un facteur clé de succès pour l'avenir de ce marché. Ceux-ci sont pris entre deux feux. D'abord, la nécessité évoquée ci-dessus d'offrir des solutions adéquates (adaptées au système en place, rapidement opérationnelles, ouvertes aux nouveaux modes d'échange avec l'environnement) à des organisations en pleine mutation. Puis, la capacité à accompagner des entreprises de taille modeste dans des choix de technologies pointues, qui présentent des risques et dont le retour sur investissement est difficilement quantifiable. Les stratégies des éditeurs oscillent entre le « tout standard », la solution en kit et la configuration légère, mais personnalisée.

La thèse a été réalisée en collaboration avec une société de services qui édite et déploie un logiciel de type ERP pour des PME/PMI. Cette société fait aujourd'hui face aux challenges que nous venons de présenter.

Ses dirigeants ont pris conscience de la nécessité de faire des sauts technologiques au niveau des architectures matérielles et logicielles, d'une part, et de créer une offre originale qui renforce la capacité de l'entreprise dans la gestion de la relation à ses clients, d'autre part. Dans ce cadre, c'est la gestion du projet qui est le cœur de cible avec en point de mire une meilleure qualité dans l'expression du besoin, et un suivi technique très rigoureux du déploiement et de l'exploitation du produit par chaque client. Même s'ils ne ferment pas la porte à une offre commerciale basée sur du standard, les cadres de la société ont exprimé le besoin d'outils supportant la définition du besoin et la capitalisation de la solution particulière mise à disposition du client, pour chaque affaire. Ils ont donc opté pour une liaison avec les milieux universitaires pour préparer ces sauts technologiques, et tout particulièrement pour lancer une expérimentation sur l'apport de la modélisation dans le suivi des affaires. En résumé, les prochaines versions du produit devront apporter des facilités dans la gestion de la qualité technique des projets.

C'est une hypothèse : le contenu fonctionnel du produit, tel qu'il est dans les versions actuelles, est satisfaisant. Il contient l'essentiel des techniques de gestion industrielle dont les entreprises de cette taille ont besoin. Ce n'est donc pas à ce niveau que nous produirons notre effort. Nous

le répétons, c'est la capacité à mieux utiliser ce contenu pour répondre aux attentes de chaque client qui est l'enjeu principal.

Compte tenu de la richesse fonctionnelle de ces ERP, suivre de manière détaillée chaque client conduit à gérer une forte diversité. La conception d'une nouvelle version plus moderne de l'ERP doit tenir compte de la charge de travail en développement et en maintenance. Les méthodes de développement doivent répondre de manière optimisée à cette gestion de diversité. Il y a eu des progrès notables en génie logiciel, ces dernières années. Ces développements sont en phase avec les propriétés recherchées par les applications : ouverture à l'échange avec d'autres applications, souplesse des accès aux données, facilité de maintenance corrective et évolutive. S'il faut retenir un fait saillant dans l'évolution du génie logiciel, c'est l'arrivée en force de la modélisation comme étape initiale du processus de développement. Nous avons opté pour l'établissement d'un lien fort entre la spécification fonctionnelle du besoin du client et l'analyse détaillée de ce besoin qui dirige le paramétrage de la solution ERP. Là encore, la modélisation a un rôle important à jouer pour bâtir un référentiel du logiciel et l'entretenir. La configuration de chaque client se représentera par un référentiel d'entreprise tiré du référentiel du logiciel par des mécanismes qui seront détaillés. Notre pari porte sur l'usage du même langage de modélisation UML pour représenter les deux parties (référentiel du modèle d'entreprise et référentiel du modèle d'application), ouvrant ainsi des perspectives pour développer des mécanismes de configuration personnalisée.

Notre plan est composé de trois grandes parties reprenant en détail ce discours pour présenter nos opinions, l'état de l'art dans différents domaines scientifiques dans lesquels nous devons puiser des concepts, des méthodes ou des outils, et illustrer le résultat du travail.

La première partie est centrée sur **une étude du Système d'Information des PME/PMI**. Quelques rappels sur la définition du Système d'Information et des éléments qui le composent seront réalisés. Les progiciels de gestion industrielle seront présentés plus en détail, puisque nous les considérons comme un élément central dans le Système d'Information d'entreprises de production. Un bref état des lieux du marché des ERP permettra de fixer les enjeux du travail de thèse. Puis nous chercherons à dresser un bilan des évolutions de ces organisations industrielles, et à mesurer les besoins induits sur les Systèmes d'Information d'une manière générale, et sur les ERP plus spécifiquement. Puisque le sujet se trouve à l'intersection du fournisseur de logiciel et de ses clients, nous ne pouvons pas ignorer le cadre utilisé pour les échanges entre ces deux parties qui est la gestion de projets. La gestion des projets ERP est un sujet qui a été très étudié depuis plus de cinq ans car ce sont des projets qui ont mauvaise réputation. Nous ferons un bilan de la connaissance acquise sur ce point. Cela nous permettra de mettre en évidence les phases dans le cycle de vie du projet où des progrès ont été réalisés, et d'autres où des progrès sont encore attendus. Notre contribution sera définie par rapport à ces attentes.

C'est **la conception d'un ERP** qui est l'objet principal de la seconde partie. La définition du métier d'architecte de Système d'Information servira de tremplin pour poser les bases de notre vision d'un projet ERP. Les notions de points de vue architecturaux, et de points de vue de modèle d'entreprise seront détaillés. Nous présenterons aussi les mécanismes de conception qui ont été utilisés pour réaliser les premières générations de ces progiciels. Cela nous permettra de faire une analyse critique de leur architecture et de leur capacité à relever les défis des nouvelles technologies de l'information et de la communication. Ces mécanismes de conception sont basés sur les méthodes de développement des projets informatiques qui étaient jugés robustes et classiques dans les années 90. La dynamique des progrès en matière de développement informatique est telle que ces mécanismes, et les organisations de projets qui les accompagnent, ont évolué aujourd'hui. Nous ferons donc un zoom sur les dernières innovations dans ce domaine, qu'il s'agisse de langage, d'architecture matérielle et logicielle, d'atelier de

développement ou de standard. C'est en effet des avancées notables dans chacun de ces secteurs qui fournissent un résultat d'ensemble que nous qualifions de saut technologique. L'approche objet se taille la part du lion dans cette évolution générale. Le besoin de séparer les parties logiques et physiques dans le développement du logiciel, ouvrant les portes à plus de portabilité et de compatibilité entre les systèmes, ainsi que les démarches qualité pour le développement d'applications avec des équipes d'analystes et programmeurs, ont fait émerger la modélisation comme fil conducteur du développement. Les ateliers de génie logiciel, dont l'usage se répand, proposent des facilités de modélisation dans la gestion des projets informatiques. En matière de modélisation objet, UML s'est imposé comme un standard. Nous en ferons une présentation. Puis, nous montrerons que la syntaxe de ce langage ainsi que la sémantique que nous lui attachons, fournissent des prémisses utiles à la modélisation d'entreprise.

Dans la troisième partie de ce mémoire, nous ferons **une proposition de cadre de référence pour la modélisation dans les projets ERP**. Après avoir analysé des travaux académiques proches de ce sujet, nous établirons les liens entre les différents modèles. Puis nous inscrirons l'utilisation de la modélisation dans le cycle de développement de la solution en utilisant une partie du cadre CIMOSA. Les différents diagrammes UML seront mis à contribution avec un objectif précis afin de tisser, de manière détaillée, des liens entre le modèle de l'entreprise, le modèle de référence de l'ERP, le modèle de l'ERP chez le client, puis l'application proprement dite. Les différentes évolutions du modèle au cours du projet doivent être instrumentées pour faciliter au maximum le travail des acteurs de l'équipe projet. Nous montrerons d'abord comment les propriétés caractéristiques de l'approche objet apportent naturellement des moyens de cheminer dans la démarche de conception. Un exemple d'application sera utilisé pour illustrer le contenu de ce cadre de référence.

La dernière partie du mémoire dresse les conclusions de ce travail et fournit des perspectives pour une suite des travaux au laboratoire de recherche et chez l'éditeur.

Première partie

Étude du Système d'Information des
PME/PMI

Chapitre 1

Systeme d'Information

1.1 Fruit de l'approche systémique

L'approche systémique est définie par Joël de Rosnay (1975) comme « *une méthodologie, permettant de rassembler et d'organiser les connaissances en vue d'une plus grande efficacité de l'action* ». Elle se finalise par une représentation générique des entités considérées, les systèmes, dans le but de repousser les limites imposées par la complexité de leur constitution et/ou de leur comportement. En ingénierie, les gains de connaissance obtenus par cette méthodologie se traduisent par des progrès dans les démarches de conception ou de pilotage de ces systèmes.

Par définition, un système est un ensemble d'éléments collaborant entre eux. Cet ensemble regroupe des composantes identifiées, abstraites ou physiques, interagissant pour atteindre un but ou un objectif commun. Chaque composante a un périmètre d'action déterminé, un environnement connu et échange avec cet environnement par des flux entrants et sortants de toute nature possible (matière, finance, énergie, information...). Par exemple, une ville, une entreprise, un ordinateur peuvent être considérés comme des systèmes particuliers.

Tout un courant de pensée s'est développé autour de cette méthodologie, citons Scherman C. Blumenthal (1969), Jean-Louis Le Moigne (1977, 1992) et Jacques Mélése (1990), pour l'appliquer à ces organisations particulières que sont les entreprises. C'est de ces travaux qu'émerge le SI (Système d'Information) défini comme un sous-système du système d'entreprise, et issu d'une décomposition que schématise la figure 1.1.

Ce premier stade de décomposition introduit trois sous-systèmes au même niveau de représentation :

- le système de transformation (dit aussi système opérant) des intrants en extrants. Il est composé de fonctions opérationnelles (ex : usiner) constituées de sous-ensembles physiques (ex : atelier d'usinage) dont la combinaison délivre un produit ou/et un service,
- le système de pilotage est divisé en niveaux décisionnels (centres de décision) délivrant les objectifs, les plans et les ordres d'exécution,
- le Système d'Information que nous allons définir plus en détail dans le paragraphe suivant.

Comme le souligne Bruno France-Lonard (2002), la théorie de la rationalité limitée dans la prise de décision, développée par Herbert A. Simon (1947), a largement participé à la reconnaissance du Système d'Information. Le modèle de décision à trois niveaux (stratégique, tactique, opérationnel) se croise avec celui de la partition en décision programmable et non

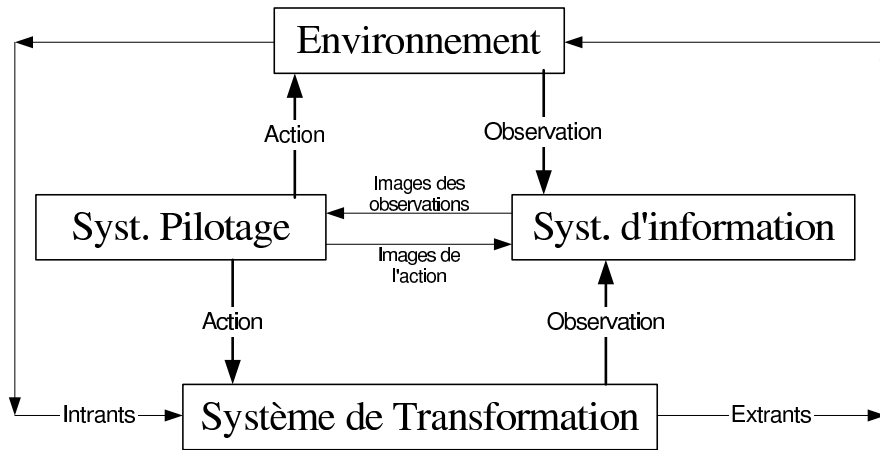


FIG. 1.1 – Décomposition d'une entité en systèmes

programmable. De ce croisement va découler un enseignement fort : les niveaux stratégiques doivent se concentrer sur des décisions non programmables qui sont de leur ressort, alors que les décisions programmables sont concentrées sur les niveaux opérationnels et là, la machine peut remplacer l'homme. Le terme de pilotage est plutôt emprunté aux théories de l'automatique, il fait référence à des actions qui forcent le système à évoluer dans une trajectoire particulière. L'approche systémique a su créer une synthèse de ces deux courants qui reconnaît que ces actions sont des décisions stratégiques et tactiques confiées au système de pilotage. Le système de pilotage est d'ailleurs parfois représenté dans une logique de contrôle/commande comme une boucle de rétroaction sur le reste du système. Le Système d'Information s'approprie effectivement l'essentiel des prises de décision programmables au plus près de l'exécution par la partie opérative, avec une vision hiérarchique classique du management. Mais le temps produit ses effets, le Système d'Information grignote peu à peu du terrain car l'information a de plus en plus de valeur dans l'entreprise. Le Système d'Information a donc tendance à apporter de plus en plus de soutien dans les mécanismes d'aide à la décision. L'art de manager ne se conçoit plus sans y inclure le rapport au Système d'Information.

1.2 Définitions

Dans les années soixante, la montée en puissance de l'informatique a permis de matérialiser le « Système d'Information », qui était alors assimilé à un système informatique applicatif, c'est-à-dire un ensemble d'applications de gestion au service de l'entreprise. Les premiers Systèmes d'Information ont permis d'automatiser des traitements répétitifs comme la comptabilité, qu'il s'agisse de la gestion de paie du personnel ou de la tenue des comptes par client, par exemple ((Jean-Louis Le Moigne, 1973), (Jean-Louis Peaucelle, 1999)).

Avec les progrès de l'informatique, l'organisation autour de ce moyen a pris corps. Le Système d'Information a alors acquis ses lettres de noblesse, devenant même un facteur clé de la stratégie de l'entreprise. Dans son ouvrage « Approche stratégique des SI », Robert Reix définit le Système d'Information comme :

« Un système composé d'entités diverses (employés, ordinateurs, réseaux, applications, bases de données, règles. . .) chargées de stocker et de traiter les informations relatives au système opérant afin de les mettre à disposition du système de pilotage. Il peut recevoir des informations du système de pilotage. Il peut émettre vers le système opérant des informations. » (Robert Reix, 2002)

Cette définition reprend la décomposition de l'approche systémique et la met en scène en évoquant les moyens utilisés (les entités qui participent au fonctionnement du Système d'Information) et les missions à remplir. Le système d'information a naturellement pour objet principal l'information. Celle-ci n'est autre qu'une donnée ayant acquis un sens particulier dans son contexte d'utilisation. Les fonctions de stockage et de transformation d'informations renvoient implicitement à une assimilation du SI à un procédé de production où la matière serait abstraite. Les trois phrases replacent ce sous-système dans son environnement faisant, là aussi, un parallèle avec des relations client/fournisseur vis-à-vis des deux autres sous-systèmes. Soulignons que parmi les entités du SI, les employés sont considérés comme des composantes à part entière. Gordon B. Davis et al. mettent en avant cette relation utilisateur-machine dans leur définition, qui est traduite par Jean-Louis Peaucelle de la façon suivante :

« Un système utilisateur-machine intégré qui produit de l'information pour assister les êtres humains dans les fonctions d'exécution de gestion et de prise de décisions. Le système utilise des équipements informatiques, des logiciels, des bases de données, des procédures manuelles et des modèles pour l'analyse, la planification, le contrôle et la prise de décision. » (Gordon B. Davis et al., 1985)

Cette définition, antérieure à celle de Robert Reix, précise ce que sont les missions de traitement de l'information. On fait référence à des activités classiques du métier de gestionnaire (manager). La liaison au système opérant se résume par des fonctions d'exécution, alors que celle relative au système de pilotage est décrite par la prise de décision. La deuxième partie de la définition permet d'ailleurs de dépasser le cadre strict du traitement de l'information pour entrer dans celui de la connaissance lorsqu'il est fait allusion à des modèles de représentation. Ces modèles de représentation sont formalisés à l'aide des mathématiques et sont capitalisés par voie informatique comme des systèmes d'équations (fonction, objectif, contraintes). Ils appartiennent bien au Système d'Information.

Peut-il y avoir un Système d'Information sans support informatique ? La réponse est affirmative. Depuis très longtemps, l'homme sait consigner l'information sous forme de dessin, de chiffres ou d'écrits, et a compris l'intérêt de faire des archives. Notons que les procédures de transformation de l'information et les règles au sein des organisations, fondées sur l'information, sont au moins aussi vieilles que ces archives. Les supports informatiques ont réformé des traitements qui étaient manuels à partir de la deuxième partie du 20^{ème} siècle, lorsque ceux-ci étaient suffisamment contrôlables pour être traduits sous forme d'un code exécutable. Toutefois, une vision moderne du Système d'Information ne peut plus se concevoir sans l'informatique.

Nous proposons d'enrichir ces définitions en utilisant trois points de vue complémentaires.

D'un point de vue fonctionnel, nous retiendrons que le Système d'Information est :

- capable de collecter l'information en provenance de la partie opérante,
- capable d'organiser cette information en vue de son traitement,
- capable de publier cette information à destination des puits d'utilisation concernés (utilisateurs ou autres systèmes),
- capable de mémoriser l'information, mais aussi de la connaissance sur l'organisation du travail,
- capable d'aider les acteurs de l'organisation dans l'exercice de leur métier, en particulier supporter la décision au sein de l'organisation.

D'un point de vue structurel, le Système d'Information est :

- vécu comme une suite d'interactions entre l'homme et des moyens informatiques,

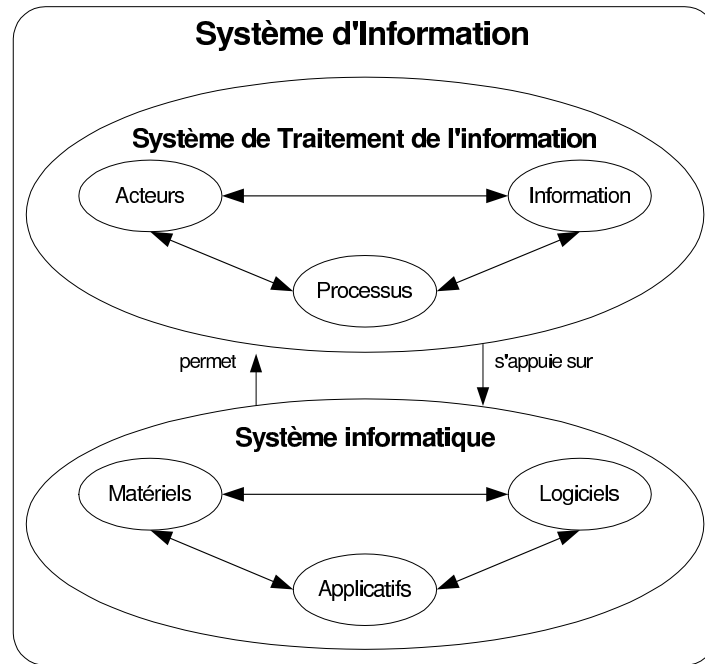


FIG. 1.2 – Système de traitement de l'information et système informatique

- décomposable au niveau des moyens en une collection d'entités matérielles et logicielles dont l'agencement est variable, et de plus en plus désigné par le terme d'architecture.

D'un point de vue organisationnel, nous retiendrons que le Système d'Information est tout à la fois :

- en osmose avec l'organisation, et en cela il est fortement sensible à une modification organisationnelle,
- un inducteur de mutation et de changement culturel pour l'entreprise.

Selon Mélissa Saadoun (2000), la raison d'être d'un Système d'Information est l'accès au bon moment à la bonne information pour prendre la bonne décision. Dans ce but, l'auteur propose une décomposition du système en trois sous-systèmes :

- le système organisationnel qui met en interaction le management (hommes, métiers, comportements et styles de management) et l'organisation (processus, procédures et structures),
- le système applicatif regroupe les applications de gestion, les applications de communication et la bureautique,
- le système informatique caractérise les infrastructures techniques.

Le système organisationnel exprime une demande qui a sa logique propre, celle d'un métier dans un contexte déterminé. Entre cette demande et l'offre du système informatique, l'entreprise exprime sa réponse dans un système applicatif.

La figure (fig. 1.2), inspirée des travaux de Chantal Morley, présente le système informatique comme un ensemble d'objets, *hardware* ou *software*, supportant le système de traitement de l'information. Ce dernier est fondé sur les éléments de l'organisation. Cette décomposition, plus simple puisqu'elle ne contient que deux sous-systèmes, place les applications au sein du système informatique.

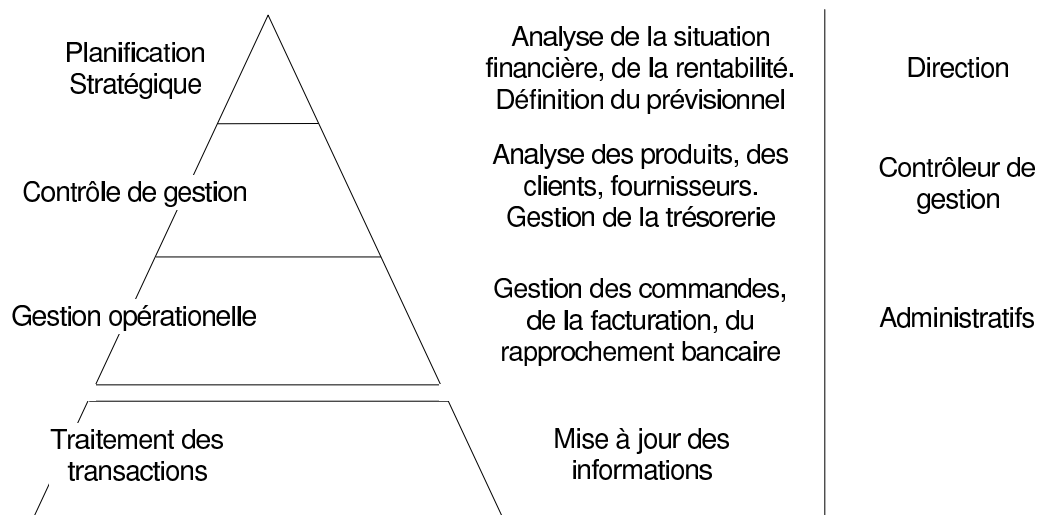


FIG. 1.3 – Cadre conceptuel suivant les niveaux de management

1.3 Les rapports difficiles du Système d'Information et de l'organisation

Le Système d'Information joue évidemment un rôle différent suivant la fonction des acteurs de l'entreprise. Gordon B. Davis et al. définissent une hiérarchie de besoins en fonctions des rôles dans la structure pyramidale d'une entreprise (fig. 1.3).

Comme le montre la figure 1.3, l'information requise par le niveau de la planification stratégique et l'information nécessaire au niveau de la gestion opérationnelle ont un contenu différent. Au plus bas de la pyramide, il y a des informations très nombreuses et de nature très variée, qu'il s'agisse des hommes, des produits, des procédés, des ateliers... À ce premier stade, l'information est une donnée assez brute avec une fréquence de changement de valeur assez élevée. Plus on monte dans les niveaux de la pyramide, plus la quantité d'informations doit être réduite par le simple fait que le nombre d'individus vers lesquels elle est publiée décroît. Il y a donc un besoin de regrouper les informations en utilisant des mécanismes d'agrégation. Les systèmes de gestion de base de données remplissent cette fonction. Ils mettent à disposition les moyens de transformer efficacement les données de base par des requêtes appropriées. Mais la valeur même de l'information ne doit pas souffrir des différentes constantes de temps des niveaux de la hiérarchie. La valeur d'un indicateur stratégique doit être en harmonie avec la fréquence de changement des données de base. Les acteurs de l'entreprise n'ont donc pas nécessairement un rapport à l'information qui se fasse par la voie hiérarchique. La démarche est plus centralisée, chaque niveau accède aux informations dont il a besoin par connexion directe à la base, au moyen de ses applications. L'informatique décisionnelle obéit à ce principe, les bases sont devenues des entrepôts de données (*datawarehouse*). Cet aspect centralisé peut conduire la direction de l'entreprise à fixer des limites sur les facilités d'accès aux informations en associant des profils aux utilisateurs en fonction de leur rôle. Les contraintes liées aux lois Informatique et Libertés imposent des précautions de cet ordre.

Mais pour être apte à délivrer une information de qualité, les bases de données de l'entreprise doivent être correctement renseignées. Le flux d'information représente le cheminement de celle-ci au travers de l'entreprise pour répondre à des logiques de déroulement de processus. À chaque étape, un acteur doit être en possession des informations délivrées par l'étape précédente. Si cette information transite par la base de données, l'acte de la saisie est une condition impérative de poursuite de processus. Il peut y avoir des mécanismes de protection

qui interdisent une transaction informatique parce qu'un processus a une réalité opérationnelle qui n'est pas en phase avec les états informatiques correspondants du simple fait d'un oubli de saisie. Mettons-nous à la place d'un acteur qui transmet une information à de nombreux autres acteurs « clients » de cette information. Son efficacité peut avoir une forte sensibilité sur la performance globale du système, et il subit nécessairement la pression par Système d'Information interposé.

Il est donc souvent traumatisant de passer d'une vision pro-active, quelque peu théorique, du Système d'Information d'entreprise à une réalité plutôt défensive des employés vis-à-vis du SI. Il y a un impact social dans les rapports de l'organisation à son Système d'Information. Nous aurons l'occasion d'aborder à nouveau ce sujet dans le chapitre sur la gestion de projets (§ 4).

De mauvais choix dans la conception du Système d'Information peuvent avoir des conséquences catastrophiques pour certaines catégories d'entreprise. Si elles se coupent de la relation à leur client parce que le système n'est pas compatible ou est en dysfonctionnement, les effets sur le chiffre d'affaires sont parfois très rapides et sans appel. Il existe de nombreux cas de ce type dans l'industrie du tourisme ou du transport aérien où les Systèmes d'Information sont devenus obligatoires dans la relation de l'entreprise à ses clients.

Au moyen de ces analyses ponctuelles et non exhaustives, nous souhaitons insister sur des facteurs non techniques qui pèsent sur les directions de Système d'Information. Élaborer un plan directeur des Systèmes d'Information est une activité à risque. Les technologies évoluent dans le but de pallier à ces dangers, mais les coûts et les délais ne permettent pas toujours de posséder les solutions les plus modernes. Une telle politique peut conduire à essuyer les plâtres si les technologies manquent de maturité. Alors entre passivité excessive et engouement frénétique pour les Systèmes d'Information, la recherche d'équilibre devient un nouvel art de manager.

1.4 Le Système d'Information et le génie industriel

Quel rapport existe-t-il entre le Système d'Information et le génie industriel ?

L'institut américain des ingénieurs en génie industriel¹ définit la discipline comme suit (Industrial Engineering, 2001) :

« Le génie industriel s'intéresse à la conception, à l'amélioration et à l'implémentation de systèmes intégrant les ressources humaines, la matière, le Système d'Information, les équipements et l'énergie. Il s'appuie sur des connaissances spécialisées et du savoir-faire en sciences mathématiques, sciences physiques et sciences sociales tout en appliquant les principes et les méthodes d'analyse et de conception des sciences de l'ingénieur pour spécifier, prédire et évaluer les résultats qui peuvent être obtenus par de tels systèmes. » (Industrial Engineering, 2001)

Cette définition met en avant trois éléments importants :

- le génie industriel se consacre à des systèmes, il n'y a pas *a priori* de restriction quant aux types des systèmes tant qu'ils sont de nature industrielle. Par ce positionnement au cœur de l'approche systémique, il inclut dès le départ le Système d'Information comme un des objets de préoccupation,
- l'exercice du métier est fondé sur une notion d'intégration de ressources diverses et l'on sait l'importance qu'a pris cette notion avec la complexification des produits, des technologies et des marchés,

¹<http://www.iienet.org/>

- dans le savoir du praticien de génie industriel, il y a des méthodes. Ce sont majoritairement des méthodes de gestion, gravitant autour des ressources évoquées précédemment : gestion de production, de projet, de qualité, des définitions techniques de produits, de connaissance... Il y a un rôle d'ensemblier dans l'exercice de l'art qui consiste, entre autres, à savoir sélectionner les méthodes et à les utiliser correctement. Le flux d'information joue un rôle de premier ordre dans ce cadre, puisqu'il se situe aux frontières (entrée et sortie) de ce traitement évolué d'information qu'est une méthode.

Il y a un rapport étroit entre le génie industriel et le Système d'Information. Ce serait très réducteur de considérer que le praticien du génie industriel n'est qu'un utilisateur avisé du Système d'Information. Les recherches en génie industriel sont orientées vers la définition et l'élaboration de méthodes. Les progrès des Systèmes d'Information sont donc en partie alimentés par l'implantation des nouvelles méthodes, de leurs avancées scientifiques et techniques. Par ailleurs, la dynamique de progrès de l'informatique est telle qu'elle apporte de nouveaux éléments de solution dans l'accès à des propriétés recherchées, telles que la flexibilité du système par exemple. Elle contribue à remettre en cause de manière régulière les hypothèses des problèmes de génie industriel. Les sciences de génie industriel peuvent donc aujourd'hui déployer des solutions d'une complexité considérée limitante il y a encore quelques années. Il y a une saine émulation entre les deux parties, et il est difficile de les dissocier.

Parmi ces méthodes, celles regroupées dans la gestion industrielle (§ 2.1.1) participent à la maîtrise des performances de l'entreprise dans la maîtrise des délais, dans la qualité et la gestion des coûts (Lionel Dupont, 1998). À titre d'illustration, nous citerons la gestion des stocks, la planification, la gestion des flux de matières et d'information, le management de projet, l'implantation de lignes d'ateliers, l'ordonnancement et, enfin, la qualité en production. Bien évidemment, l'information constitue le liant entre ces méthodes pour concourir à la performance. Dans certains cas, les méthodes seront traduites en algorithmes, où par choix judicieux des valeurs de certains paramètres, le code informatique correspondant produira un traitement automatisé : des décisions programmables seront confiées à la machine. L'exemple de la gestion de stock par point de commande pourrait tout à fait illustrer cela. Si la valeur du point de commande est connue de l'ordinateur, le code peut très bien déclencher la commande d'achats à partir des observations de l'évolution dynamique du stock, sans recours à une intervention de l'homme.

La gestion industrielle ne couvre pas toutes les méthodes de gestion dont l'entreprise a besoin. Néanmoins, le côté « ensemblier de méthodes » du génie industriel passe par une vision transversale sur les processus de gestion et sur les informations de l'entreprise. À une époque où la stratégie de l'entreprise se focalise sur sa chaîne de valeur et sur sa relation au client, ces processus transversaux sont devenus des fils rouges pour les dirigeants d'entreprise. Il est d'ailleurs naturel dans cette philosophie de parler d'alignement stratégique du Système d'Information lorsqu'on veut éviter que les courroies de transmission qui relient l'équipe dirigeante au reste de l'entreprise restent bloquées. La figure 1.4 est un tableau dont les colonnes représentent ce processus transversal qui va des fournisseurs aux clients d'une entreprise en traversant les fonctions opérationnelles de la chaîne de valeur.

Comme nous avons déjà largement insisté sur le rôle de la décision et du partage des responsabilités entre les différents niveaux hiérarchiques de l'entreprise, nous n'avons pas à commenter plus longuement les lignes de ce même tableau, qui correspondent aux échelles de temps.

Ce tableau a une définition qui correspond à une grille GRAI modélisant les centres de décisions de l'entreprise. Son usage est dérouté pour y consigner les fonctions remplies par les progiciels pour « faire » ces décisions ou aider à les « faire ».

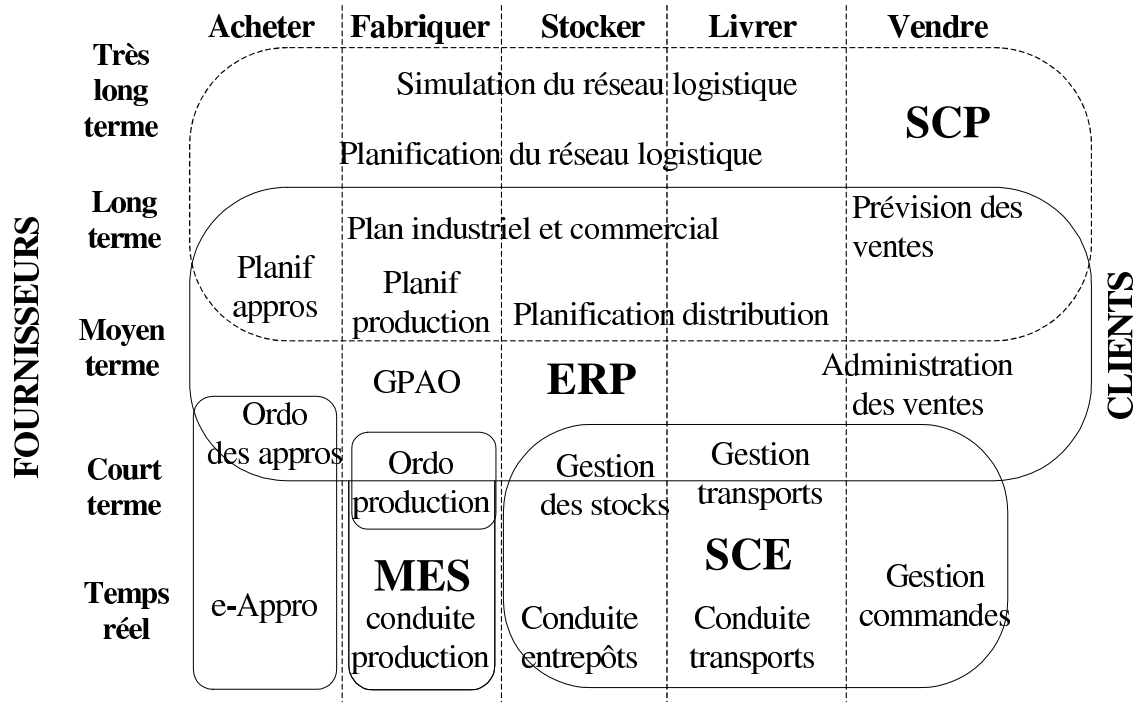


FIG. 1.4 – Positionnement des outils de gestion

Les premiers outils informatiques de gestion industrielle étaient centrés sur la gestion de production et sont apparus dans les années 60. Leur couverture fonctionnelle était limitée aux colonnes « fabriquer » et « stocker », et la portée des décisions ne dépassait pas le moyen terme. Ces outils se sont rapidement développés pour intégrer d'abord la colonne « acheter », puis la colonne « vendre », et ont formé une famille désignée par GPAO (Gestion de Production Assistée par Ordinateur). Peu à peu, la nécessité de prendre en compte la disponibilité des ressources de tout type (machine, homme, finance, énergie...) a élargi la couverture fonctionnelle des progiciels en entreprise. Au plus haut et au plus bas niveau, des solutions sont maintenant disponibles sur le marché pour partager des fonctions avec les partenaires de la chaîne logistique.

Une photo de l'état du marché est fournie par Thierry Courtois (2001). Nous y distinguons :

- les progiciels ERP (Enterprise Resource Planning) (§ 2.2) dits encore PGI (Progiciel de Gestion Intégré),
- les progiciels SCP (Supply Chain Planner) : outils permettant d'agrèger les données des différents Systèmes d'Information de partenaires différents dans le but d'optimiser la planification et de synchroniser les exécutions dans la chaîne logistique,
- les progiciels MES (Manufacturing Execution System) et SCE (Supply Chain Execution) : outils d'aide à très court terme ou temps réel des domaines de la production ou de la gestion des achats et des ventes (livraison).

Nous nous rendons bien compte, avec la figure 1.4, qu'il existe une difficulté dans le choix d'un outil adapté aux règles de l'entreprise. La direction informatique ne possède pas toutes les aptitudes pour respecter l'ensemble des contraintes suivantes :

- besoins du client : elle ne connaît pas toutes les méthodes de gestion industrielle,
- contraintes de mise en œuvre : si elle connaît les méthodes, elle ne peut pas avoir la connaissance de tous les raffinements dans leur mise en œuvre au sein de l'organisation (paramétrage du Système d'Information),

- contraintes d'exploitation : elle doit anticiper les besoins pour que l'acte d'achat ne soit pas rapidement obsolète car les habitudes de l'organisation évoluent, on lui demande de s'adapter de plus en plus fréquemment,
- contraintes industrielles : elle doit évaluer la difficulté dans la mise en œuvre d'un nouveau système par rapport au triptyque délais, coût et qualité, ainsi que considérer la réutilisation des anciens systèmes,
- contraintes contractuelles : elle doit prévoir un transfert de connaissance entre le prestataire de service et son organisation,
- contraintes humaines : elle doit prévoir les différents changements organisationnels et culturels qu'implique la mise en œuvre de nouveaux systèmes,
- contraintes techniques : elle doit effectuer un travail de veille sur l'évolution des normes et des nouvelles technologies de l'informatique,
- contraintes de l'environnement : elle doit s'assurer du respect de la législation, du contexte social et politique de l'entreprise (conseils de la CNIL (Commission Nationale de l'Informatique et des Libertés) par exemple).

La transition des directions informatiques vers les directions de Système d'Information est une forme de réponse apportée à ce problème. L'architecture du Système d'Information n'est pas seulement un exercice d'architecture informatique. Le passage par un achat de produits logiciels est une forme d'externalisation qui induit le développement d'une capacité à exprimer les besoins et à vérifier la qualité des choix effectués. Le praticien du génie industriel, par sa capacité d'ensemblier, a un rôle à jouer dans ce cadre des projets relatifs au Système d'Information. Il est un artisan de la juste adéquation du logiciel à l'organisation.

Chapitre 2

Progiciels de gestion industrielle

2.1 Évolutions des modes de gestion et des outils informatiques

Répondre aux besoins de la gestion industrielle des entreprises par des applications professionnelles, c'est le credo du marché des éditeurs de progiciels de gestion. Au cours de ces dernières années, les éditeurs ont introduit de nouvelles capacités dans leur offre, faisant écho aux évolutions dans les méthodes de gestion. Leur évolution s'est même souvent transformée en mutation au gré des modes dans le paysage des familles de progiciels, et des progrès des technologies informatiques. Nous essayons par la suite de faire une analyse basée sur un parallèle entre évolution historique des modes de gestion et évolution historique des solutions progicielles.

2.1.1 Modes de gestion

Sur la figure ci-dessous (fig. 2.1) sont reportées les grandes étapes de la transformation des méthodes de gestion industrielle.

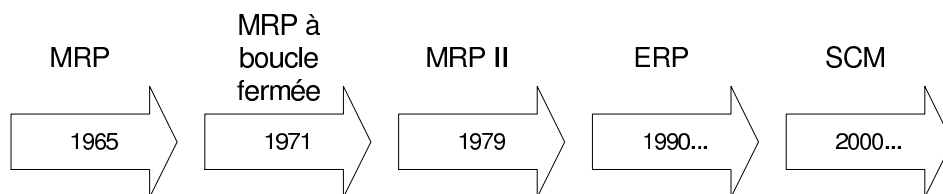


FIG. 2.1 – Du MRP au SCM

Nous reprendrons plus en détail les cinq étapes suivantes : MRP, MRP à boucle fermée, MRP2, gestion poussée des ressources, chaîne logistique.

2.1.1.1 MRP

Le MRP (Material Requirements Planning) est un processus de traitement d'informations basé sur des bilans de matière. Dans sa plus simple expression, le MRP est un simple calcul des besoins. Il définit les créneaux de lancement des ordres de fabrication et les besoins en matières premières par une remontée de l'information à contre courant du flux de matière,

des produits finis vers les composants et les matières premières. Ce sont les nomenclatures techniques, les délais provenant des gammes opératoires, les niveaux de stock et les besoins en produits finis qui structurent et alimentent ces calculs.

De plus, il est échéancé, il fournit des préconisations de ré-ordonnement des ordres lancés lorsque les dates de besoin ne sont pas en phase. Vu à l'origine comme un bon moyen de gérer les stocks, il est aujourd'hui considéré avant tout comme une technique de planification, c'est-à-dire une méthode pour établir et maintenir des dates d'exigibilité correctes.

2.1.1.2 MRP à boucle fermée

Le MRP à boucle fermée est un système bâti autour d'un MRP. C'est une évolution de la technique de base qui utilise les principes de la planification hiérarchisée ((Gabriel R. Bitrant et al., 1981), (Colette Mercé, 1987)). L'organisation planifie les activités en délivrant plusieurs catégories d'information sur des horizons de temps différents. En partant des niveaux les plus élevés de la hiérarchie, ce sont respectivement le Plan Industriel & Commercial (PIC), le Programme Directeur de Production (PDP) et la Planification des Besoins en Capacités (PBC). Ces fonctions supplémentaires représentent tout le niveau tactique de l'entreprise en gestion de production. Il s'établit des liaisons entre ces différents niveaux afin d'assurer un service de qualité en bout de chaîne avec des coûts de revient maîtrisés.

Une fois cette phase de planification réalisée et les plans acceptés comme réalistes et accessibles, les fonctions d'exécutions peuvent entrer en jeu. Ces fonctions incluent le contrôle de la production par la mesure des entrées / sorties (capacités), l'établissement du plan détaillé et sa distribution dans l'atelier, ainsi que l'état des retards à la fois de l'atelier et des fournisseurs, (plan fournisseur)...

2.1.1.3 MRP II

Le MRP II (Manufacturing Resource Planning) est une extension du MRP à boucle fermée, où l'étape de planification des besoins en composants est réalisée en tenant compte des capacités finies des ressources. Le calcul peut être itéré afin d'assurer une faisabilité minimale du plan vis-à-vis des exigences consignées dans le PDP.

C'est ce souci de programmer l'utilisation des ressources qui tisse un lien très fort avec le contrôle de gestion. En effet, le bon dimensionnement et la bonne utilisation des moyens mis à disposition sont acquis avec des techniques de planification incluant cette faculté de prédire les charges et de les affecter dans la limite du possible.

2.1.1.4 Vers une gestion poussée des ressources : l'ERP

Le MRP2 a ouvert la voie à un contrôle strict des ressources. Centrées sur les flux de matières, les différentes ressources productives de l'entreprise sont mises à contribution dans le cadre du plan de production. Les stocks de matière à tous les stades de transformation, les ressources matérielles (les machines, les pièces de rechange, les moyens de transport des pièces...), les hommes, l'énergie, les finances sont mobilisés pour satisfaire le processus de production dans une volonté de coordination maximale.

Cette gestion poussée des ressources est motivée par une recherche de coûts de revient industriels les plus bas. Peu à peu, tous les facteurs de coût sont étudiés à la loupe, et les sources

de gain sont exploitées. Les années 80 vont être une période d'effervescence où de nouvelles modes prennent corps. Les stocks, s'ils constituent une manière de sécuriser la bonne exécution des étapes de transformation, sont des immobilisations financières qui sont pourchassées. Le juste-à-temps va diminuer la valeur des immobilisations au plus proche des clients. Puis le Kanban va assurer la transmission des informations par une série de stimuli d'aval en amont, remontant dans les ateliers, instaurant une synchronisation forcée des stocks et des ordres de fabrication. La planification prévisionnelle est bousculée par ce concept qui prône une réactivité et une qualité totales. L'heure est à la tension des flux physiques sur le périmètre qui va des achats fournisseurs aux livraisons aux clients. Cette évolution va changer le rapport entre l'entreprise et l'information, on va développer les bases de données pour stocker l'information, et pouvoir l'exploiter rapidement. L'informatique se développe dans les ateliers, au plus proche des opérations. Il s'établit une sorte de concurrence entre :

- une philosophie fondée sur la planification prévisionnelle qui fournit un cadre confortable et qui demande des politiques d'adaptation en cas d'aléas,
- une philosophie fondée sur la réaction pour adapter le système en permanence à des demandes connues tard et qui base sa mise en œuvre sur la transmission des informations.

Du mouvement de la qualité, qui constitue une vague de fond qui a accompagné ces différentes évolutions de la production, il restera des traces... Le rapport de l'entreprise à ses informations se trouvera également réformé. À côté du facteur de coordination des ressources, il y aura la connaissance améliorée de chaque ressource. On a un besoin croissant de statistiques sur les données de base, comme les rendements, les taux de défaillance des machines, les taux de rebut, la non qualité sur les produits, les temps de travail et les productivités des opérateurs, les ruptures d'approvisionnement ou de livraison... pour générer un flux du bas de la hiérarchie vers les couches supérieures, et enrichir les prises de décision. Le volume d'informations à traiter devient alors très important (et même limitant dans les premiers temps). La traçabilité des produits devient une exigence des clients dans beaucoup de secteurs industriels.

Parallèlement, le concept de projets est en train de transférer dans les structures et organisations industrielles. Beaucoup d'entreprises, dont le métier repose sur des affaires différentes pour chaque commande, dont chaque produit est un cas particulier, trouvent un processus global de gestion qui cadre bien leurs activités. Les ressources matérielles, humaines et le calcul des coûts viennent s'associer à la planification des délais. L'information et son traitement sont des éléments importants et, comme en gestion de production, des structures sont établies (décomposition du travail ou WBS (Work Break-down Structure), décomposition du produit ou PBS...) Le champ d'application de la gestion se trouve étendu par la gestion de projets. Des services de R&D, aux sociétés de service, beaucoup d'organisations vont adopter ce moyen pour quantifier, affecter, puis contrôler leurs ressources.

2.1.1.5 SCM

Après une période d'introspection où l'entreprise se concentrait sur sa propre organisation et sur ses ressources internes, ces dernières années ont vu apparaître une nouvelle mode qui consiste à développer une coopération à l'échelle d'un ensemble de partenaires. C'est le concept de gestion de la chaîne logistique (Supply Chain Management, SCM). L'entreprise est incitée à travailler sur des sources de gain aux interfaces avec ses fournisseurs et ses clients. Il y a forcément un rôle central de l'information dans ce type de structure. On peut d'ailleurs représenter la chaîne logistique dans le cadre d'une vision systémique, et expliquer en partie son organisation par l'architecture de ses Systèmes d'Information (Hervé Pingaud, 2003).

Le SCM doit permettre de piloter les flux de l'entreprise et de ses partenaires depuis le fournisseur du fournisseur jusqu'au client du client. L'intérêt de ce concept est d'avoir une vision

du transport et de la production, de l'achat et des ventes de manière globale. Chaque maillon de la chaîne doit faire preuve d'un certain état de maturité pour rester actif au sein de la chaîne traitant l'information et la diffusant, dans le cadre des règles du partenariat.

Ces nouvelles organisations en réseau de partenaires ont fait naître un besoin spécial au niveau de la planification collaborative et du suivi des activités.

2.1.1.6 Évolution des méthodes

Les méthodes ont donc évolué avec le temps. Logiquement, les besoins à satisfaire par les logiciels et les besoins de données en entrée et sortie des méthodes implantées dans ces logiciels ont également changé. Nous évoquons les tendances suivantes pour résumer synthétiquement cette évolution :

- l'accès généralisé à l'information : les managers sont des consommateurs exigeants d'informations détaillées (ex : état d'avancement du produit) et d'informations agrégées (ex : taux de service des différentes machines),
- une couverture globale de l'entreprise ou de plusieurs entreprises : afin d'optimiser le fonctionnement, il est indispensable de prendre en compte toutes les ressources de l'entreprise (humaines, matérielles, informationnelles et symboliques).

Pour être complet, notre panorama se doit d'évoquer la difficulté de mise en œuvre du processus transversal de la figure 1.4. Le client est devenu de plus en plus exigeant en termes de délai, de coût et de qualité. Beaucoup de marchés ont adopté une stratégie de différenciation basée sur la diversité des options offertes aux clients pour un même produit (voiture, avion, ordinateur, maison...). Pour la gestion, ce positionnement a deux conséquences :

- une nécessité de gérer la diversité sur le plan technique, plus d'options, c'est plus de composants, des nomenclatures et des gammes plus compliquées, des planifications plus difficiles, des fournisseurs plus nombreux et sollicités différemment,
- une relation au client qui doit être personnalisée, et qui induit un véritable mode de gestion particulier pour la prospection, le contact, le profil de consommation, la négociation, le suivi...

La couverture globale dont nous parlions plus haut doit être appréhendée dans ce contexte de diversification où l'on quitte la voie du « tout standard » pour entrer dans l'ère du « sur mesure ». Le volume d'information est bien sûr impacté, et dans ce climat, la coordination entre les différentes ressources devient aussi extrêmement délicate.

2.1.2 Couverture logicielle

Les éditeurs de logiciels ont régulièrement modifié leur offre afin de proposer aux entreprises des moyens informatisés pour cadrer avec l'évolution de leur besoin. Les technologies informatiques ont progressé dans le traitement des informations, mais également dans les communications entre machines, ouvrant une large perspective aux éditeurs. Mais ces outils ont au moins autant évolué d'un point de vue fonctionnel que d'un point de vue architectural.

Sur la figure suivante (fig. 2.2), nous proposons un positionnement de l'offre qui est calqué sur l'évolution des modes de gestion.

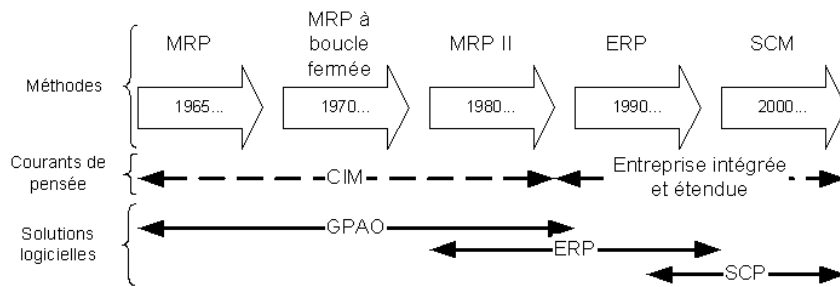


FIG. 2.2 – L'évolution des modes de gestion et des courants de pensée

2.1.2.1 De la GPAO à l'ERP

Au cours des années 60-70, les premiers systèmes de GPAO (Gestion de Production Assistée par Ordinateur) sont apparus sur le marché. Il s'agissait de produits de niche par leur couverture limitée et par rapport aux secteurs industriels cibles.

Les outils de GPAO sont composés de modules qui correspondent à des fonctions importantes du processus global de gestion : la gestion des approvisionnements, la gestion des stocks et la planification. Ce contenu correspond à l'implémentation d'un MRP au cœur de l'application. Cette famille d'outils a su évoluer au cours du temps pour introduire des capacités de planification hiérarchisée, puis des liens vers des ressources multiples.

D'une offre de niche recouvrant les besoins d'un domaine spécifique à un métier (processus verticaux), comme la comptabilité par exemple, nous sommes passés progressivement à une situation de forte intersection des offres, avec des solutions recouvrant plusieurs domaines de l'entreprise. Jusqu'au début des années 1990, seules la gestion de la production avec la GPAO et la gestion de la conception et fabrication de produit avec la CFAO, avaient ce trait caractéristique de chercher à toucher des processus transversaux. L'arrivée des ERP a changé la donne.

L'ERP a une large couverture fonctionnelle. L'intégrité des données y est affichée comme une garantie. La gamme des traitements d'information a une homogénéité telle qu'on peut effectivement y gérer des ressources de nature différente de façon cohérente. Jean-Louis Lequeux définit un ERP comme une application ayant au minimum les modules suivants :

- gestion commerciale,
- GPAO (Gestion de Production Assistée par Ordinateur)¹,
- gestion comptable et financière,
- ressources humaines,
- gestion client,
- gestion planning.

2.1.2.2 Du CIM à l'entreprise intégrée

Apparu au début des années 1970², le concept de CIM (Computer-Integrated Manufacturing) a très vite été dépassé. Le terme était associé à une conception utopique de l'usine sans homme,

¹seulement pour les entreprises industrielles

²Harrington Jr., Joseph, Computer Integrated Manufacturing, Robert E. Kreiger Co., FL, 1973.

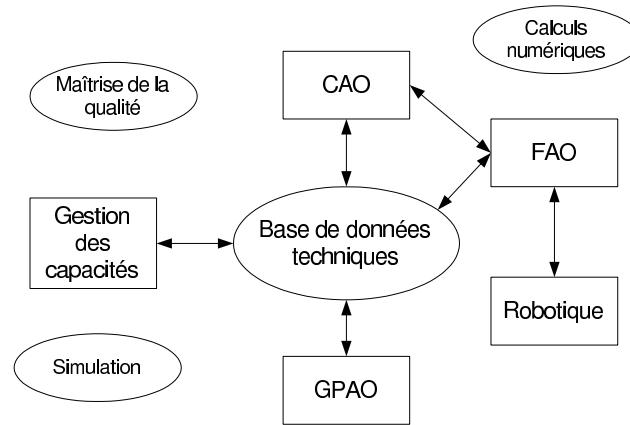


FIG. 2.3 – CIM et outils complémentaires

et fondé sur l'hypothèse que les techniques de l'automatique et de l'informatique permettraient de conduire les systèmes avec une autonomie presque totale. Les réalités sociales, humaines, économiques ont très vite réduit les ambitions du CIM.

Le CIM est un ensemble de méthodes, de techniques et de moyens informatiques au service de la technique de production, visant à accroître la productivité dans la conception, la fabrication et la gestion des produits fabriqués par une entreprise. Cet ensemble est généralement construit autour de la base de données techniques (Product Data Management) décrivant les gammes et les nomenclatures de produits. Un noyau central est connecté, alimenté, et/ou fournit des données aux outils suivants :

- conception et Fabrication Assistée par Ordinateur (CAM/CAD Computer-Aided Manufacturing - Computer-Aided Design),
- outil de calcul numérique (CAEA - Computer Aided Engineering Analysis),
- gestion des ateliers et des capacités machines (CAPP - Computer-Aided Process Planning),
- GPAO (Gestion de Production Assistée par Ordinateur) (CAMP - Computer-Aided Management and Manufacturing),
- simulation,
- robotique,
- la maîtrise de la qualité totale (TQC - Total Quality Control).

En parcourant cette liste d'outils, nous nous rendons bien compte qu'il est peu probable qu'un éditeur propose une solution complète et homogène afin de répondre à toutes les dimensions de l'ingénierie des produits et des procédés. Les outils de CIM, par leur complexité fonctionnelle, ont été obligés de développer des architectures originales, ouvertes sur les échanges et la collaboration. Tout un courant de standardisation a été nécessaire pour progresser dans cette voie.

Or, si les outils de type ERP sont généralement proposés sous forme de module, pour que chaque utilisateur puisse choisir son niveau d'intégration dans une architecture intégrée et homogène, on constate que la solution ne peut aujourd'hui émaner que d'un concepteur unique. Une architecture intégrée permet de connecter progressivement les différents modules suivant l'évolution du besoin fonctionnel. L'architecture homogène certifie, quand à elle, que les éléments constitutifs sont de même nature et peuvent travailler de concert (entre chaque module, les interfaces homme-machine sont harmonisées par exemple).

Force est de constater que l'évolution conjointe des mœurs industrielles, des moyens informatiques et des architectures de Système d'Information projette le CIM à nouveau sur le devant de la scène. La notion d'entreprise intégrée est au cœur de l'actualité, et se positionne comme une ambition (un peu mythique) de progrès pour les Systèmes d'Information.

Ce que l'on retire de ce constat, c'est que la course à l'intégration est un exercice de style où deux écoles s'affrontent. Le tableau 2.1¹ résume les approches de ces deux écoles. La première colonne est typique de la stratégie d'un éditeur unique avec une offre complète et de grande qualité. La seconde colonne s'inspire davantage d'une intégration tirée de la diversité en imposant simplement des règles du jeu. La gestion du projet de Système d'Information n'a évidemment pas du tout les mêmes caractéristiques dans chacun des cas.

De	Vers
Intégration standardisée	Intégration flexible
Système fermé	Système ouvert
Management autocratique	Management horizontal

TAB. 2.1 – Deux styles différents pour l'intégration

2.2 L'ERP

L'ERP, pour Entreprise Resource Planning, est une solution informatisée qui offre la possibilité de gérer l'ensemble des moyens de l'entreprise (Jean-Luc Deixonne, 2001) (ressources humaines, matérielles et financières) en privilégiant deux aspects : une communication entre les différents acteurs métiers et une cohérence de l'information. C'est le centre nerveux du Système d'Information.

Le terme ERP a été francisé sous l'appellation « progiciel de gestion intégré ». Cette traduction n'est pas fidèle au sens initial du terme anglophone. En effet, il fait surtout référence à un caractère technique qui est l'intégrité des données (intégré se rapporte à progiciel, et non à gestion). La référence à la méthode de gestion globalisée des ressources, à la prééminence de la planification, n'est plus explicite dans la traduction. Plusieurs auteurs ont contribué à la définition de la famille des ERP. Ces définitions ne sont pas tout à fait identiques, et elles ont évolué avec le temps. Ce dernier point n'est pas surprenant dans la mesure où les produits ERP ont des évolutions qui suivent les besoins du marché et l'essor des TIC (Technologies de l'Information et de la Communication).

2.2.1 Définitions

La définition donnée par le CXP², (Jean-Louis Tomas, 1999), regroupe un ensemble de caractéristiques pour décrire un ERP, telles que :

« Pour être intégré, un progiciel de gestion doit :

- émaner d'un concepteur unique ;
- garantir à l'utilisateur l'unicité de l'information, assurée par la disponibilité de l'intégralité de la structure de la base de données à partir de chacun des modules, même pris individuellement ;

¹<http://www.intranet.management.mcgill.ca/course/msom/MBA/mgmt-tec/students/cim/TEST.HTM>

²CXP Informations, février 1994.

- *reposer sur une mise à jour en temps réel des informations modifiées dans tous les modules affectés ;*
- *fournir des pistes d'audit basées sur la garantie d'une totale traçabilité des opérations de gestion ;*
- *couvrir soit une fonction (ou filière) de gestion, soit la totalité du Système d'Information de l'entreprise. »*

C'est une sensibilité très technique, proche du sens donné par la traduction française en se focalisant d'emblée sur la signification du qualificatif « intégré ». On y parle effectivement d'intégrité des données, d'une part, et d'intégration des fonctions avec une couverture pouvant être variable, d'autre part. La définition est peut être un peu floue lorsqu'elle fait référence à un concepteur unique. Il s'agit certainement, pour le CXP, de l'identification d'un éditeur n'étant pas une simple façade commerciale, mais assurant une véritable ingénierie du produit logiciel. La traçabilité de l'information est un point fort pour cet auteur, on ne la retrouvera pas sous forme explicite dans les autres définitions. Enfin, la notion de mise à jour en temps réel est une garantie indispensable pour rendre tangible le couple intégrité/intégration évoqué ci-dessus.

Dans la même période (Jean-Louis Lequeux, 1999) nous fournit une définition qui nous semble plus large :

« L'ERP est défini par un sous-ensemble de Système d'Information qui intègre les caractéristiques globales suivantes :

- *la gestion effective de plusieurs domaines de l'entreprise par des modules intégrés ou des progiciels susceptibles d'assurer une collaboration des processus. Deux types de processus existent : les processus horizontaux pour l'approche orientée client et les processus verticaux pour l'approche métier. Les modules intégrés les plus courants sont : GPAO, gestion commerciale, gestion comptable et financière, ressources humaines, gestion client ;*
- *l'existence d'un référentiel unique des données. Le référentiel est défini comme étant l'ensemble des références des données ainsi que les indications nécessaires pour retrouver les données elles-mêmes sur une base de données ;*
- *l'adaptation rapide aux règles de fonctionnement (professionnels, légales ou résultant de l'organisation interne de l'entreprise) ;*
- *l'unicité d'administration du sous-système applicatif ;*
- *l'uniformisation des interfaces homme machine ;*
- *l'existence d'outils de développement ou de personnalisation de compléments applicatifs. »*

Cet auteur a le mérite d'inclure l'ERP dans le périmètre du Système d'Information. Il évoque clairement les processus collaboratifs qui sont à l'origine du besoin d'intégration. En réfutant le concepteur unique, il anticipait l'évolution des technologies d'interopérabilité, offrant ainsi une certaine résistance de la définition à l'effet du temps. Par contre, il énumère un ensemble de contraintes qui vont assurer une harmonie d'ensemble, et qu'on serait en droit d'exiger d'un concepteur unique. Enfin, la définition fait état d'un facteur essentiel : l'adaptabilité de l'outil à l'entreprise, ce que nous évoquerons par la suite par le paramétrage d'applications et les développements spécifiques.

Fidèle à son champ d'investigation qu'est la gestion industrielle, l'APICS insiste sur des points fonctionnels :

« Un Système d'Information orienté finance pour identifier et planifier les ressources globales de l'entreprise nécessaires pour fabriquer, livrer, et facturer aux

clients. Un système d'ERP diffère d'un système typique MRP II suivant des impératifs techniques tels que l'interface utilisateur, la base de données relationnelle, l'utilisation d'un langage de développement évolué et d'outils d'ingénierie logicielle, l'architecture client/serveur, et la définition d'un système ouvert et portable. »

L'ERP est alors vu comme le moyen de rendre effectif la mise en œuvre d'une méthode MRP2, et d'assurer un réel contrôle financier. Il ne s'agit pas d'une remise en cause des fondements de la méthode, mais de se donner la possibilité technique de suivre le processus principal d'une entreprise orientée vers la satisfaction de ses clients.

2.2.2 Le système nerveux de l'entreprise

Si l'on admet une analogie avec le corps humain pris comme une entité où l'information circule entre les différentes parties grâce au sous-système nerveux, l'ERP occupe une place si importante dans le Système d'Information d'une entreprise qu'il n'est pas exagéré de l'assimiler à son système nerveux. L'intégration recherchée par l'exploitation de ce système lui confère, en effet, une lourde responsabilité de synchronisation de l'activité des services en conformité avec les processus opérationnels les plus importants. L'ERP touche donc des points essentiels du fonctionnement de l'entreprise, et cela ne va pas sans poser quelques problèmes (garantir l'alimentation en information entre les services), ou présenter certains risques (paralysie d'activité)...

Les chiffres tirés des enquêtes de terrain sont explicites : la décision d'utiliser un ERP est un véritable projet d'entreprise. Bien sûr, la mise en place d'un ERP est une opération longue (60 % des projets durent de 7 à 24 mois) et coûteuse (les dépenses représentent 5 à 10 % du chiffre d'affaires, le coût des licences ne représentant que 20 à 30 % des dépenses) (Robert Canonne et Jean-Louis Damret, 2002). Mais la satisfaction des utilisateurs de l'ERP est un critère de qualité qui fait quasiment l'unanimité parmi les managers. Dans une étude récente réalisée par (Ernst & Young, 2003) sur un échantillon de PME/PMI, 90 % des dirigeants ont plébiscité la satisfaction des utilisateurs, alors qu'un critère plus déterminant en termes d'efficacité de projet comme le ROI (Return On Investment) n'obtiendrait que 38 % des suffrages. Ce constat prouve que la décision d'acquiescer un ERP ne se fonde pas sur une volonté de réforme profonde des processus, des procédés ou des produits, mais se présente plus comme un acte de management où l'on souhaite d'abord que les acteurs continuent à animer les processus. Par analogie, nous dirons que le système nerveux continue de fonctionner, avec des gains induits en cascade. Les changements organisationnel et culturel occasionnés par la mise d'un nouveau Système d'Information font que les dirigeants souhaitent plus une bonne appropriation et adaptation du système par les utilisateurs, qu'intrinsèquement le bon fonctionnement du système lui-même (souhait plus implicite par rapport au choix de ce type d'outils).

Nous avons cherché à développer un peu l'analyse de ces sources de gain dans les deux enquêtes à notre disposition.

Dans les résultats de l'étude de (Ernst & Young, 2003), la décision d'implanter un ERP est motivée par l'optimisation de la performance (70 % des personnes interrogées) et l'accès plus rapide à l'information (54 %). Il s'agit de critères qui sont en droite ligne d'une concentration de l'attention sur l'activité du système nerveux par l'observation des processus horizontaux comme les nomme Jean-Louis Lequeux. Ces indicateurs se rapportent au résultat produit par le processus, résumé par le terme performance, et à une amélioration obtenue sur un facteur clé du déroulement de ce processus : la vitesse d'accès à la bonne information.

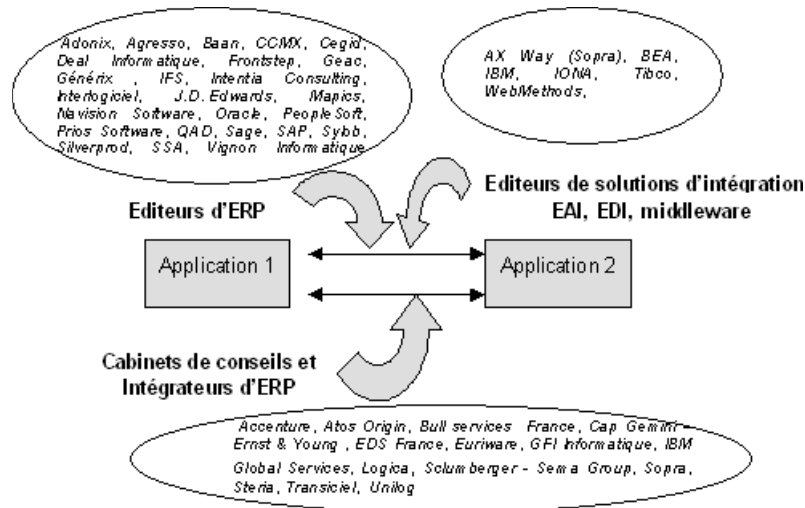


FIG. 2.4 – Principaux acteurs du marché des ERP

Dans l'enquête de nos collègues des arts et métiers, c'est la simplification des systèmes (60 %) et la restructuration de l'organisation (52 %) qui sont évoquées. L'opinion est plus directement dirigée vers l'analyse technique du système nerveux. Il est possible d'interpréter la simplification des systèmes comme la possibilité d'exercer un meilleur contrôle du fonctionnement des processus horizontaux. La restructuration de l'organisation fait état d'une opportunité de remettre en cause ces processus au moment où on les formalise (avant le déploiement de l'ERP), étape connue comme un exercice de réingénierie (Business Process Reengineering).

2.2.3 Le marché des ERP

Avant de décrire le marché actuel et futur des ERP, il nous semble utile d'introduire les différents types d'acteurs économiques qui contribuent à la dynamique de ce marché. Nous aurons l'occasion de revenir plus en détail sur la complémentarité de leur rôle dans la section consacrée à la gestion de projet ERP. Nous pouvons les répartir en trois classes :

- les éditeurs d'ERP, au sens de concepteur unique (SAP, Oracle, PeopleSoft...),
- les éditeurs de solutions d'intégration (BEA, IBM, WebMethods...),
- les cabinets de conseils et les intégrateurs (Accenture, Ernst & Young, Cap Gemini...).

2.2.3.1 Intégrateurs et Éditeurs d'ERP

« L'utilisateur et l'informaticien ne possédant pas, à eux deux, toute l'expertise requise pour implanter la solution informatique » (Jean-Louis Tomas, 1999), il est nécessaire de faire appel à des experts de la gestion de projet pour la mise en œuvre de l'ERP, en parallèle aux experts ayant une connaissance des outils. La différence entre ces deux groupes d'acteurs étant très mince, de plus, s'agissant généralement de services offerts par la même société, nous les regrouperons sous le nom d'intégrateur.

Les discours émanant des éditeurs varient entre deux pôles extrêmes. D'un côté, il y a ceux qui focalisent officiellement leur activité sur leur métier de base, l'édition, laissant, au moins dans le discours, à des tiers intégrateurs le soin d'apporter les prestations de services associées. On retrouvait initialement dans cette catégorie les éditeurs les plus importants du marché par le

	SAP (% 2001)	PeopleSoft (% 2001)
Vente Licence (1)	2 404 940 (-12 %)	530 077 (-18 %)
Maintenance	2 540 291	
Conseils	2 311 094	
Formation	433 979	
Service (2)	5 285 364 (+12 %)	1 411 341 (+3 %)
Autres revenus (3)	85 057 (-10 %)	7 530 (-93 %)
Total (1 + 2 + 3)	7 772 361	1 948 948

TAB. 2.2 – Revenus des grands éditeurs (en milliers de dollars)

chiffre d'affaires ou par la taille de la société (SAP, Oracle, Peoplesoft, Siebel...). De l'autre côté, il y a les éditeurs qui prônent une offre clé en main, l'intégration est alors comprise. Les leaders de l'édition ont d'ailleurs modifié leur approche en se positionnant dans le domaine des services par l'intermédiaire de filiales spécialisées (ex : les divisions Oracle Consulting, SAP systems integration...). On constate que les services en direct de ces éditeurs représentent aujourd'hui une part non négligeable de leur chiffre d'affaires (68 % du C.A. pour SAP et 72 % du C.A. pour PeopleSoft). Le tableau suivant (tab. 2.2) décompose le chiffre d'affaires des deux principaux éditeurs du marché des ERP. Nous nous apercevons que seuls les revenus liés aux services ont augmenté dans une économie des projets ERP morose.

Entre ces deux positions, il existe une grande variété de stratégies combinant plusieurs approches. En résumé, même si les deux métiers d'éditeur et d'intégrateur restent différents, la frontière entre les deux est de plus en plus floue. Cette frontière devrait, tout de même, continuer à exister pour les projets de grands comptes, évitant ainsi les parti pris pour la sélection de l'outil. Par contre, pour des sociétés telles que des PME/PMI, la question se pose un peu différemment, l'éditeur de la solution logicielle est souvent amené à jouer une partie du rôle d'intégrateur.

2.2.3.2 Situation du marché français des ERP

En 2001, l'ensemble des revenus des éditeurs d'ERP a atteint la somme de 1 011 millions d'euros sur le marché français. Ce marché, qui était en pleine expansion dans les années 95-2000 (+80 % - source IDC), subit un net ralentissement. La croissance du marché total, pour l'année 2002, s'élève seulement à 12,9 %, confirmant la tendance amorcée l'année précédente d'une faible dynamique des nouveaux projets.

Ce ralentissement peut s'expliquer par les éléments suivants :

- le passage à l'an 2000, puis le passage à l'euro, ont créé une dynamique de nouveaux projets à la fin des années 1990, par un effet de psychose collective impactant très positivement des projets ERP, et ces effets s'estompent maintenant,
- la saturation du marché sur les couvertures traditionnelles des ERP, principalement chez les grands comptes, a réduit les cessions de licence. (Robert Canonne et Jean-Louis Damret, 2002) indiquent que la durée de vie d'un ERP est en moyenne de 5 à 10 ans (50 %). Ce délai de renouvellement serait trop faible, d'après certains éditeurs, si on le compare au « turn-over » des outils de bureautique. Quoi qu'il en soit, la première vague est presque finie, et la seconde n'a pas encore débuté,
- la situation économique des entreprises, et principalement des PME/PMI plus sensibles aux marchés financiers, a ralenti le rythme des investissements. Les entreprises préfèrent

	2002 (<i>source PAC^d</i>)	2001 (<i>source IDC</i>)	Nbre Clients (2001) (France/Monde) (<i>source CXP</i>)
SAP	44,5 %	29,7 %	1 000 / 25 000
PeopleSoft	12,2 %	10,7 %	600 / 11 000
Oracle	8,6 %	10,6 %	350 / 8 500
Sage	-	8,4 %	NC / 3 800
Intentia	7,8 %	6,9 %	300 / NC
Cegid	-	7,5 %	450 / NC
JD Edwards	4,4 %	4,1 %	300 / 4 000
Adonix	4,1 %	3,0 %	170 / 250
Viveo	-	2,7 %	1 200 / NC
Générix	2 %	1,9 %	200 / NC
Autres	16,4 %	14,5 %	NC

TAB. 2.3 – Parts de marché des éditeurs en France

engager leurs ressources financières dans des projets plus courts et plus rapidement fonctionnels.

Le tableau 2.3 décrit les répartitions de marché entre les principaux éditeurs par rapport aux revenus de la vente de licences. Pour comprendre la dynamique du marché, il est nécessaire de prendre également en compte la stratégie de fusion/acquisition entre ces éditeurs. Par exemple, en 2002, Microsoft, éditeur absent du marché des « solutions d'entreprise », se lance dans la compétition. Pour cela, il rachète Navision pour la somme de 1,45 milliards d'euros². Cet achat est la finalisation d'une relation de deux ans entre les deux acteurs (accord de partenariat sur la commercialisation et distribution des produits Microsoft). Plus récemment, en 2003, PeopleSoft (n°2) rachète JD Edwards (n°5) pour 1,68 milliards de dollars tandis que Oracle (n°3) lance une OPA hostile sur PeopleSoft, avec ou sans JD Edwards.

Nous pouvons comparer le comportement de ce marché à celui de l'automobile avec quelques années de décalage. Rappelons-nous les relations de Renault avec Nissan, Volkswagen avec les composantes de son groupe (Skoda, Seat, Audi, Bentley et Lamborghini) et encore bien d'autres, BMW, General Motors... Les éditeurs de solutions logicielles de gestion sont maintenant entrés dans cette logique de fusion/acquisition qui est une réponse stratégique à une réduction rapide de la demande après une période d'expansion forte du marché. Ce phénomène de cannibalisation des éditeurs a des origines diverses : marketing, commerciale ou technologique.

Mais l'offre des grands éditeurs n'est pas encore prédominante sur tous les secteurs. Il y a une offre de petits éditeurs sur le secteur des PME/PMI, qui existent depuis longtemps et y sont bien implantés. Les difficultés rencontrées par des grands éditeurs sur leurs plus gros clients font qu'ils s'intéressent depuis peu à des entreprises de taille plus petite, espérant ainsi maintenir leur croissance et préserver leur profit. La compétition a donc pris une nouvelle intensité, non seulement entre les grands, mais également entre les grands et les petits éditeurs (Bernard Laur, 2001).

Cette offre diversifiée limite la capacité d'observation des jeux d'acteurs car beaucoup d'éditeurs, qui revendiquent la classification ERP pour leur produit, n'ont en réalité que des produits plus limités qu'il est difficile de placer au même niveau que SAP, Oracle Applications ou

²<http://www.microsoft.com/france/infos/presse> - 07/05/2002

JDE. Dans ses statistiques, le CXP a l'habitude de citer une trentaine de références d'éditeurs français d'ERP. De plus, certaines sociétés, qui disent vouloir implanter un ERP, sont tout à fait satisfaites de ces solutions « dégradées ».

2.2.3.3 Situation par secteurs d'activité clients

En 2001, l'industrie de production de biens est restée, comme les années précédentes, le principal client des éditeurs de progiciels ERP en absorbant plus de la moitié du marché des revenus licences, tandis que pour l'année 2002 ce secteur a subi un net recul de 18,6 %. En dépit du ralentissement de l'économie française, les entreprises industrielles, principalement l'industrie pharmaceutique, ont maintenu leurs investissements en solutions ERP à un rythme plus élevé que celui du marché.

La dynamique des entreprises du secteur des services, après avoir montré des signes de frémissement depuis ces deux dernières années, a enregistré des tendances différenciées :

- durement touchée en 2001 par l'éclatement de la bulle spéculative, l'industrie des télécom a vu sa capacité d'investissement réduite, limitant d'autant les achats en solutions ERP par les entreprises de ce secteur,
- les entreprises financières sont demeurées également frileuses au cours de l'année 2001, maintenant certes leurs achats en solutions ERP, mais à un rythme moins élevé que celui du marché,
- à l'inverse, les entreprises du secteur des services aux entreprises, ainsi que celles du commerce de gros et détail, ont massivement investi en 2001 en achat de solutions ERP (source IDC).

2.2.3.4 Situation par taille d'entreprise cliente

Selon IDC, le nombre de licences (tous secteurs confondus) vendues aux PME françaises de moins de 100 salariés a augmenté de 77 % en 2001. Le marché des ERP a connu une progression sensible (11,8 %) des ventes sur le segment des entreprises de moins de 100 salariés, grâce à l'effort entrepris par certains éditeurs pour adapter leur offre à ce créneau.

Taille de l'entreprise	Part de marché
moins de 100 salariés	10 %
de 100 à 500	15 %
de 500 à 1000	22 %
de 1000 à 2000	25 %
plus de 2000	28 %

TAB. 2.4 – Part de marché par taille de l'entreprise (Vente licences)

Ancré à l'origine sur le segment des grands comptes, le marché de l'ERP se démocratise. Les entreprises des franges basses du marché sont massivement entrées dans l'ère de l'ERP en 2001, du fait de la disponibilité d'offres intégrées adaptées à leurs tailles et délivrées par des éditeurs. La demande des PME, comme celle des grands comptes, est principalement tirée par l'évolution des technologies et du contexte économique. Ces entreprises ont besoin des mêmes composants que les grands comptes pour rationaliser les coûts et assurer la cohérence dans le déroulement d'un processus. Les principaux modules installés sont d'ordre fonctionnels

(gestion des achats, des commandes, suivi de production), par contre les PME sont à l'écoute et envisagent le déploiement de modules plus e-business (chaîne logistique, CRM, B2B).

Avec la complexification croissante de leurs Systèmes d'Information, les entreprises de cette frange sont aujourd'hui de plus en plus réceptives à l'argument autour d'une solution intégrée. Pour atteindre ce marché, les gros éditeurs tentent de réduire les coûts liés au projet. Pour cela, ils proposent de plus en plus de solutions pré-configurées, déployables plus rapidement, dédiées à un secteur particulier (textile, automobile, agro-alimentaire...).

2.2.3.5 Nouvelle tendance du marché avec l'Open-Source

Les logiciels libres et l'open-source trouvent une place croissante dans les Systèmes d'Information des PME/PMI : les plus petites structures l'adoptent, notamment pour des raisons économiques. En effet, le modèle économique sous-jacent fait moins peur et présente de nombreux atouts : souplesse du développement, fiabilité, capacité de personnalisation des applicatifs, gratuité pour les logiciels libres ou faible prix pour les open-source.

D'après IDC, Linux pénétrait plus de 32 % des entreprises françaises en juillet 2000. Le taux de pénétration des PME/PMI de 100 à 500 salariés était de 17 %.

Si le logiciel libre/open-source était réservé jusqu'à présent aux couches « basses » du logiciel (systèmes d'exploitation, logiciels techniques, bases de données...), il gagne désormais du terrain sur les hauteurs du fonctionnel. L'offre de logiciels libres dans les domaines applicatifs de la gestion de production, des commandes, de la relation clients et des ressources humaines commence à se développer de manière notable.

Plus de mille offres sont d'ores et déjà disponibles sur plates-formes Linux, dont les PGI des plus grands éditeurs : SAP, PeopleSoft, Oracle... Rien n'empêche de faire cohabiter progiciels et logiciels libres au sein du même Système d'Information.

Le premier PGI « full open-source », ERP5¹, a été présenté à l'occasion du salon Linux solutions 2003. Un premier déploiement industriel d'ERP5 a été réalisé au sein d'une PMI de 250 personnes. ERP5 est plutôt un progiciel sectoriel qu'un PGI générique. On considère qu'un ERP est sectoriel lorsqu'il est spécifiquement dédié à un secteur d'activité. Dans le cas d'ERP5, il est proposé pour une PME/PMI travaillant dans le secteur du textile.

De plus, il ne s'agit pas du seul produit de ce type disponible. Il existe des produits sommaires, bas de gamme, intéressants pour des TPE (Très Petites Entreprises), comme GNUCash², assurant la comptabilité, SQL Ledger³, proposant une comptabilité et une gestion commerciale. Mais, des produits de moyennes gammes, ayant des fonctionnalités plus élaborées, sont aussi disponibles comme Nola⁴, Compierre⁵...

Cette innovation technologique risque de bouleverser les données du marché. Les premiers éléments de changements deviennent de plus en plus perceptibles. Nous pouvons citer par exemple :

- le cas SAP : après avoir décidé de fournir gratuitement sa base de données SAPdb, on nous annonce la fusion entre la base de SAP et le produit MySQL. L'intérêt de cette fusion est d'utiliser la popularité de MySQL dans le monde libre et d'avoir la performance d'une base de données entreprise avec SAPDB ;

¹<http://www.erp5.org>

²<http://webaccountant.sourceforge.net>

³<http://www.sql-ledger.org>

⁴<http://nola.noguska.com>

⁵<http://www.compiere.org/>

- le cas de SUN : cette société a mis depuis quelques années sa suite bureautique en open-source. Avec l'avènement d'Internet et la facilité de téléchargement, elle réalise environ 300 000 téléchargements de l'application par mois (cela comprend la mise à jour et la première installation). De plus, on peut recenser 15 000 développeurs potentiels (statistique de téléchargement des fichiers sources) pour faire évoluer cette application.

Les entreprises sont tentées de plus en plus de réduire les coûts de licences et les coûts liés à l'évolution d'une solution. Si des éditeurs proposent des outils, systèmes de gestion de données ou suite bureautique, disponible en open-source et proposant les mêmes fonctionnalités, alors les éditeurs d'ERP devront prendre en compte cette tendance et faire évoluer leurs produits pour proposer de nouvelles possibilités afin de se connecter à ces nouveaux systèmes.

Chapitre 3

Mutation du paysage industriel des PME/PMI

3.1 Les PME/PMI en France

On dénombre environ 2 413 000 entreprises de moins de 500 salariés en France. Plus de la moitié de ces entreprises n'ont pas d'employés, sont en sommeil, où correspondent à des « holding » ou SCI (Société Immobilière). La répartition des entreprises restantes, suivant leur taille, est la suivante¹ :

	Nbre d'entreprises	avec TPE	sans TPE
de 1 à 9 salariés	1 040 000	86,5 %	-
de 10 à 199 salariés	154 000	13 %	97,4 %
de 200 à 499 salariés	4000	0,5 %	2,6 %

TAB. 3.1 – Répartition des PME/PMI par leur taille

Pour le reste, 2 000 entreprises sont considérées comme des grands comptes avec plus de 500 salariés).

3.1.1 Situation économique générale des PME/PMI

3.1.1.1 Situation 2001

En 2001, les résultats des PME/PMI françaises demeuraient positifs, en dépit d'une économie mondiale fragilisée par un net recul. 56 % des entreprises augmentaient leur chiffre d'affaires, alors que 16 % connaissaient une baisse. Notamment, les PME de petite taille amélioraient leurs résultats, alors que les entités plus grandes (50 à 199 salariés) étaient en léger retrait.

3.1.1.2 Situation 2002

Pour l'année 2002, les dirigeants auraient noté un recul de 2 % de l'investissement dans l'ensemble de l'industrie, et de 3 % dans l'industrie manufacturière. La situation est contrastée

¹source INSEE 2001

suyant les secteurs : dans les secteurs des biens d'équipement et surtout de biens de consommation, la croissance serait égale à 3 ou 4 %, alors que le secteur des biens intermédiaires connaîtrait un recul de 8 %.

3.1.1.3 Prévisions 2003 / 2004

D'après une enquête réalisée par BNP Paribas Group, 39 % des dirigeants de PME/PMI envisagent une amélioration et 15 % prévoient plutôt une baisse pour 2003. Cette relative confiance n'aura pas d'effet positif sur l'investissement global, qui devrait rester stable par rapport à 2002. La rigueur que les dirigeants se sont imposés l'an passé devrait par contre leur permettre d'accroître leur rentabilité, ce qui peut présager une reprise des investissements pour 2004.

3.1.1.4 Taux d'investissement informatique

Dans cette conjoncture, les entreprises ont diminué leurs investissements informatiques pour assurer la profitabilité de leur entreprise. Dans ce contexte, le recul de 5,2 % des ventes de licences¹ d'ERP, sur le marché français témoigne d'une certaine logique de prudence des entreprises. La seule catégorie d'entreprise à enregistrer une dynamique positive (0,4 %) des projets ERP est la catégorie haute des PME/PMI (entreprises de 100 à 499 personnes). Cette croissance n'a pas réussi à compenser la baisse des projets dans les autres catégories.

Du côté des éditeurs d'ERP, les principaux leaders du marché ont pu résister à ce déclin des investissements informatiques. Mais, leur dynamique de croissance du chiffre d'affaires s'est ralentie par rapport aux années précédentes. Les petits éditeurs ont eu plus de mal à résister à cette baisse de projets informatiques. Leurs activités ont subi un recul de 2,5 à 5 fois supérieur au recul enregistré par le marché.

Pour compenser ces différentes baisses, les éditeurs se sont rabattus sur leur base installée permettant de stabiliser le recul des revenus de ventes de licences.

	CA 2001		CA 2002		Croissance 2001/2002
	Valeur	Poids	Valeur	Poids	
Base installée	240,6	62 %	245,8	67 %	2,2 %
Nouveaux Projets	146,3	38 %	121,1	33 %	-17,2 %
Total	386,9	100 %	366,9	100 %	-5,2 %

TAB. 3.2 – Dynamique des projets sur le marché français des éditeurs d'ERP (millions d'euros)

3.1.2 Niveau d'informatisation des PME/PMI

En 2000 (source IDC), le taux de pénétration des ERP au sein des entreprises françaises était de 31,1 %. Pour l'année 2002, une étude de 01 Informatique indique que 60 % de PME françaises sont équipées d'un progiciel. Cet échantillon est représentatif de l'ensemble des entreprises de 50 à 499 salariés classées en quatre secteurs : industrie manufacturière, industrie de process, administration, services et transport. En outre, 18 % d'entre elles ont actuellement

¹http://www.idc.fr/presse/cp_erp2003.htm

un projet d'installation en cours. À court terme, les trois quarts des PME françaises seront donc équipées d'un véritable système informatique intégré, facilitant la gestion quotidienne de leur comptabilité, de leur action commerciale ou de leur production.

Selon IDC, le marché des PME/PMI, malgré une saturation des possibilités de déploiement, intéresse toujours les éditeurs d'ERP. Cette tendance s'explique par les trois points suivants :

- l'extension des solutions déjà déployées par ces « dits » éditeurs : cette extension consiste à rajouter un ou plusieurs nouveaux modules fonctionnels ;
- le renouvellement de solutions non pérennes : lors d'un premier déploiement, souvent, les PME/PMI font l'acquisition de produits simplistes usurpant le nom d'Enterprise Resource Planning. Après avoir atteint les limites de leurs outils et dans le cadre du renouvellement de leurs Système d'Information, elles font plus appel à des éditeurs plus solides du marché ;
- la mise en œuvre de nouvelle architecture telle qu'Internet : la mise en œuvre de nouvelle architecture, pour répondre aux besoins de leur client, va favoriser la mise en place de nouveaux projets.

Nous pouvons justifier le deuxième point avec l'étude menée par Robert Canonne et Jean-Louis Damret. Dans cette étude, la durée de vie des ERP au sein des organisations a été estimée et les résultats obtenus sont présentés dans le tableau 3.3.

Durée de vie estimée du système ERP	Fréq.
moins de 3 ans	2,15 %
de 3 à 5 ans	17,20 %
de 5 à 7 ans	23,66 %
de 7 à 10 ans	25,81 %
plus de 10 ans	31,18 %

TAB. 3.3 – Durée de vie estimée d'un ERP

Dans plus de 80 % des cas, les entreprises estiment la durée de vie de leur système à plus de 5 ans. Il aurait été intéressant de croiser ces chiffres avec la taille de l'entreprise pour voir s'il existe une corrélation entre la durée de vie de l'ERP et les sommes engagées.

3.1.3 Tendances du marché à long terme

Une étude du groupe AMR Research¹ propose une évolution du marché européen des applications d'entreprise. Cette nouvelle tendance est caractérisée par les chiffres suivants :

- le marché européen sera de 21,3 milliards de dollars en 2006 pour un marché total avoisinant 70 milliards de dollars,
- le marché des ERP (Enterprise Resource Planning) subira une baisse des revenus (de 6,1 milliards de dollars en 2002 à 6,5 milliards de dollars en 2006). Cette baisse est principalement due à l'extension des modules du SI (Système d'Information) avec les outils tels que les SCM (Supply Chain Management) et GRC (Gestion de la Relation Client)
- le marché des SCM (Supply Chain Management) passera de 1,5 milliards de dollars en 2002 à 3,3 milliards de dollars en 2006.

¹<http://www.amrresearch.com/> - Analytix Report 2001-2006

- le marché des GRC (Gestion de la Relation Client) augmentera de 27 % pour approcher 8,7 milliards de dollars en 2006.

Cette analyse est seulement limitée à l'Europe. Nous pouvons quand même considérer que l'augmentation du marché sera dans les mêmes proportions pour les autres continents.

3.2 De nouvelles formes de partenariat entre entreprises

Les entreprises, aussi bien les grands comptes que les PME/PMI, restent et demeurent un milieu soumis à de fortes contraintes (AFNET, 2002) :

- celles, externes, du client, du fournisseur, du concurrent, du marché, de la société (qualité de service, lois, normes, impôts...),
- mais aussi celles, internes, des salariés et des structures organisationnelles mises en place pour la faire fonctionner.

Dans un contexte de mondialisation, les dirigeants des PME/PMI doivent faire évoluer et modifier leurs organisations, utiliser toutes les ressources disponibles (la mise en place d'un SI par exemple), coopérer et collaborer avec des partenaires (quelque fois concurrents) afin de se positionner sur des projets qui seraient inaccessibles pour une seule entité.

On demande donc à des entreprises d'être plus réactives, de s'adapter plus rapidement aux nouvelles lois et décrets sur le travail (passage au 35 heures par exemple), de mettre en place des conventions avec des partenaires possédant le savoir-faire, aussi bien des entreprises privées que des institutions publiques. Ces différentes évolutions doivent être justifiées et corrélées avec les informations financières de l'entreprise.

Un des éléments de réponse est la mise en place d'un Système d'Information adapté à cette nouvelle forme d'organisation. Le SI permettra aux dirigeants de prendre des décisions justes et proportionnées, corrélées aux informations fournies. Ces informations pourront et devront être partagées avec ces nouveaux partenaires.

Quel que soit le Système d'Information déployé, il doit donc être capable de s'adapter, se paramétrer suivant les différentes stratégies de partenariat des entreprises. Nous distinguerons deux genres d'évolution organisationnelle dans les entreprises :

- les évolutions internes : l'entreprise souhaite modifier son mode de gestion,
- les évolutions externes : l'entreprise souhaite collaborer, d'une manière plus ou moins forte, avec des partenaires.

3.2.1 Évolution organisationnelle interne

Le Système d'Information doit être capable de s'adapter à l'organisation interne de l'entreprise. Il peut s'agir de prendre en compte des contraintes physiques (par exemple la gestion d'une entreprise en multi-sites) ou de contraintes organisationnelles (par exemple la modification de son mode de gestion de stock pour un produit : passage du mode point de commande au mode sur stock).

Mais si l'évolution organisationnelle interne se concrétise pour partie autour d'un processus qui est orienté vers les clients, les clients ont des habitudes qui changent. En particulier, les clients, et c'est nouveau, sont présents sur Internet. Ils souhaitent développer des relations commerciales, de soutien logistique... à travers ce nouveau média. Il y a nécessité de s'adapter en permanence à ces nouvelles formes de sollicitation où l'information prend une place prépondérante.

3.2.2 Évolution organisationnelle externe

L'évolution des relations contractuelles entre les entreprises (donneurs d'ordre/fournisseurs, entreprise en partenariat...) veut qu'il y ait de plus en plus de changements organisationnels aux frontières des entreprises. Ces entreprises s'orientent de plus en plus vers des nouvelles formes d'organisations multipartenaires. Ces formes de collaboration, de partenariat et de coopération (Matthieu Lauras et al., 2003) nécessitent des échanges d'information accrus entre des systèmes différents. Les relations entre les entreprises et avec des entités extérieures n'ayant pas forcément un caractère stable ou pérenne, les solutions informatiques se doivent d'être simples, rapides et très adaptables.

Nous distinguons plusieurs types de relations :

- les relations entre des clients (incluant des donneurs d'ordres) et des fournisseurs,
- les relations avec les partenaires (financiers, organismes institutionnels),
- les relations des entreprises en réseaux.

Le premier type de relation (donneurs d'ordre/fournisseur) fait apparaître principalement un besoin d'échanges d'informations. Le fournisseur va chercher les informations correspondant aux futurs besoins (par exemple : les commandes) de son client. Il s'engage sur la prestation et enclenche le règlement. Le donneur d'ordre gère le carnet de ses commandes et surveille la qualité des services. Le moyen de communication entre le donneur d'ordre et le fournisseur doit être fédérateur et reconnu par les deux parties. Il y a souvent un effort commun pour établir le processus de communication inter-organisationnelle. Dans ce type de relation, nous pouvons aussi placer les relations existantes entre les entreprises au sein de la chaîne logistique. Cette relation demande plus que de l'échange d'informations. Dans ses travaux, Matthieu Lauras et al. développe un processus de coopération en logistique. Ce processus identifie les différentes activités par rapport à trois horizons (Matthieu Lauras et al., 2003) :

- à long terme, l'amélioration de la profitabilité globale par l'optimisation des processus,
- à moyen terme, la réduction des coûts par la rationalisation des flux,
- à court terme, la garantie de la qualité de service par la synchronisation des opérations.

Nous nous apercevons que la planification de la chaîne logistique dépasse le simple cadre d'échange d'informations pour aller jusqu'au partage de ressources, optimisation et synchronisation des processus entre les entreprises et le suivi des flux financiers, des ressources et informationnels.

Dans le deuxième type de relation, il y a plusieurs catégories de partenaires : banques, institutions organisationnelles (services de l'état, de la région...) ou métiers (corporations...). Ces partenaires sont des consommateurs d'informations sur l'entreprise. Il y a peu ou pas de normalisation des relations, et le besoin est surtout au niveau de l'accès aux informations. Il peut s'agir de consulter le bilan prévisionnel sur la situation économique de l'entreprise, d'une situation par rapport à un marché financier (bourses).

Dans le dernier type de relation (réseaux d'entreprise), nous entrons véritablement dans un concept de coopération et de collaboration entre des entreprises. Il peut s'agir, par exemple, d'un travail collaboratif en conception ou en exploitation afin de mettre en commun les ressources de chacun. Cette relation nécessite la mise en place de contrat de partenariat entre les entreprises participantes. Ce contrat stipule les activités de chacun mais aussi les moyens mis à la disposition des partenaires, par exemple le Système d'Information de l'entreprise. Les relations entre les différents Systèmes d'Information ne sont pas au simple niveau de l'échange d'informations. Cette fois, il est nécessaire d'utiliser des fonctions d'un autre système afin d'optimiser sa propre gestion. Nous pouvons prendre comme exemple un calcul de besoins

dépendant du calcul de besoins de son fournisseur ou du corps de métiers intervenant sur la même affaire. Dans le BTP, cas d'étude du projet (Patrick Burlat, 2001), un ensemble d'entreprises doit pouvoir mettre en commun ses savoirs, ses disponibilités afin de répondre ensemble aux appels d'offres qu'il ne pourrait pas couvrir seul.

3.2.3 Synthèse

L'appropriation d'un Système d'Information par les grandes entreprises génère des besoins vis-à-vis de leurs partenaires. Les grandes entreprises ont, dans un souci de mise en place d'une démarche qualité ou de mise en œuvre d'un Système d'Information, optimisé leurs processus internes, mais les entreprises sont restées plus au moins closes vis-à-vis de l'extérieur. Le concept d'entreprise étendue est alors apparu afin de faire intervenir dans les processus organisationnels les entreprises partenaires. Ce système brise les frontières de l'entreprise (principalement des donneurs d'ordres) et offre la possibilité de mettre en place une gestion de la chaîne logistique avec des outils pour supporter cette nouvelle vision (par exemple le SCP (Supply Chain Planner)). Les clients et les fournisseurs doivent alors adhérer à ce nouveau cadre relationnel pour une gestion articulée autour des flux des entreprises. Cette nouvelle gestion oblige les dirigeants de PME/PMI d'adapter leurs organisations par rapport à ces nouveaux besoins afin de proposer les services adéquats.

Ces évolutions organisationnelles entraînent la mise en œuvre de nouveaux besoins et de nouveaux services. Dans le cas d'une évolution interne, nous nous rendons compte qu'il est nécessaire de :

- mettre en œuvre un SI adaptable à l'évolution de l'organisation,
- fournir de nouveaux services aux clients,
- évoluer par rapport aux nouvelles technologies (aujourd'hui Internet, demain, ...).

Dans le cas d'une évolution externe, nous retrouvons les éléments suivants :

- mise en œuvre de services (ou d'interface) pour l'extérieur. Ces interfaces permettent d'accéder aux informations provenant d'un Système d'Information d'une entreprise distante,
- partage des traitements entre Systèmes d'Information,
- mise en place de procédure, de contrat entre les différents partenaires, définissant le rôle et les activités de chacun.

Il est clair qu'un besoin se dégage de ces différentes évolutions, il s'agit de la mise en œuvre d'un Système d'Information adaptable. Il est important que les entreprises n'attendent pas cinq années (tab. 3.3) avant de renouveler leur système, mais qu'elles puissent le faire suivant l'évolution de leurs organisations.

Enfin, un des besoins identifiés est dans l'utilisation en commun d'un SI pour suivre et piloter des activités dans le cadre de travail collaboratif ou coopératif. Des SI prenant en compte ces relations entre partenaires restent à définir (emplacement physique de l'information, condition d'accès au SI)

Par rapport à ces différents concepts, il est clair que les Systèmes d'Information doivent évoluer afin de tendre vers ces nouvelles organisations. Ce n'est pas à l'entreprise de se réorganiser pour correspondre au système mais au système de s'adapter à l'organisation et aux besoins de celle-ci.

3.3 Une pression forte sur la maîtrise des coûts

Parce que la concurrence sur les prix est forte et que les marchés sont de moins en moins captifs, la maîtrise des coûts de revient industriels est une obligation permanente pour l'entreprise. Dans la relation d'une PME/PMI à son donneur d'ordre, par exemple, les marges du fournisseur sont souvent très petites. Le jeu des réponses à enchères et des appels d'offre obligent les sous-traitants à prendre des risques et à jouer sur les limites de rentabilité. Le moindre écart par rapport aux estimations est une source potentielle de perte. Mais, dans la suite, le passage à l'acte soulève des difficultés. En effet, la comptabilité analytique et le contrôle de gestion sont des fonctions dont l'exercice est complètement dépendant de l'accès à la réalité opérationnelle de l'entreprise, via l'information. Or, il est utopique de considérer que le Système d'Information contient toute l'information de l'entreprise. C'est souvent un point limitant pour l'analyse comptable.

Parmi les éléments qui composent le coût de revient d'un produit, on est amené à chiffrer des coûts d'amortissement des matériels, et des coûts de fonctionnement directs et indirects. Dans tous les cas, ces coûts sont liés à des ressources. L'ERP prône une vision globale des ressources de l'entreprise. Il est donc à même de faciliter la structuration de l'entreprise dans la maîtrise de ces coûts, et les modules de comptabilité sont conçus pour permettre des analyses périodiques selon plusieurs axes et à des niveaux différents de profondeur. Ils produisent, à échéance régulière ou sur demande circonstanciée, des rapports (états) sur le suivi des coûts. Si elle est bien préparée, la tâche de l'analyste est facilitée. Il passe d'une situation avant ERP où l'essentiel de son activité était la collecte et la synthèse, à une situation post ERP où il produit la synthèse et cherche des préconisations d'amélioration afin de maintenir les équilibres prévus.

Nous avons eu l'occasion de constater, à de multiples occasions, que les entreprises de petite taille (et même de grosse taille) ne se sont pas nécessairement dotées des structures pour assurer cette maîtrise. Il en va pourtant de la survie de la plupart d'entre elles. C'est un atout dans la décision d'implanter un ERP que de pouvoir concentrer l'attention sur les coûts et de réfléchir aux moyens de surveiller avec réalisme leur évolution.

3.4 La relation d'une PME/PMI à son Système d'Information

D'une manière générale, la relation entre une entreprise et son Système d'Information est une relation très forte. L'entreprise doit se fixer des axes de progrès et des outils de contrôle pour vérifier que les objectifs ont été réellement atteints. Le SI (Système d'Information) permet de supporter les différents processus opérationnels de production, de gestion, de pilotage et de contrôle représentant une partie des processus permettant d'atteindre ces objectifs. Le SI assure la persistance des informations manipulées par les processus de l'entreprise, ces informations représentant la mémoire de l'entreprise. Le SI touche donc au cœur de métier de l'entreprise.

Avant de détailler plus précisément cette relation au sein d'une PME/PMI, il est important de rappeler qu'il ne faut pas confondre Système d'Information et système informatique. La PME/PMI a surtout une culture du système informatique. Celui-ci est vu comme un moyen, et non comme une fin. Il y a une espèce de conviction que les progrès en la matière se font naturellement sous la pression des technologies, avec l'arrivée de nouveaux matériels et de nouvelles versions de logiciel. La culture du Système d'Information oblige à accepter un rôle stratégique au système, et à l'identifier réellement comme un moyen de manager différemment.

La mise en place d'un Système d'Information peut modifier le fonctionnement de la PME. Le SI peut :

- influencer sa stratégie :
 - le rôle stratégique du Système d'Information apparaît progressivement et par sollicitation externe sous la pression des nouvelles formes d'organisation qui demandent des moyens de communication toujours plus modernes : se connecter à ces réseaux implique de n'être pas trop à la traîne. Beaucoup d'entreprises le découvrent avec étonnement, l'apprennent parfois à leur dépend et adoptent souvent une attitude plutôt défensive ;
 - l'adéquation entre les besoins de l'entreprise et les fonctionnalités proposées par l'éditeur peuvent être au bénéfice de l'entreprise. Cette adéquation peut modifier l'organisation de l'entreprise, afin d'utiliser les meilleures pratiques, généralement imposées par le système. Les nouvelles fonctionnalités de l'outil peuvent aussi apporter des gains par un support des relations entre les différents partenaires (clients/fournisseurs...);
- modifier son organisation :
 - les canons de la décision hiérarchisée en plusieurs niveaux ne sont pas réellement applicables à ce type d'entreprise. Le niveau opérationnel est sur-dimensionné par rapport aux deux autres niveaux. Ceci conduit naturellement à un comportement plutôt axé sur la capacité à réagir que sur une organisation préprogrammée. Mais les évolutions contextuelles mettent à rude épreuve ces capacités à réagir en les poussant à l'extrême ;
 - Les organisations ne sont pas dotées de direction spécialisée sur le Système d'Information, c'est souvent une fonction remplie par un cadre à qui l'on confie d'autres fonctions en parallèle. Cela coupe l'entreprise de la veille technologique dans le domaine, et il est difficile de bâtir des plans directeurs ;
- améliorer sa performance :
 - les nouvelles formes de négociation commerciale avec les clients induisent une maîtrise poussée des coûts de revient, à se pencher sur la chaîne de valeur et à avoir un réel contrôle de gestion. Les nouvelles certifications qualité incitent également à prendre ce chemin. Peu à peu, les concepts de processus horizontal et d'intégration des fonctions métier prennent du sens pour les dirigeants d'entreprise. Le Système d'Information pourra trouver toute sa légitimité grâce à cette ouverture,
- entraîner des coûts financiers :
 - les perspectives économiques sont toujours empruntées d'une plus grande incertitude pour cette catégorie d'entreprises. La capacité à mobiliser des fonds pour investir sur un projet ERP est limitée. On rencontre alors un problème délicat de recherche d'équilibre entre vouloir et pouvoir. Les mécanismes de décision sont sous la responsabilité d'un faible nombre de personnes qui sont déjà surchargées par les activités quotidiennes. Le recours à des conseils n'est pas habituel, et pas forcément très bien perçu ;
 - beaucoup d'entreprises développent un complexe lorsqu'elles sont confrontées à des problèmes résolus par des grandes entreprises, surtout s'il s'agit d'un client. Ce sont des systèmes trop compliqués, pas utiles, peu abordables. Pourtant, en matière de Système d'Information, il existe des solutions (comme l'open-source) qui lèvent des barrières financières et techniques. Il y a une prise de conscience des collectivités publiques de l'importance du sujet, et des aides pour accéder aux TIC.

Aujourd'hui, l'offre du marché est relativement peu lisible, les salons ERP sont des foires aux sigles et aux dernières modes, dans lesquelles le dirigeant d'entreprise se sent vite dépassé. À qui peut-il expliquer son métier, ses domaines d'excellence, les problèmes qu'il rencontre ? Qui peut comprendre ses besoins et comment garantir qu'il cherchera une solution appropriée ?

La mise en place d'une gestion de projet (§ 4) pour la mise en œuvre d'un progiciel doit apporter des éléments de réponse et doit garantir une satisfaction du client vis-à-vis de la solution déployée.

3.5 Quels ERP pour ce type d'entreprise ?

Ce type d'entreprise, la PME/PMI, a les mêmes besoins dans le déploiement des modules fonctionnels représentant le cœur de l'ERP (gestion des commandes, gestion des achats, gestion de production par exemple) que les grands comptes. Par contre, les PME/PMI n'ont pas les mêmes moyens financiers pour déployer les solutions logicielles des gros éditeurs dans l'état actuel. Ces moyens financiers sont limités et brident le travail à réaliser sur le projet de déploiement. Les éditeurs font évoluer de plus en plus leurs offres commerciales pour conquérir cette nouvelle cible.

Les offres du marché doivent, du côté des gros éditeurs, évoluer pour atteindre ce marché ou doivent s'améliorer pour les éditeurs dont il s'agit déjà de leur cible privilégiée. Les différents axes d'amélioration, pour les ERP dédiés à ce type d'entreprise, sont :

- architecture logicielle,
- module vertical,
- module transversal.

Nous détaillerons dans la partie suivante (Partie II) les différentes évolutions de l'architecture logicielle liées, principalement, à l'émergence de nouvelles technologies et à l'identification de nouveaux besoins de la part des utilisateurs.

En ce qui concerne le déploiement de modules, il est important d'identifier les différentes catégories de PME/PMI ciblées. Nous nous situons dans un marché très vaste avec des entreprises de tailles différentes et de secteurs d'activités variés. Il est donc nécessaire d'identifier différentes catégories de PME/PMI ayant une solution logicielle et une gestion de projet dédiée. Dans le cas des TPE ou PME (de 10 à 49 personnes par exemple), nous serions plus dans la configuration suivante :

- déterminer les modules qui vont être utilisés,
- proposer une solution pré-configurée,
- adapter l'organisation de l'entreprise pour supporter les meilleures pratiques mises en œuvre par les processus de la solution logicielle.

Pour les PME/PMI (de taille moyenne, de 49 à 199 personnes), l'expression et l'analyse des besoins jouent un rôle plus important dans le projet et dans l'adéquation de la solution avec l'organisation de l'entreprise par rapport au cas précédent. Mais dans l'objectif de réduire la durée du projet, donc les coûts globaux de mise en œuvre, il est préférable de déployer une solution pré-configurée. Après avoir identifié les différents modules à déployer, il doit être possible de mettre en place une gestion des flux d'informations (workflow) et une définition des rôles de chaque personne. Pour chaque domaine de gestion, nous devons identifier qui fait quoi (saisie et validation) et avec quelle information utilise-t-il ou crée-t-il. Pour cette taille d'entreprise, il y a peu d'intérêt d'approfondir l'ensemble des activités de l'entreprise, une vision macroscopique peut suffire.

Pour les PME/PMI (de taille importante, de 200 à 499 personnes), nous devons être capable de déployer une solution logicielle correspondante à l'organisation (interne ou externe) de l'entreprise et à leur mode de gestion (à la commande, sur stock...). L'expression et l'analyse des besoins trouvent leur place. Elles permettent d'identifier, de façon détaillée, les acteurs du système, les informations circulant dans l'entreprise, d'identifier et de caractériser les processus propres à l'entreprise et l'enchaînement des activités qui les composent afin d'adapter au mieux la solution logicielle à l'entreprise (définition du déroulement des processus de gestion, identification des rôles, configuration d'un workflow). Les projets liés à cette taille de PME permettent et nécessitent d'identifier, de caractériser et d'optimiser les principales activités, cœur de métier de l'entreprise.

Il n'y a pas d'évolution majeure dans la mise à disposition de modules verticaux pour ce type d'entreprise. L'attente des PME/PMI, toutes tailles confondues, se situe plus dans l'acquisition et la mise en œuvre d'une solution intégrée (unicité de l'information) par rapport à des applications internes et variées n'assurant pas la cohérence de l'information. Ces informations sont la mémoire de l'entreprise. Il doit être possible de retrouver des informations sur les précédentes années comptables, par exemple. Par rapport aux modules verticaux, l'attente des utilisateurs se situe plus dans la simplicité et la souplesse d'utilisation des différents modules. La simplicité est caractérisée par la facilité à comprendre et à utiliser la solution. Les utilisateurs ne doivent ni être des informaticiens, ni lire une documentation en plusieurs tomes, ni suivre des formations de plusieurs mois pour utiliser un module particulier. Alors que la souplesse nécessite la mise en place d'une solution non rigide, modifiable facilement et capable, après quelques réglages, de s'adapter à une nouvelle organisation. L'utilisateur doit être capable de saisir les informations sans ordre précis et en limitant au maximum le nombre de renseignements obligatoires. L'éditeur doit proposer aux clients et futurs utilisateurs des modules « intuitifs » capables de devancer, si possible, les manipulations de l'utilisateur.

Les modules transversaux touchent de plus en plus les PME/PMI. Jusqu'alors réservés aux entreprises multinationales, ils pénètrent de plus en plus cette nouvelle taille d'entreprise, les PME. Ces modules transversaux permettent d'améliorer les processus organisationnels et les relations avec l'extérieur. Nous entendons par modules transversaux, des outils tels que les SCP (Supply Chain Planner) ou encore les outils de GRC (Gestion de la Relation Client). L'intérêt de ces modules est que les entreprises ne s'intéressent plus qu'au flux financiers mais elles prennent aussi en compte les flux d'informations et de matières entre les différents partenaires. Les dirigeants de PME/PMI souhaitent que les modules verticaux (gestion de production par exemple) soient plus intégrés avec des modules transversaux (Gestion de la Relation Client, BtoB, gestion logistique) afin de pouvoir prendre de meilleures décisions, en connaissance, et donc mieux piloter leurs entreprises vers le profit. De même, il est important de signaler que ces modules apportent une aide à la décision sur les choix à prendre, aussi bien d'un point de vue opérationnel que d'un point de vue stratégique.

Les futurs ERP doivent être tournés vers Internet avec des architectures ouvertes permettant l'ajout de nouveaux modules, de nouvelles fonctionnalités, sans casser le système existant, ni la base de données cœur de la solution. Mais il est important de garder une cohérence du système, principe de base dans la définition d'un ERP. La solution doit être ouverte vers Internet pour répondre à la mobilité des personnes et des entreprises. Même les PME/PMI ont des filiales ou des partenaires à l'étranger, le dialogue ne doit pas être rompu. L'outil doit répondre aux besoins de mobilité de la force de vente en restant toujours connecté au système central.

D'un point de vue financier, les clients souhaitent toujours que le coût global du projet (licences et prestations de service liées au projet – installation, déploiement et maintenance) soit de moins en moins onéreux. Cela doit être possible si nous proposons des solutions pré-configurées

ou pré-packagées aux clients. Cette tendance de réduction des coûts sera confirmée par les deux évolutions suivantes :

- avec la mise en place de méthodologie et de gestion de projet adaptées à cette cible,
- avec l'avènement de l'open-source permettant de réduire principalement les coûts de licences.

La mise en place d'une méthodologie adaptée aux PME/PMI est très importante. Nous nous en rendons compte avec la difficulté que connaissent les grands éditeurs d'ERP (SAP, People-Soft...) pour accéder à cette nouvelle cible. Les projets étaient trop longs, trop coûteux pour cette taille d'entreprise. Une vision trop détaillée, trop microscopique est utilisée alors qu'une vision macroscopique pouvait suffire.

De plus, avec l'avènement de l'open-source et la mise à disposition gracieusement d'outils d'ERP aux entreprises, les coûts de licences pour la mise en œuvre de ces outils devient obsolètes. En contrepartie, les coûts de service pour l'implémentation peuvent augmenter pour aider les entreprises à démarrer avec ce genre d'outils et cette nouvelle philosophie.

Afin d'assurer leur pérennité, les éditeurs devront donc proposer de plus en plus de services. Cela sera possible grâce aux gains de productivité obtenus par la mise en œuvre de nouvelle méthodologie de gestion des projets de déploiement, mais aussi dans la conception de leurs produits. Enfin le marché des progiciels et les prestations de services s'orientent vers une nouvelle approche d'extension des Systèmes d'Information : l'EAI (Enterprise Application Integration) et l'urbanisation des systèmes. Nous préciserons ces deux concepts dans les prochaines sections (§ 5.3).

Les éditeurs ou les intégrateurs doivent proposer des démarches projets adaptées à la taille et aux possibilités de l'entreprise cible.

L'objectif est de populariser un système qui doit être complet sans être complexe pour qu'il soit de plus en plus accessible pour les PME/PMI et les TPE.

Chapitre 4

Gestion des projets ERP

4.1 Définition du projet ERP

L'AFITEP, association francophone de management de projet, définit un projet comme un ensemble d'actions à réaliser pour satisfaire un objectif défini, dans le cadre d'une mission précise, et pour la réalisation desquelles on a identifié non seulement un début, mais aussi une fin.

L'implantation d'un ERP est un projet, puisqu'il demande un travail spécifique, délimité dans le temps, avec une mission de déployer une nouvelle solution informatique pour supporter les fonctions de gestion. Nous utiliserons par la suite la contraction projet ERP pour le désigner. Il y a souvent plusieurs objectifs, et nous avons déjà évoqué le fait que l'un des principaux est la satisfaction des utilisateurs. Enfin, la période de temps séparant la décision de réformer le Système d'Information et la mise en exploitation de l'ERP, est finie.

La gestion de projet consiste à réaliser des activités complexes orientées vers la production d'un bien ou d'un service mobilisant sur une période finie un ensemble de ressources ((Vincent Giard, 1991) et (Gilles Carel et al., 2001)). Elle permet donc de préciser quelles seront les ressources nécessaires pour la réalisation de chaque tâche dans l'ensemble du travail. Il s'agit de ressources, humaines, matérielles et financières. La réalisation du projet est assujettie au respect des coûts de revient, de la qualité du résultat et des délais de réalisation. La gestion de projet fixe également l'organisation, elle distribue les rôles des acteurs entre maîtrise d'ouvrage et maîtrise d'œuvre.

4.2 Management des projets

En se basant sur une enquête auprès de 150 entreprises, toutes tailles confondues, et où les PME/PMI (de moins de 500 salariés) constituent 70 % de l'échantillon, Robert Canonne et Jean-Louis Damret publient les résultats suivants :

- pour 60 % des entreprises, le délai de déploiement de l'ERP a été de 7 à 24 mois. Seul 10 % de ces entreprises ont eu un délai inférieur,
- les coûts sont détaillés dans le tableau suivant¹ (tab. 4.1). D'après les auteurs, 20 % des coûts représente l'achat des licences, et 10 % représente l'achat d'équipement. Le reste

¹Il est nécessaire de préciser aux lecteurs que malheureusement ces chiffres ne sont pas croisés avec la taille de l'entreprise, donc le nombre d'utilisateurs du système.

Dépenses engagées (Millions euros)	Fréq.
moins de 0,29 M€	28,00 %
de 0,3 à 0,74 M€	20,00 %
de 0,75 à 1,5 M€	18,67 %
de 1,6 à 7,4 M€	25,33 %
de 7,5 à 15 M€	2,67 %
de 16 à 30 M€	1,33 %
de 31 à 45 M€	4,00 %

TAB. 4.1 – Dépenses engagées

est imputable aux coûts internes de personnel, et aux coûts de services de conseil et d'assistance.

De nombreuses études montrent que le nombre d'échecs dans le déploiement d'un ERP est très important. En 1995, une étude réalisée par le Standish Group (CHAOS chronicle¹) estime à 69 % le taux d'échec de ces projets. En 2002, l'étude de Robbins-Gioia LLC² précise que 51 % des implantations d'ERP sont des échecs.

Les principales causes d'échecs sont de deux ordres :

- difficultés liées au projet : dérive sur les coûts humains engendrés par l'allongement des délais de réalisation,
- difficultés liées à la culture de changement : incompréhension dans l'utilisation et le fonctionnement du système et par conséquent refus de la part des utilisateurs.

La dérive sur les coûts peut avoir deux raisons, l'une concernant une sous-évaluation de la complexité du projet et l'autre un manque de compétences des parties prenantes. C'est pour cela qu'il est très important de travailler sur le périmètre fonctionnel du futur système et sur la composition des équipes projet (Redouane El Amrani, 2003).

La satisfaction des utilisateurs (Imen Maaloul et Lassaâd Mezghani, 2003) est un critère important à surveiller dans la mise en œuvre d'un nouveau système. L'implantation d'un nouveau système engendre des difficultés aussi bien individuelles qu'organisationnelles. Ces réticences se concrétisent par un échec dans l'appropriation de l'outil par les utilisateurs.

Il est clair que ces causes sont des caractéristiques fortes d'échecs des projets SI en général, et ERP en particulier. L'échec se constate rapidement alors que le succès se constate plutôt lentement.

La mesure du succès est plus difficile à mettre en place. Tout d'abord, parce que très peu d'entreprises mettent en place des indicateurs pour mesurer le retour sur investissement (ROI). Ces métriques doivent mesurer les résultats de l'entreprise et la performance du système dans le temps. En outre, le succès dans le déploiement d'un outil à court terme n'assure pas automatiquement un succès et une amélioration de la pérennité de l'entreprise, et de sa gestion, à long terme.

4.2.1 La décomposition d'un projet ERP en tâches

Il y a deux étapes importantes dans un projet ERP (fig. 4.1) : la sélection de l'ERP et son déploiement.

¹<http://www.standishgroup.com/chaos>

²http://www.robbinsgioia.com/new_events/012802_erp.aspx

La sélection de l'ERP est une étape préliminaire au déploiement qui peut se décomposer en quatre macro tâches :

- étude d'opportunités décrivant l'existant, les objectifs du projet, les bénéfices attendus, les enjeux et les facteurs clé de succès, les risques et les contraintes,
- expression du besoin se traduisant par un cahier des charges fonctionnel qui explique le périmètre fonctionnel, fournit la cartographie des flux, les règles de gestion et décrit les processus,
- constitution d'une liste d'éditeurs, qualifiés par rapport à leurs références et au degré d'adéquation avec le besoin, qui sont soumis à des jeux de tests,
- classement de ces éditeurs en fonction de leurs réponses et d'une liste de critères techniques et financiers, puis négociation de la solution choisie avec la tête de liste pour préparer le déploiement.

À ce stade de prospection et pour des raisons de charge de travail, il est rarement question de rentrer dans le détail des processus et les jeux de tests ne poussent pas les progiciels dans leur retranchement. Si le cahier des charges fonctionnel explique ce que veut l'entreprise, la réponse de l'éditeur traduit ce que peut faire le progiciel par ses processus standard. La mesure de l'adéquation entre demande et réponse est alors difficile, car la tentation de modifier le besoin initial pour coller au standard est forte. Cette tentation est d'autant plus forte que bon nombre de projets ont un objectif implicite de réorganisation des processus de gestion (Robert Canonne et Jean-Louis Damret, 2002).

Seconde étape importante, le déploiement de l'ERP peut être décrit en cinq macro tâches : le lancement, la conception détaillée de la solution, la réalisation de la solution, l'intégration et le passage en production (Jean-Luc Deixonne, 2001). Jean-Louis Tomas détaille douze tâches qui peuvent être considérées comme une décomposition plus poussée des macro tâches précédentes :

- la planification : plan de communication, besoins d'interfaces, plan des risques majeurs, (lancement),
- l'analyse opérationnelle : définir les processus opérationnels, documenter, former les utilisateurs et décideurs du projet, (conception détaillée),
- la formation des équipes projet (lancement),
- la configuration de l'ERP : adéquation entre les processus et les modules implantés, (réalisation),
- le prototypage : vérification d'un fonctionnement correcte, (réalisation),
- la simulation en taille réelle, (réalisation),
- la fermeture des TFP (Trous Fonctionnels Potentiels) : développement de spécifique, si nécessaire, pour répondre à des besoins de l'entreprise, nullement ou incomplètement couverts par l'ERP, (réalisation),
- les modifications spécifiques, (réalisation),
- la connexion avec l'existant, (intégration),
- la documentation utilisateur, (réalisation),
- la préparation de mise en production (passage en production),
- le déploiement (passage en production),

L'outil est choisi, c'est donc le temps de la juste adéquation entre le logiciel et l'organisation. L'analyse se veut plus détaillée que dans l'étape de sélection. Le problème de la méthode reste pourtant quasiment le même. Il s'agit toujours d'exprimer un besoin et de trouver les solutions dans le progiciel en proposant un bon paramétrage des fonctions standard. Les jeux de tests

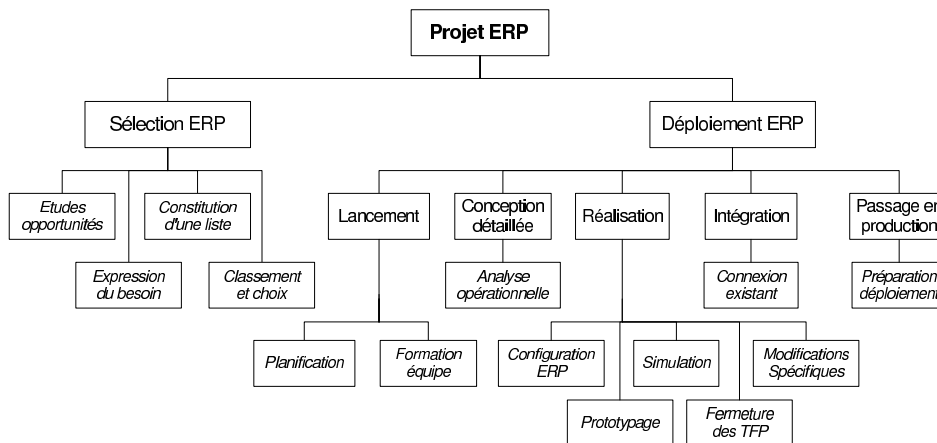


FIG. 4.1 – WBS d'un projet ERP

sont multipliés, notamment parce que le souci de validation grandit avec l'approche du jalon de mise en production du progiciel. Si une fonction recherchée n'est pas proposée en standard et que l'organisation veut en disposer, le trou fonctionnel est comblé par un développement spécifique. La réalisation de certaines tâches est conditionnée au résultat de tâches qui les précèdent.

4.2.2 L'organisation des équipes du projet

Nous retrouvons les acteurs suivants :

- les éditeurs qui fournissent des solutions logicielles,
- les intégrateurs qui assurent la mise en adéquation du logiciel au client,
- le client final, ou ses conseils.

La maîtrise d'œuvre (MOE formée par l'éditeur et l'intégrateur) est responsable de la construction, du déploiement et de la maintenance du Système d'Information. La maîtrise d'ouvrage (MOA client et conseil) garantit l'alignement entre la stratégie, les processus et les Systèmes d'Information. Elle est constituée d'un niveau stratégique, porteur de la vision métier, et d'un niveau opérationnel, capable d'en assurer la traduction en vision Système d'Information (Alain Berdugo et al., 2002). Les principales activités de ces deux parties peuvent être décrites de la manière suivante :

- pour la Maîtrise d'Œuvre :
 - elle apporte et forme la MOA sur les nouvelles technologies ;
 - elle traduit les besoins exprimés par la MOA en proposant des solutions logicielles adaptées ;
 - elle maintient et exploite les solutions déployées ;
- pour la Maîtrise d'OuvrAge :
 - elle dirige le projet afin de le mener à bien ;
 - elle représente les utilisateurs et détermine des principales fonctions à déployer ;
 - elle suit la réalisation et l'intégration du Système d'Information pour vérifier que les besoins identifiés correspondent aux solutions proposées.

Dans le contexte de PME/PMI, avec des moyens humains et financiers généralement limités, la relation entre le client MOA et l'éditeur de solution logicielle est souvent très directe. Il est rare, dans ces projets ERP, de faire intervenir un intégrateur entre ces deux parties, principalement à cause du surcoût. Donc c'est généralement à l'éditeur d'assumer pour partie le rôle d'intégrateur, situation où il est à la fois juge et partie.

L'expression du besoin dans la phase de sélection et la conception détaillée dans la phase de déploiement sont des tâches où MOE et MOA sont en contact et doivent établir les attendus de la mission.

Le déroulement du projet peut s'accompagner de difficultés, dès ces phases initiales :

- difficulté dans l'expression du besoin par la maîtrise d'ouvrage (MOA) se soldant par un cahier des charges fonctionnel incomplet voire incohérent,
- difficulté, corrélée à la précédente, dans l'expression d'une offre en adéquation avec le cahier des charges par la maîtrise d'œuvre (MOE) avec le risque de déployer une solution non adaptée.

Ces difficultés initiales peuvent se trouver amplifiées dans le cycle de vie du projet du fait de l'instabilité dans la position adoptée par l'une ou l'autre des parties, et de la durée assez longue des projets. Par exemple, la MOA peut introduire des changements dans l'organisation, dans les missions, dans les processus de gestion, ou même introduire des nouvelles contraintes liées à la modification de son environnement. De son côté, la MOE peut faire évoluer ses versions sur le plan fonctionnel ou sur le plan technique, elle peut aussi changer son interlocuteur.

Il est donc nécessaire d'établir un dialogue constructif entre MOA et MOE dès le début du projet, mais le référentiel initial qui concrétisera les résultats constructifs de ce dialogue pourra être adapté au gré des variations introduites par les interlocuteurs tout au long du projet. Pour supporter le dialogue, la mise en place de méthodes adaptées, dans un langage accessible à tous, est primordiale.

4.3 Les méthodes utilisées pendant la sélection du progiciel

Dans la majorité des projets, le cahier des charges fonctionnel est essentiellement littéral. Il peut s'appuyer sur des questionnaires, comme celui du CXP ou du CETIM (annexe A), dans le but d'être aussi complet que possible dans l'énoncé des besoins.

En phase de sélection, c'est l'éditeur qui répond aux questions et le maître d'ouvrage doit miser sur la bonne foi de ce dernier pour les points qui ne sont pas pris en compte par le jeu de tests, puis appliquer des pondérations entre les réponses pour réaliser son classement des propositions.

La modélisation d'entreprise pourrait rentrer en jeu à ce stade, pour permettre d'améliorer l'analyse des premières étapes de l'implémentation (Valérie Botta-Genoulaz et al., 2001). Elle offre des méthodes et des outils permettant de mieux comprendre et d'analyser les processus opérationnels autour du système de production. Son adaptation à la couverture fonctionnelle des processus touchés par l'ERP peut être réalisée. Le marché des progiciels de modélisation d'entreprise s'est développé récemment, et bon nombre d'acteurs revendiquent l'utilisation de leur outil (§ 4.4.2) dans l'optique de réaliser cette cartographie des flux et des processus. L'offre contient parfois le lien avec un progiciel particulier, pour assurer une bonne communication.

4.4 Les méthodes utilisées pendant le déploiement

4.4.1 Méthodes

4.4.1.1 Conception détaillée

Le premier élément nécessaire dans le déploiement de l'ERP est l'utilisation de connaissances sur l'application. Pour exploiter cette connaissance, le client, l'intégrateur et l'éditeur ont à choisir le niveau de granularité de l'analyse et éventuellement un formalisme de représentation.

Le niveau de granularité du modèle pour la spécification détaillée est une question délicate. L'entreprise peut employer des processus standard et rester à un niveau très macroscopique, ou elle peut descendre à un niveau très fin de détail sur tout ou partie d'un processus donné pour vérifier des éléments critiques de son mode de gestion. Le modèle doit donc permettre la modification du niveau de généralité de la représentation des processus, au gré de l'utilisateur.

Pour le formalisme, en ingénierie des Systèmes d'Information, la méthode MERISE est très employée. Elle est basée sur deux composantes : la représentation des données du système par le modèle de données, la représentation des traitements du système (les processus) par le modèle de traitement. Elle est employée dans beaucoup de projets, mais souvent avec une variation du formalisme de base sur les traitements. Comme par exemple la technique de modélisation d'ARIS, utilisée par l'éditeur SAP, qui propose un traitement par les EPC (Event Process Chain) pour décrire les processus.

Actuellement, quelques éditeurs proposent un modèle de l'application qui est une source utile de connaissance. Cela peut constituer un appui pour l'expression de besoins. Pour les intégrateurs de SAP, l'utilisation du modèle dans la démarche de déploiement ASAP fût un échec alors que les intégrateurs d'Oracle continuent de déployer leur système au travers de leur modèle et leur outil. La possession d'un modèle de la solution et son utilisation dans une démarche d'intégration est un avantage concurrentiel pour les éditeurs dans la qualité (coût, délai et performance) du déploiement de leurs solutions.

4.4.1.2 Paramétrage

Le paramétrage est la configuration de la solution logicielle pour l'adapter au contexte de l'entreprise, afin d'obtenir ainsi le Système d'Information.

Jean-Louis Tomas (1999, 2002) définit la configuration d'une solution logicielle comme la réalisation de trois étapes (adéquation, configuration et prototype) afin d'obtenir une acceptation de la part de la MOA sur le résultat obtenu. Nous pouvons synthétiser cette démarche par la spirale décrite par la figure suivante (fig. 4.2).

L'étape d'adéquation permet d'identifier un à un les processus et les documents de l'entreprise, définis précédemment lors de l'étape d'analyse, dans la solution logicielle. Si cela n'est pas possible, s'il existe une inadéquation, cela veut dire que nous avons identifié un TFP (Trous Fonctionnels Potentiels). Deux solutions sont avancées : modifier l'ERP par le développement d'un spécifique pour cette entreprise ou modifier l'organisation de l'entreprise.

L'étape de configuration est composée de deux tâches : le paramétrage de la solution et la réalisation de scénarios pour le prototypage. Le paramétrage de la solution logicielle consiste à exécuter les décisions prises durant l'étape d'adéquation. C'est-à-dire paramétrer, valuer les différentes données structurelles. Nous pouvons citer par exemple :

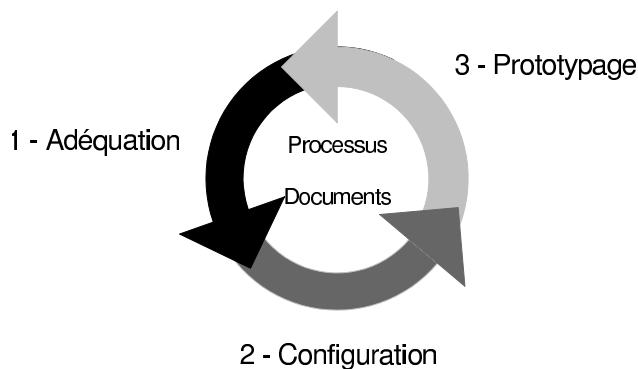


FIG. 4.2 – La spirale adéquatation/configuration/prototypage

- définition des processus de l'entreprise,
- identification des acteurs, rôles et groupes de l'organisation,
- affectation des rôles pour tel processus,
- identification des documents de l'entreprise,
- codification des documents,
- identification des événements attachés au processus,
- définition des règles métiers.

La deuxième tâche consiste à documenter les scénarios qui seront exécutés dans l'étape de prototypage. Cette documentation permet de transférer les informations, à la personne en charge du prototypage, sur :

- les fonctions de l'ERP utilisées,
- les paramètres,
- les fonctions de navigation utilisées,
- les écrans (affichage des champs, des valeurs paramétrées...).

L'étape de prototypage consiste donc à tester la solution paramétrée afin de vérifier à l'aide d'un scénario, reproductible à volonté, que pour la réalisation de tel processus nous obtenons le résultat souhaité. Si ce processus ne donne pas satisfaction, ne respecte pas le cahier des charges, alors nous pouvons boucler sur la spirale adéquatation, configuration et prototypage. Dans le cas contraire, s'il y a acceptation du scénario, alors nous pouvons sauvegarder la configuration de l'ERP et passer à l'étape suivante d'intégration et de passage en production.

Des travaux de recherche sont menés pour faciliter la réalisation de ces étapes. Jon Atle Gulla et Terje Brasethvik (2002) travaillent actuellement à la réalisation d'un outil pour SAP afin de faciliter le paramétrage de la solution R/3. Il existe actuellement, dans l'outil ARIS, un modèle de SAP R/3 permettant d'avoir une vision des différents éléments (processus, informations, règles) paramétrables. Ces auteurs précisent qu'ils sont arrivés jusqu'aux limites des outils d'aide au paramétrage, tel que ARIS, et favorisent l'utilisation d'un formalisme plus générique et adaptable au contexte de paramétrisation d'un ERP. Ils proposent donc d'utiliser le formalisme UML (Unified Modeling Language) et, plus particulièrement, le diagramme d'activités pour représenter l'enchaînement logique des processus de l'entreprise et réaliser une projection entre les éléments du diagramme (activités, note, objet, logique d'enchaînement) et les éléments de la solution (écran, codification, document, workflow).

Dans le cas de grands comptes, l'étape de paramétrage est généralement longue et difficile (plus de 15 % de la durée du projet) due à la complexité de l'organisation. Dans le contexte

de PME/PMI, nous avons moins de processus et de documents à paramétrer, par contre, nous avons une grande variété d'entreprises (secteurs d'activités variés), avec des processus plus « standard », et des durées de projet proportionnellement inférieures à celle des grands comptes. Les éditeurs proposent donc de plus en plus des solutions pré-configurées. Après avoir identifié le mode de fonctionnement de l'entreprise, le secteur d'activités (négoce, industrie fabrication, industrie process) avec des outils tels que les grilles GRAI (§ 7), nous pouvons fournir une pré-configuration adaptée à cette entreprise et à son mode de gestion.

4.4.2 Outils des éditeurs ERP

Les éditeurs ont des outils d'aide pour le déploiement de leurs solutions. Les outils des gros éditeurs sont :

- ARIS¹ pour la méthode ASAP² (Accelerated SAP) de SAP,
- DEM³ (Dynamic Enterprise Modelling) de BAAN,
- Fast Forward⁴ d'Oracle.

Nous n'avons pas trouvé, à notre connaissance, de documents sur le retour d'expérience lié à l'utilisation de ces outils. Les seules descriptions en notre possession proviennent des sites Web des éditeurs à vocation trop commerciale.

Les objectifs de ces outils sont :

- d'accélérer l'implémentation de l'ERP,
- de structurer le projet en utilisant l'outil comme support,
- de proposer un outil d'aide à la configuration de la solution,
- de fournir un moyen de représentation de l'organisation de l'entreprise
- d'offrir la possibilité, avec l'outil, de faire évoluer les paramètres et l'organisation de la solution logicielle.

Ces outils apportent une aide aux différents équipiers du projet en facilitant la gestion des informations sur les processus ou les documents mise en œuvre de la solution logicielle. Si nous devons gérer, à la main, l'ensemble de ces informations sur un tableau mural d'un simple processus représentant le suivi d'une commande dans une entreprise, cela deviendrait illisible et ingérable.

4.5 Analyse critique

4.5.1 Sélection de l'ERP

L'utilisation des grilles dans la sélection de l'ERP n'est pas très bien adaptée à la recherche et la définition des besoins du client. Cette démarche souffre d'un manque d'accès à la définition technique de chaque outil pour permettre de juger la cohérence de la réponse. Cette grille est normalement remplie par l'éditeur ou l'intégrateur. Nous retrouvons généralement des taux de correspondance entre les besoins et les fonctionnalités de 99 %. De plus, en utilisant plus de 500 questions, il est très difficile de mettre l'accent sur le processus sensible de l'entreprise.

¹<http://www.ids-scheer.com/>

²<http://www.sap.com/france/services/methodes/fonctionsclles.asp>

³<http://www.baan.com/solutions/ibaanoptimization/dem/overview/>

⁴<http://otn.oracle.com/products/designer/content.html>

La communication, pour être efficace, devrait permettre au client d'aller puiser lui-même les réponses aux questions qu'il se pose à travers une représentation formelle des capacités du progiciel, d'où le recours à un modèle de l'application appréhendé comme un modèle de Système d'Information. Si le client, en utilisant le modèle, trouve les réponses à ses questions, cela permettrait de le conforter dans son approche du sujet.

4.5.2 Déploiement de l'ERP

Le principal problème, dans le déploiement de l'ERP, est la profusion des différents formalismes. Chaque éditeur, chaque intégrateur, possède son propre formalisme inspiré de sources diverses, que ce soit MERISE, SADT...

Cette diversité s'oppose aux vertus recherchées par des moyens de communication entre la Maîtrise d'Ouvrage et la Maîtrise d'Œuvre, qui doivent parler un langage commun (c'est-à-dire une syntaxe et une sémantique). La syntaxe permet de décrire les règles régissant les relations entre les éléments du langage, tandis que la sémantique apporte un sens au modèle, résultant de la mise en relation des précédents éléments, dans un contexte donné. Il est nécessaire que la Maîtrise d'Œuvre forme et spécialise ses consultants aux outils attachés à chaque application pour adopter une démarche rationnelle auprès de ses clients, qui n'ont généralement pas le temps de s'approprier le modèle résultant.

Deuxième partie

Conception d'un ERP

Chapitre 5

Architecture d'ERP

5.1 Vers une vision large de l'« Architecture »

Dans son acception courante, l'architecture est tout à la fois l'art, la manière et la technique de construire des édifices, le plus souvent selon des règles déterminées (Dictionnaire Hachette). C'est une définition d'un métier exercé principalement dans le secteur du bâtiment et des travaux publics. L'architecte est la personne qui :

- exerce cet art,
- emploie la manière et la technique pour établir des plans de l'édifice,
- exerce un contrôle de l'exécution des travaux, conformément à ses spécifications.

Est-il vraisemblable d'étendre plus largement l'usage de ce terme à des systèmes ? Peut-on parler d'architecture d'entreprise ? d'architecture de Système d'Information (et donc d'architecture d'ERP) ?

Voilà une série de questions qui peuvent apparaître anachroniques à ce stade de l'exposé. Pourtant, elles sont le fondement même de notre étude, et nous allons essayer d'en expliquer la légitimité dans cette seconde partie du mémoire, et en particulier dans ce chapitre.

L'architecture de système est un terme qui est donc fondé sur une analogie, et qui nous semble être d'un emploi justifié quand les deux points suivants sont avérés :

- c'est de la conception de systèmes complexes. En effet, il n'est pas possible de représenter entièrement le résultat de la conception dès le début. Il y a trop d'éléments à appréhender et de multiples façons de regarder l'objet produit. La vision de l'ensemble est extrêmement difficile à imaginer avant la réalisation effective. On se contente donc de formes de représentation qui assurent le respect des volontés du client et qui le satisfont au fur et à mesure de l'avancée du travail. Le client peut, dans la mesure du possible, être amené à faire des ajustements en cours de réalisation de l'ouvrage pour corriger des lacunes ou des incompréhensions initiales. La conception est basée sur la définition, puis l'assemblage de composants ou modules, eux-mêmes issus de la contribution de plusieurs parties où des matériaux et des métiers différents sont appelés à intervenir. Il faut essayer, dans tout l'espace de décision qui conduit le travail jusqu'à la réalisation finale du produit, de trouver des équilibres entre créativité, faisabilité et satisfaction client (délai, coût, qualité), ainsi que respecter des propriétés attendues comme la solidité du produit, la facilité d'utilisation, la maintenabilité...

- c'est un travail très spécifique, et qui doit être géré comme un projet, car chaque cas est particulier (le client a des goûts personnels, le contexte est différent, les moyens sont variables, ...). Il faut satisfaire le besoin du client, qui a une idée assez générale de ce qu'il veut et de ce qu'il ne veut pas. Il faut l'accompagner dans la définition de son besoin jusqu'à un stade où la conception détaillée devient possible. Cet accompagnement traverse un champ d'objectifs et de contraintes qui vont guider les choix. Cette approche progressive doit être structurée et documentée, ne serait-ce que pour fixer un cadre juridique dans les responsabilités de chaque partie. Le dialogue est supporté par des documents divers et variés (esquisse, croquis, perspectives, plans, cahier des charges, spécification détaillée, acte d'achat...). À partir d'un certain stade de définition, il faut faire participer des techniciens pour obtenir des garanties sur la faisabilité, et avoir des informations techniques fiables. Il peut y avoir un coordinateur des différents métiers, le fameux maître d'œuvre, qui fait office de référent vis-à-vis de l'architecte. C'est de cette coopération tripartite (architecte, maître d'œuvre, client) qu'émerge l'organisation du travail de réalisation et sa conduite.

John A. Zachman (1987) a dressé un parallèle entre les démarches d'architecte en bâtiment et d'ingénierie de produits (avion et Système d'Information). Nous reportons la synthèse de ces travaux dans le tableau 5.1. Cette analyse l'a conduit à proposer un cadre générique d'architecture d'entreprise, que nous présenterons ultérieurement.

Générique	Bâtiment	Aéronautique	Système d'Information
Domaine	Croquis	Concepts	Objectifs/Contexte
Représentation client	Dessins	Décomposition fonctionnelle	Modèle métier
Représentation conceptuelle	Plans d'architecte	Conception/nomenclatures	Modèle du Système d'Information
Représentation du fabricant	Plans de réalisation	Organisation de la fabrication	Modèle des applications
Spécifications	Plans des corps de métiers	Assemblage et plan de montage	Description détaillée
Représentation du langage	<i>Hors sujet</i>	Codes numériques	Description du langage machine
Produit	Bâtiment	Avion	Système d'Information

TAB. 5.1 – Représentation comparée des démarches d'architecte et d'ingénieur

L'architecte en Système d'Information est un métier référencé comme tel par les organisations spécialisées dans la gestion des carrières et des emplois (http://www.onisep.fr/national/fiches_métier) :

« L'architecte en Système d'Information intervient pour créer, refondre un système jugé obsolète ou pour en construire un, commun à plusieurs entreprises appelées à fusionner. Le Système d'Information peut être divisé en sous-systèmes : commercial, comptable, industriel, etc. L'architecte commence par étudier l'existant : les

*systèmes d'exploitation, les matériels, les logiciels, les réseaux en place. Plusieurs systèmes peuvent cohabiter sans être compatibles. En fonction du schéma directeur élaboré avec l'entreprise, l'architecte en Système d'Information construit la cartographie du Système d'Information. Il choisit les technologies qui vont être utilisées dans le système, dans le respect des contraintes de coût, de délai, de sécurité... et en exploitant au mieux les possibilités existantes. **Il doit aussi comprendre comment fonctionne l'entreprise et formaliser les différents métiers de l'entreprise en concepts fondamentaux.** Il décide des outils d'information, de l'implantation des postes de travail, de l'organisation des procédures. Il pilote le déploiement du Système d'Information. Son rôle n'est pas strictement technique. Il prévoit et organise les évolutions et la cohérence du Système d'Information, en repérant et en analysant tout événement susceptible d'avoir un impact sur celui-ci. Il informe et conseille la direction de l'entreprise sur les modifications technologiques ou organisationnelles que peut entraîner un nouveau Système d'Information. C'est un métier qui requiert un niveau technique très élevé et une maîtrise des nouvelles technologies : des systèmes d'exploitation, systèmes gestionnaires de bases de données ou équipements réseaux. Il faut posséder une connaissance approfondie de l'entreprise et de ses métiers, une expérience confirmée de la conduite de projets. La faculté à écouter et à communiquer, à travailler en équipe, la capacité à négocier et convaincre, le sens stratégique et politique, sont déterminants. »*

À la lecture des qualités souhaitées pour ce candidat, on ne s'étonnera pas que bien des projets n'aient pas mis la main sur l'oiseau rare ! Il n'en reste pas moins que l'ensemble des missions déclinées pour cet architecte sont essentielles au succès des projets de SI, et en particulier aux projets ERP. Toute contribution qui tend à supporter techniquement la charge de travail et la complexité des tâches est de grande utilité pour l'architecte et pour l'entreprise.

L'exercice du métier est organisé autour de trois points de vue principaux :

- **le point de vue fonctionnel** : la mise en place d'une architecture doit être en adéquation avec les exigences du client. Cette partie fonctionnelle est en charge de l'identification, la définition et la réalisation des spécifications du client. Il faut donc une perception fine du système, au plus près des métiers et des utilisateurs.
- **le point de vue logique** : chaque exigence du client doit aboutir à des choix techniques, abordables (coût, délais, performance), du système. Il s'agit de mettre en adéquation des exigences du client avec les possibilités du système et du marché. Par exemple, l'application de gestion des commandes n'est pas la même si l'administration des ventes traite une ou des milliers de commande(s) par jour.
- **le point de vue physique** : chaque système s'insère dans un environnement technique particulier. Il est donc nécessaire de prévoir l'intégration et la connexion avec les systèmes existants. L'architecte est alors en charge de la conception, du design du niveau du système (composants, interfaces). Il développe une vision macroscopique de celui-ci. Par exemple, il est possible que le système ait besoin de se connecter à un système de gestion de bases de données existant et de récupérer les données dans les bases en service, ou encore de coopérer avec un logiciel métier déjà en service.

Le premier point de vue, dit fonctionnel, est relatif à la formulation du problème (l'analyse). Les deux autres points de vue sont plutôt tournés vers la définition de la solution cherchée, sur le plan des capacités de gestion de l'information et surtout de son traitement pour le point de vue logique, et sur l'implémentation pour le point de vue physique (la conception).

En pratique, gagner du temps et être efficace se conjuguent par la maîtrise des moyens de représentation. Selon Ricardo Chalmeta et al. (2001), il y a une demande forte de ces architectes pour des outils et des démarches d'analyse fondés sur les modèles. Et ces auteurs précisent

que les modèles d'entreprise et les méthodologies de conception qu'ils sous-tendent forment les architectures de référence. C'est un cadre qui guide la représentation formelle des opérations et des supports opérationnels d'un « système d'entreprise intégré », employé pendant le projet d'ingénierie du Système d'Information. Ces auteurs font bien la distinction entre les architectures de référence appliquées à l'entreprise dans sa globalité, et celles appliquées au Système d'Information intégré. Nous aurons donc à détailler ces deux approches.

5.2 Quelques caractéristiques architecturales d'un ERP

Puisqu'un ERP est une application logicielle à caractère standard, il se présente comme une forme de solution et sa description n'est inhérente qu'aux points de vue de la logique et du système. Ces systèmes sont très compliqués et il est hors de question de dresser ici un bilan exhaustif de leur architecture. Nous présentons les principes de base communs à la plupart de ces systèmes.

5.2.1 Point de vue logique de l'architecture

L'architecture logique d'un ERP a les traits caractéristiques suivants (Jean-Michel Tysebaert, 2001) :

- construit autour d'un SGBD unique,
- offre des ensembles de traitements regroupés par métiers dans des modules,
- met à disposition des structures de données caractérisant les différentes ressources,
- possède un système de contrôle de cohérence pour fixer des conditions de lancement des traitements.

Nous reprenons ci-dessous chacune de ces caractéristiques.

Dans le but d'intégrer correctement les informations, les architectures d'ERP sont toutes construites autour d'un Système de Gestion de Bases de Données (SGBD). Ce SGBD a la charge d'une ou de plusieurs bases de données, en relation étroite avec le reste de l'application. La plupart des gros ERP offrent des adaptations aux bases les plus répandues du marché, ils sont dits « multibases ».

Un module est un sous-système de l'ERP qui remplit une fonction de haut niveau. Plusieurs modules sont en général mis à disposition et le périmètre fonctionnel couvert par l'application est relatif aux modules mis en service. Les modules peuvent être classés en trois domaines distincts :

- Le domaine métier est constitué d'une série de modules remplissant les fonctions des processus qui sont situés sur la chaîne de valeur de l'entreprise : des commandes à la livraison en incluant les achats et la facturation client. C'est le territoire intérieur à l'entreprise où les flux d'information sont directement corrélés aux activités de production et de logistique. S'il s'agit d'une entreprise manufacturière, par exemple, nous parlerons de modules de gestion des achats, de gestion de production, de gestion des stocks, etc.
- Le domaine du pilotage est constitué de modules dédiés à l'analyse et à la synthèse du fonctionnement du système entreprise. Il peut s'agir de facilités d'accès à des prévisions, d'évaluation de performance, de consolidation des activités... Les activités financières et comptables, la relation client, les tableaux de bord du contrôle de gestion font partie de ce domaine.

- Le domaine du support prend en charge l'administration du système informatique. Ce domaine couvre l'ensemble des fonctionnalités permettant d'entretenir la structure de données et de modifier les paramètres de configuration de la solution logicielle.

Jean-Michel Tysebaert (2001) dresse un inventaire des principaux domaines d'activités de l'entreprise : la gestion commerciale, la gestion de production, la logistique des ventes et celle des achats, la gestion des stocks, la gestion des ressources humaines, la comptabilité, le contrôle de gestion et les finances. Pour chaque domaine, il utilise deux modes de représentation : une représentation externe et une représentation interne de l'entreprise, suivant différents niveaux de granularité (que nous présentons un peu plus loin). La représentation externe situe le fonctionnement de l'entreprise par rapport aux acteurs de son environnement. Par exemple, le client « achète » des produits à l'entreprise. La représentation interne détaille les traitements à effectuer au sein de l'organisation. Par exemple, l'acte de vente du produit à ce client induit un certain nombre de traitements : identifier le client, valider la référence produit, livrer le produit, facturer le produit, transférer ces informations à la comptabilité...

Les processus, ici la logique des traitements de l'information répartis dans différents modules, utilisent (en création, modification et suppression) un ensemble d'informations. Ces informations sont régies par une structure de données particulière, souvent appelée référentiel de données. Cette structure caractérise les différentes ressources de l'entreprise. Ces ressources sont les moyens humains, matériels et financiers de l'entreprise. Par exemple, pour un calcul des besoins, nous allons utiliser les informations sur les machines (capacité, rendement, ...) et les produits (nomenclature, gamme, qualité, ...) afin de faire des propositions de planification des achats de composants, ou de lancement des opérations de production. Partant du principe qu'on ne peut pas transformer ce qu'on a mal décrit, les capacités de traitement de l'information sont dépendants de la nature et du type de l'information disponible. Le concept de granularité de l'information fait référence à ce degré de perception de la réalité. On ne peut pas gérer de la même façon selon que l'on a accès, par exemple, à une représentation des stocks à l'échelle d'une palette, d'un colis ou d'une référence produit.

Le système de gestion de la cohérence de l'information a plusieurs rôles. Un premier rôle, le plus simple, est la vérification des informations saisies et du type de ces informations (date, chaîne de caractère, numérique) par rapport au référentiel de données. Un deuxième rôle, plus complexe, est la vérification du bon usage de cette information par les différents acteurs métier. Une même information sera utilisée par plusieurs modules dans des traitements différents. Le système de gestion de la cohérence est chargé de signaler une incohérence lors d'un traitement. Pour cela, il dispose d'un jeu de conditions préliminaires à chaque traitement, qu'il va vérifier. En cas de problème, il peut être utile de proposer à l'utilisateur des pistes de diagnostic afin de poursuivre son travail. Par exemple, dans l'exécution du traitement « calcul du besoins », il est possible que le niveau de stock des produits soit mal saisi (stock négatif) ou soit omis (information manquante), dans ce cas un message explicite doit être produit. Notons qu'un moteur de workflow, qui désigne la prise en charge automatisée des processus par l'ordinateur, peut être considéré comme faisant partie de ce mécanisme de gestion cohérente, puisqu'il fixe des relations d'ordres partiels dans le traitement des tâches.

Pour prendre connaissance de l'architecture logique d'un ERP, il faut donc au minimum :

- avoir étudié le référentiel de données,
- pour chaque module, savoir accéder aux traitements disponibles en fonction du besoin exprimé à différents niveaux de granularité de l'information, et appréhender le fonctionnement et les règles du système de gestion de cohérence.

Notons que pour les applications majeures du marché de l'ERP, telles que SAP R/3 ou Oracle Applications, il est impossible pour une personne seule d'atteindre ce seuil de connaissance

minimum. C'est une intégration par un travail d'équipes avec plusieurs spécialistes qui amènera à la couverture requise du périmètre fonctionnel.

Peu de références dans la littérature fournissent des détails suffisants pour faire de l'auto-apprentissage sur cette architecture logique sur un produit ERP donné. Pour illustration, nous renvoyons le lecteur au handbook de Campus Press sur SAP (SAP, 1999). En plus de 1 200 pages de texte, vous ne pourrez pas reconstruire la structure de données, même au niveau des besoins d'un module. Le contenu, extrêmement intéressant, laisse in fine une impression de survol du sujet !

Il y a donc dans cet état de fait un élément clé de notre raisonnement, comment mener efficacement le travail de l'architecte sur le point de vue fonctionnel en se trouvant limité par la connaissance sur le point de vue logique ? Comment concevoir un référentiel sur lequel se baser pour avoir accès à cette connaissance ?

5.2.2 Point de vue physique de l'architecture

5.2.2.1 Exigences de performance et d'exploitation

Le point de vue physique de l'architecte s'intéresse aux exigences de performance de l'utilisateur et du client sur :

- la montée en charge de l'exploitation et le taux de service : capacité des serveurs de répartir le traitement sur plusieurs machines, ou capacité du serveur à satisfaire plusieurs clients au même instant,
- la maintenance en exploitation : capacité du système à modifier ou faire évoluer un de ces modules sans impacter les modules connexes,
- l'ouverture du système : capacité du système à coopérer avec des systèmes du voisinage et aptitude à réutiliser des systèmes (ou composants) déjà existants.

À ces performances, nous pouvons ajouter les contraintes d'exploitation, comme par exemple :

- avoir des points d'accès à la solution logicielle par des terminaux de technologie différente (PC nomades, organiseurs, téléphones portables...),
- mettre en place une gestion des rôles (droits d'utilisation, connexion avec l'annuaire de l'entreprise...),
- fonctionner sur les systèmes d'exploitation de l'entreprise (utilisation d'outils de bureautique, connexion avec des systèmes existants sur des plates-formes spécifiques : gestion de données techniques, CAO...),
- avoir une interface homme-machine ergonomique (faciliter la prise en main pour les utilisateurs, proche des standards de faits - copier, coller - par exemple...),
- garantir une sécurité suffisante au niveau des informations à caractère stratégique.

5.2.2.2 Choix architecturaux

La dynamique de progrès du secteur de l'informatique et de la communication est telle que l'architecture physique est un thème qui nécessite une veille permanente et des remises en cause assez fréquentes. La pertinence des choix architecturaux est souvent mise à mal par l'usure du temps sur des périodes de quelques mois. Même si les technologies évoluent dans le sens d'une portabilité des applications et d'une compatibilité des systèmes toujours plus grandes, il y a

encore une grande diversité de l'offre et des jeux concurrentiels importants entre les différents fournisseurs informatiques (sur les logiciels de base et les moyens de communication) qui induisent une prise de risque.

L'architecture client/serveur est aujourd'hui un point commun à la plupart des produits ERP. Elle fournit des solutions pour satisfaire certaines des exigences et contraintes énoncées ci-dessus :

- le poste client peut être un client lourd (application sur chaque poste) ou un client léger (via un navigateur Web) ;
- une ouverture à plusieurs types de canaux : Internet, Serveurs téléphoniques (SMS), PDA ;
- le poste serveur assure les transactions entre le poste client et la base de données, mais aussi certains des calculs de modules métier de l'entreprise ;
- plusieurs systèmes de gestion de bases de données sont proposés, en général il s'agit de SGBD reconnus sur le marché et qui font partie des directives techniques des entreprises ;
- le fonctionnement sur plusieurs systèmes d'exploitation est également possible ;
- des facilités sont proposées pour les ouvertures au reste du Système d'Information (protocole d'échange de données, accès aux fonctions, communication synchrone avec d'autres applications, ...).

Les points de vue logique et physique de l'ERP sont intimement liés. Qu'il s'agisse de l'accès aux données, de la logique opérationnelle des traitements, des moyens d'utiliser l'application, l'architecte doit assurer une conception harmonieuse entre les deux points de vue. Dans le cadre de notre mémoire, nous allons toutefois plutôt nous focaliser sur le point de vue logique en relation avec le point de vue fonctionnel.

5.2.3 Représentations architecturales

Dans cette section, nous illustrons par quelques exemples simples des représentations sur l'architecture logique et l'architecture physique d'un ERP.

5.2.3.1 Point de vue logique de l'architecture

La figure 5.1 décrit un processus invoquant plusieurs modules d'un ERP. Nous retrouvons, aussi bien, des domaines verticaux (métier) avec :

- la gestion des ventes : saisie et enregistrement des commandes,
- la gestion de la relation client : suivi de l'état d'avancement d'une commande au niveau de la livraison et de la facturation,

que des domaines transversaux (support et pilotage) avec :

- la gestion des données techniques : définition des produits et des procédés par des informations utilisées dans les modules métier,
- le domaine d'administration du système : gestion des utilisateurs, gestion du paramétrage...

Chacun des modules remplit une fonction particulière qui lui est propre. L'ensemble conduit à gérer la vente d'un article à un client, et à recevoir en contrepartie un paiement enregistré dans la comptabilité... Le processus se met en œuvre par une série de manipulations des acteurs

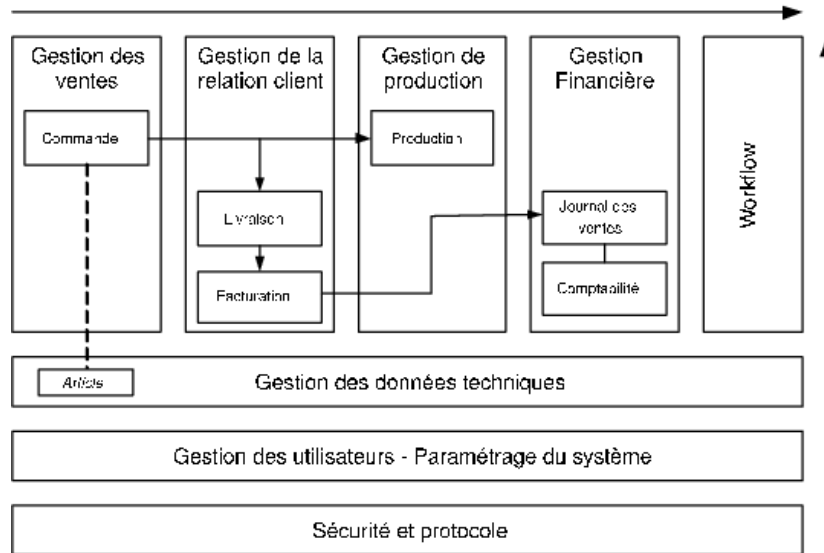


FIG. 5.1 – Représentation macroscopique du processus sur l'architecture logique

métier sur cette suite de modules dans un sens déterminé par les flèches qui symbolisent le flux d'informations entre les fonctions (entrée, sortie).

Cette représentation permet de préciser les modules éligibles et leur interaction pour la réalisation d'un travail donné. Elle définit le périmètre de l'application nécessaire pour le processus concerné. Elle est très orientée vers les fonctions, mais ne laisse pas de place aux utilisateurs, à leur logique de travail propre, aux documents produits, et même à la description plus détaillée des informations nécessaires pour chacune des fonctions. C'est pourtant le cœur même de la logique de fonctionnement de l'entreprise qui repose sur une notion de scénario où le processus est « mis en scène » au moyen de l'ERP. La représentation de la figure 5.1 est donc complétée par la représentation graphique de la figure 5.2. On y retrouve les éléments suivants :

- les événements déclencheurs des activités (réception devis, réception offre, validation...),
- les activités au cœur du processus (devis, offre de prix, commande...),
- les documents générés par chacune des activités (édition devis, édition commande, édition journal...),
- les informations utilisées ou produites par une activité (article, transporteur...),
- les acteurs intervenant dans chacune des activités par les rôles prévus dans l'application (administration des ventes, logisticien, comptable...).

À ce stade de définition, l'utilisation de l'ERP pour implanter le processus est suffisamment complète. C'est donc bien une modélisation de processus calquée sur la connaissance de l'architecture logique qui sert de base à la transition du point de vue fonctionnel au point de vue logique. Mais cette représentation dépasse le périmètre strict de la logique applicative pour intégrer l'environnement immédiat : acteurs, événement externe, document, ...

Cette approche a été reconnue depuis quelques années et a été appliquée avec succès à quelques cas. Elle est mise en œuvre par deux voies principales :

- les éditeurs d'ERP ont développé des modèles de référence de leur application. Il s'agit en général d'une base de données qui contient des représentations graphiques des différents traitements disponibles. Il est possible de les assembler sur une palette graphique, puis de les modifier afin de créer les interactions entre les traitements, et les modules. La sémantique du langage utilisé pour cet assemblage est plus ou moins évoluée selon

les éditeurs. Il n'y a pas de normes sur les modèles de référence permettant de s'affranchir de l'aspect propriétaire. En phase de sélection du projet ERP, il faudrait donc théoriquement bâtir autant de modèles qu'il y a de candidats.

- Les éditeurs du marché de la modélisation des processus ont naturellement un rôle à jouer dans cette approche. Chaque éditeur propose un langage propre pour décrire les processus d'entreprise. En général, la sémantique de ces langages est plus puissante que celle utilisée nativement pour les modèles de référence dans les ERP. Partant de ce capital de connaissances, ils proposent de l'utiliser à des fins de définition, puis de maintenance du Système d'Information. Des fonctions spéciales sont développées pour trouver les relations entre ce modèle et celui de référence d'un ERP particulier. Mais les lois du marché et les contraintes économiques sont telles que seul SAP est présenté comme un partenaire dans ce cadre. Nous n'avons pas trouvé trace de développement couvrant une gamme de produits plus large...

Il y a donc un manque d'universalité dans la mise en œuvre de la démarche lorsqu'on prend la position du maître d'ouvrage et que l'on souhaite connaître et mesurer le potentiel d'une solution ERP donnée.

Il nous semble tout à fait intéressant, sur le plan académique, de pousser l'étude pour déterminer de manière claire le pouvoir d'expression requis pour le langage de modélisation, qui dépasse peut être un simple modèle de processus. Mais le choix d'une représentation reste délicat car il est guidé par deux positions antagonistes :

- dans l'optique d'un dialogue entre MOA et MOE cherchant la mise en correspondance de l'architecture fonctionnelle avec l'architecture logique, nous devons aborder la question de l'acceptation de ce langage par les deux parties présentes, comme un moyen de communication efficace (l'équivalent des plans de l'édifice produits en architecture). Ceci signifie que le modèle de référence de l'ERP doit pouvoir être traduit dans ce langage.
- Dans un souci de respect de la cohérence entre le modèle de référence et l'application proprement dite, le langage doit être aussi proche que possible des langages mis en œuvre dans les processus de développement du génie logiciel, et donc par les équipes en charge du développement et de la maintenance de l'application. Cette dernière contrainte a pour but de minimiser l'effort de gestion des versions (et des configurations d'une même version) du modèle de référence. Enfin, les mécanismes de gestion de connaissances permettant de passer du modèle du besoin au modèle de référence, et vice versa, doivent être imaginés.

5.2.3.2 Point de vue physique de l'architecture

La figure 5.3 représente un exemple d'architecture physique de type client/serveur. Le cylindre de droite représente le stockage des données sous forme d'un ensemble de base de données structurées par thème (client, article, besoins, commande, production, ...). La partie client (rectangle de gauche) représente des traitements pris en charge par le poste client (client lourd). Un accès direct aux bases de données concernées est matérialisé par des flèches dont le sens désigne s'il s'agit d'une extraction ou d'une écriture d'information. Placé au milieu, le poste serveur est chargé des traitements consommateurs de temps de calcul, comme un calcul des besoins en composants dans un MRP, par exemple. Cette figure n'illustre pas le cas où un module serait distribué pour partie sur le client et pour partie sur le serveur, mais il existe. Dans ce cas, il est nécessaire de faire apparaître la couche de présentation, la couche de logique de communication entre client et serveur, la couche de traitements (logique métier) sur le serveur.

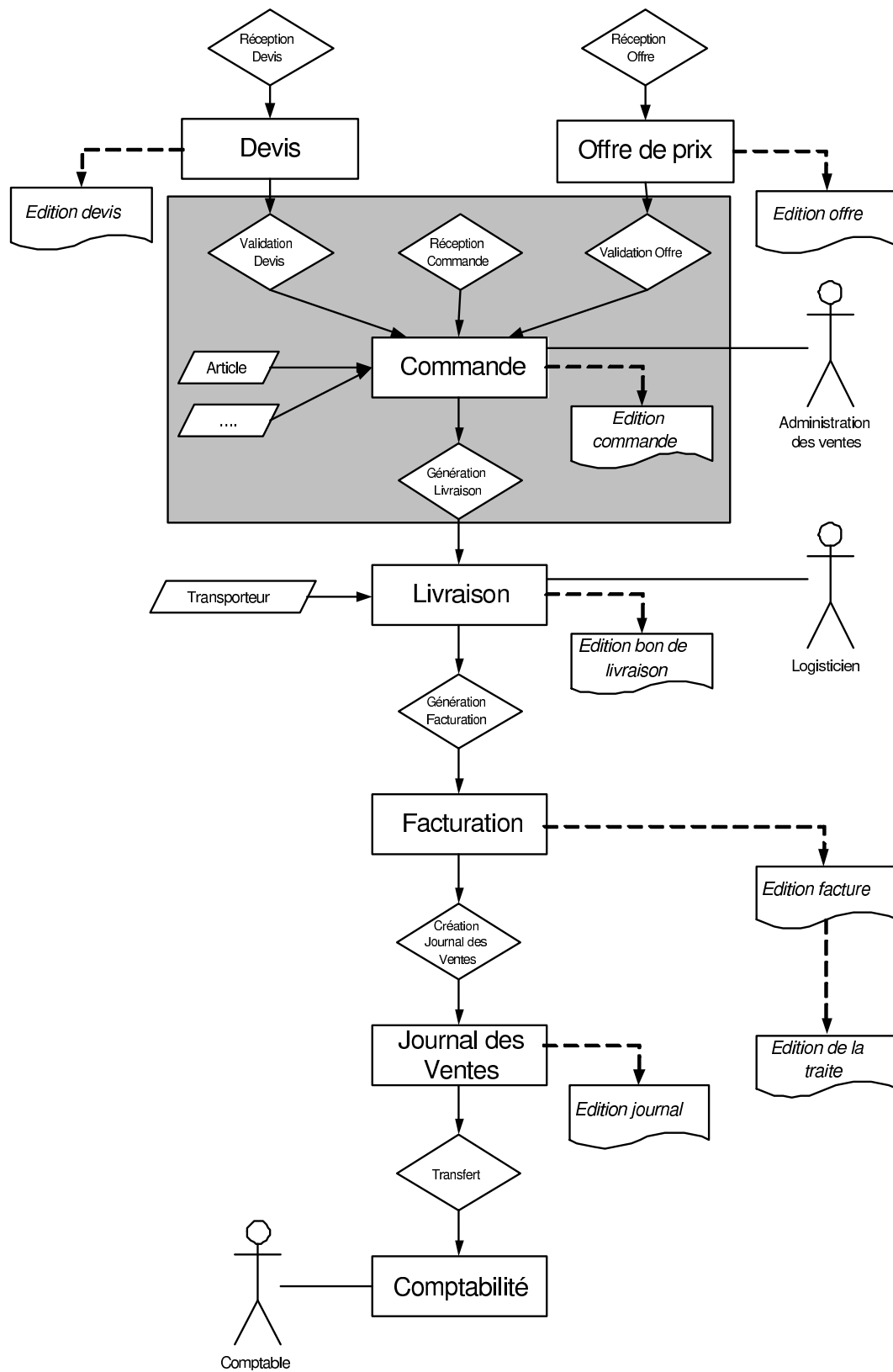


FIG. 5.2 – Enchaînement d'activités - Processus de « Ventes »

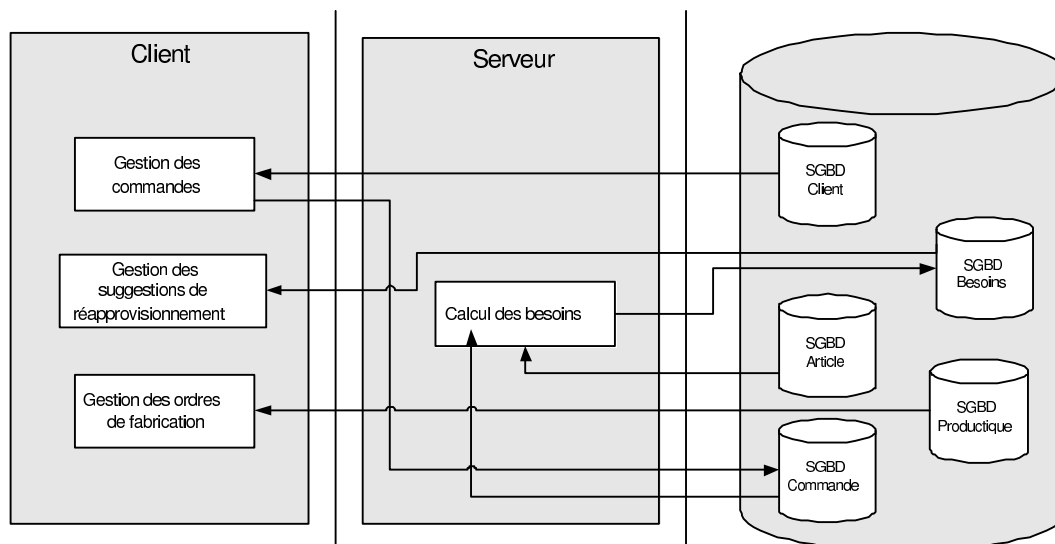


FIG. 5.3 – Architecture physique

5.3 Nouvelles tendances en architecture logique

Comme le souligne Robert Reix (2002), l'interopérabilité est la capacité de différents systèmes informatiques à coopérer pour la réalisation en commun de traitement. L'interopérabilité traduit, de façon concrète, un certain degré de compatibilité entre les applications. Rendre possible l'assemblage de composants (sous-systèmes) de différentes sources (tout ou partie de modules de plusieurs fournisseurs d'ERP, par exemple) implique d'intégrer la notion d'interopérabilité comme une exigence dès les premières étapes de la conception des dits composants. L'interopérabilité doit permettre de rendre les Systèmes d'Information plus efficaces (Michel Schneider, 2001).

Il y a encore peu de temps, la construction de ce type d'architecture logicielle était d'un coût très élevé, dû à la nécessité de développer des passerelles pour rendre effective la compatibilité entre les applications. De plus, la pérennité de ce genre de passerelles suite aux mises à jour des applications ne pouvait être garantie du simple fait de l'absence de protocoles standard de communication, réduisant l'intérêt de ce type de pratique.

Différentes approches architecturales tendent à réduire cette lacune, en imposant ces nouveaux standards de communication. Nous pouvons citer le principe de l'urbanisation des systèmes (Gérard Jean, 2000), les concepts de l'EAI (Laurent Avignon et al., 2000) ou encore les services Web (Hubert Kadima, 2003).

La mise en œuvre de ces principes appelle des choix techniques dans le domaine de la communication directe d'applications par les réseaux (middleware), en cela elles touchent les points de vue logique et physique. Mais quelle que soit la solution technique choisie, il est indispensable de disposer d'un modèle, cartographie des processus, décrivant les interactions entre les fonctions utilisées dans les différents modules déployés pour maîtriser l'architecture du système cible et synchroniser les échanges de différentes natures. Il faut donc insister sur le fait que ces modes basés sur une communication accrue entre des sous-systèmes distincts posent l'usage d'un modèle de représentation comme un prérequis à la conception, et pas comme un soutien éventuel.

5.3.1 EAI

Le sigle EAI (Enterprise Application Integration) pourrait se traduire en français par Intégration des Applications d'Entreprise. Le but de l'EAI est d'instaurer une architecture basée sur des échanges entre des applications, principalement au sein d'une même entreprise. L'intégration d'application permet de faire dialoguer des systèmes, souvent hétérogènes, entre eux.

L'EAI doit remplir 4 fonctions :

1. interfacier : il doit être capable d'extraire et d'injecter des données dans une application. Il s'agit de la création, spécification et description de l'interface de l'application avec laquelle nous allons dialoguer ;
2. transformer : il doit être capable de convertir les données. Grâce à un moteur de transformation, il est possible de convertir des informations produites par une application en informations acceptées par une autre application. Cette fonction porte le nom de mapping en utilisant un format « pivot » ou « fédérateur » (XML) qui joue le rôle de traducteur ;
3. router : il doit être capable de transporter les données jusqu'au destinataire. Pour cela, on utilise (s'appuie sur) des protocoles du middleware (SOAP, CORBA, etc.) qui sont utilisés pour réaliser des communications synchrones ou asynchrones entre applications.
4. gérer : il doit pouvoir suivre l'état du processus. Lors de l'échange de données entre deux applications, un processus permet de valider les différentes étapes du transfert. C'est souvent un moteur de workflow qui assure cette fonction.

Les connecteurs de solution logicielle doivent permettre de réduire l'effort nécessaire d'intégration de plusieurs solutions logicielles du Système d'Information. Les connecteurs fournissent deux types de services :

- la transformation des données aux formats manipulés par les solutions logicielles,
- la relation aux interfaces spécifiques de type API (Application Program Interface) de la solution logicielle.

Le contrôle des échanges utilise la notion de « trigger », c'est-à-dire la capacité de déclencher un programme sur l'apparition d'un événement interne particulier, sensibilisant un processus à un état donné.

5.3.2 Urbanisation des Systèmes d'Information

Le concept d'urbanisation d'un SI repose sur une volonté d'anticipation de l'évolution architecturale : un découpage et des grands principes de construction qui permettront de faire évoluer le Système d'Information et l'informatique au même rythme que la stratégie et l'organisation de l'entreprise.

*« Urbaniser son Système d'Information consiste à en dresser la cartographie de manière à le dessiner en termes de blocs fonctionnels et de flux - d'où le parallèle avec la ville, ses quartiers, ses artères, etc. But du jeu : rendre le Système d'Information assez flexible pour suivre les évolutions de la stratégie d'entreprise. »
(Gérard Jean, 2000)*

Les principaux concepts émanant de l'urbanisation du Système d'Information sont :

- de donner de la visibilité et de la souplesse à l'évolution du Système d'Information (principe de modularité),

- de mutualiser les grandes infrastructures (principe d'économie),
- de garantir une cohérence d'ensemble (principe de cohérence),
- de répartir les responsabilités entre les différents niveaux d'acteurs de l'entreprise en favorisant la prise de décision au plus près du terrain (principe de subsidiarité).

Il s'agit principalement de fournir une nouvelle valeur au système. Il ne s'agit pas de refaire une implémentation, mais une extension du système existant. Il est souhaitable de réutiliser l'existant (le passé) s'il répond aux besoins des utilisateurs, les applications seront reliées par une plate-forme d'échange, idéalement l'EAI. L'évolution du système en alignement stratégique avec l'organisation future amène son lot de réformes et impacte l'architecture. Pour ces évolutions, il est souhaitable d'utiliser une approche par composants. Et pour répondre aux exigences d'intégration, ces composants seront connectés à la plate-forme et devront dialoguer avec les anciennes applications (XML se taille la part du lion pour la structuration des messages échangés). Il y a un souci de ne pas rompre avec le passé par des fractures technologiques portant sur la totalité du système, mais de rentrer dans une démarche d'amélioration continue en introduisant progressivement les nouvelles technologies.

5.3.3 Services WEB

Les services Web sont des composants logiciels encapsulant des fonctionnalités métiers de l'entreprise et distribués, c'est-à-dire accessibles via des protocoles standard du Web. Le service est donc mis à disposition sur un site pour utilisation depuis n'importe quelle plate-forme de développement.

D'un point de vue technologique, il s'agit d'offrir la possibilité d'interconnecter des applications via le Web. Ce sont des standards (XML, SOAP, WSDL, UDDI) qui permettent de décrire les interfaces de communication en essayant de masquer la complexité des technologies middleware (Corba, DCOM, Java/RMI).

Ce type de solutions se développe dans le cadre des applications pour les affaires du grand public (tourisme, transport, ...), il y a peu de développement orienté vers le monde de la gestion des entreprises pour l'instant.

5.4 Nouvelles tendances en architecture physique

Pour résoudre les inconvénients du modèle client/serveur, dépendance trop forte entre le client et le serveur, répartition des traitements sur le client, le modèle 3-tiers est proposé. Le premier tiers se charge de la présentation, le troisième tiers se charge de la persistance des données, tandis que le nouveau tiers, intermédiaire, est en charge du traitement des informations. Nous synthétisons sur la figure suivante (fig. 5.4) une représentation de ce nouveau modèle.

Lorsqu'il y a un serveur Web dans cette architecture, celle-ci peut aussi prendre le nom d'un modèle 4-tiers. Dans un souci de lisibilité, nous resterons sur la dénomination 3-tiers.

Les différentes couches du modèle jouent les rôles suivants :

- présentation (1-tiers) : le poste client devient donc un poste léger. Il a en charge la présentation des interfaces et le dialogue avec le serveur de traitement. Généralement, on utilise un navigateur Web ;

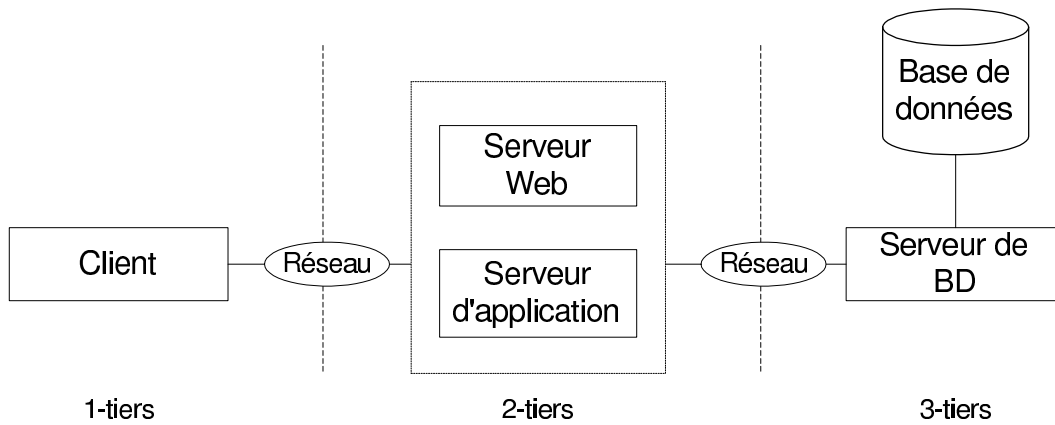


FIG. 5.4 – Modèle 3-tiers

- traitement (2-tiers) : le serveur de traitement est en charge de deux activités. L'une correspond à la génération de l'interface utilisateur tandis que l'autre correspond au traitement des informations. Il existe donc un dialogue entre le serveur de traitement et le client, mais aussi entre le serveur de traitement et le serveur de données ;
- persistance (3-tiers) : le serveur de base de données assure principalement la gestion des données dans la base.

Nous détaillerons dans la prochaine section les caractéristiques de ces différentes couches (§ 5.4.1). L'avantage de ce type d'architecture est la relative indépendance des couches. Avec le modèle 2-tiers, une modification d'une couche pouvait engendrer des coûts très importants pour préserver la logique de communication entre les parties. Avec une architecture 3-tiers, seules les interfaces entre les différentes couches sont touchées par une modification. Un changement du côté du serveur perturbera moins le bon fonctionnement du poste client.

5.4.1 Un exemple d'architecture n-tiers avec la technologie J2EE

Dans cette section, nous souhaitons illustrer une architecture n-tiers basée sur une technologie Java (J2EE). Le schéma suivant (fig. 5.5) présente l'organisation générale de ce type d'architecture. Il montre bien les effets sur l'architecture logique d'une évolution de l'architecture physique. Le fonctionnement repose sur des communications organisées entre des parties distribuées sur plusieurs machines et qui remplissent de concert les fonctions pour lesquelles elles sont programmées (saisie de données, édition d'interface, connecteur de base de données, connecteur d'application, ...).

Ce type d'architecture apporte les avantages suivants :

- indépendance de la plate-forme de déploiement : un des avantages de Java est la possibilité d'installer une application sur des plates-formes systèmes différentes (Windows, Linux, Unix) sans avoir à modifier le code ;
- utilisation d'un poste léger côté client : ce type d'architecture envoie des pages Web vers le navigateur du client. Il n'y a plus d'application lourde réalisant des traitements ;
- utilisation d'un cadre (framework) de conception : on utilise un « Modèle Vue Contrôleur » (MVC), un guide défini par des architectes comme une « bonne pratique » :
 - le modèle s'applique au serveur d'application. Il se charge des principaux traitements du système. Il correspond à une vision métier ;

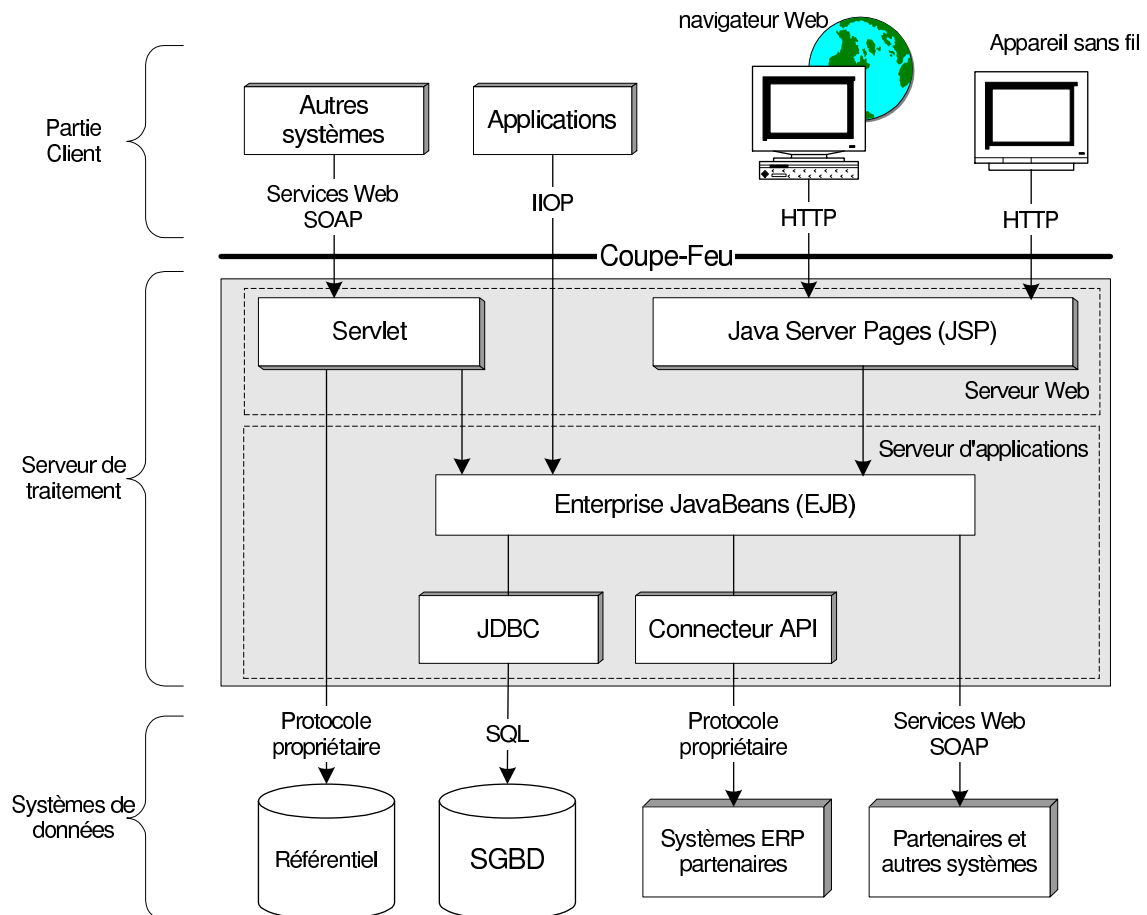


FIG. 5.5 – Architecture logique d'un système basé sur J2EE

- la vue correspond à la production des interfaces. Un serveur de type JSP se charge de générer les pages Web qui vont être affichées sur le navigateur ;
- le contrôleur se charge de la vérification des données qui vont être transférées entre la vue et le modèle. Il s'agit d'un servlet sur l'exemple ci-dessus...

5.5 Synthèse

L'architecture de Système d'Information n'est pas un art facile. Elle doit évoluer suivant les exigences des consommateurs/utilisateurs, toujours croissantes, et puiser dans les nouvelles technologies qui imposent une cadence rapide d'innovation. L'exercice de l'art implique d'accepter des sauts technologiques, et de savoir les négocier, dans un climat de « tout communicant ». Les mots clé de processus, de modèles, d'interopérabilité sont au cœur de la pratique actuelle de cet art et sont devenus des pièces maîtresses des tendances comme l'EAI, l'urbanisme, et les Web services. Toute expérience relative à cette architecture de Système d'Information, et donc en particulier d'ERP, doit tenir compte de ces sources de progrès sous peine d'obsolescence rapide.

En termes de démarche de conception, le principe de « componentisation » (David Sprott, 2000) est commun à beaucoup de disciplines d'ingénierie. Que ce soit des fabrications d'automobiles, d'ordinateurs ou encore des téléphones portables, ils sont assemblés à partir d'un ensemble de composants, selon une approche modulaire, composants qui collaborent dans une

architecture commune dite intégrée. Les solutions architecturales existent maintenant pour qu'il trouve une application dans le domaine qui nous intéresse. Kuldeep Kumar et Jos van Hillegersberg (2000) ont proposé de concevoir un ERP à base de composants afin de répondre aux problématiques de variété et de complexité des fonctionnalités. Comme le dit David Sprott, cela doit se faire avec le souci d'améliorer les quatre critères suivants :

- intégration,
- adaptabilité,
- facilité de mise à jour,
- facilité de mise en œuvre.

Parmi ces quatre critères, celui sur la facilité de mise en œuvre ramène à la relation entre le point de vue fonctionnel et les autres points de vue de l'architecture. Et cela ne fait que renforcer l'idée d'une meilleure utilisation du modèle à tous les niveaux.

Si toutes les technologies sont présentes pour répondre à la majorité des exigences du client, l'architecte doit faire des compromis entre pas de communication et trop de communication. Parmi ces compromis¹, nous citerons :

- **cohérence vs. performance** : la gestion de la cohérence est une partie sensible d'un SI. L'information est saisie une seule fois et elle est utilisée par plusieurs modules. Il est nécessaire de s'assurer que cette information soit viable. Pour cela, nous pouvons mettre en place une surveillance par des règles permettant de tester si l'information est valide. Il est clair que plus il y aura de règles, plus la performance du système sera dégradée ;
- **sécurité vs. ouverture** : la mise en place d'une sécurité est quelque part contradictoire avec la volonté d'ouverture du système. Afin de faire ces ouvertures avec un maximum de garantie, il est nécessaire d'engager plus de moyens pour assurer les contrôles d'accès ;
- **sécurité vs. coûts** : la mise en place d'une sécurité, pour authentifier les utilisateurs, crypter les flux d'informations, a des coûts d'administration et de réseaux.

¹<http://www.rd.francetelecom.fr/fr/conseil/mento14/chap4c.html>

Chapitre 6

Ingénierie des Systèmes d'Information

6.1 La gestion des projets en génie logiciel

Nous avons présenté en première partie du mémoire, au chapitre 4, la gestion des projets ERP en entreprise. Nous abordons, avec ce chapitre, un autre pan du sujet qui porte sur la gestion des projets de Système d'Information. Les projets ERP sont des projets de Système d'Information particuliers reposant sur la décision de déployer une solution préexistante. Mais on ne peut effectivement pas ignorer la manière dont les ERP sont construits par les éditeurs si on cherche à avoir accès à des connaissances sur ces produits. En effet, comme nous l'avons déjà dit, en particulier sur le choix de langage de modélisation, la synergie recherchée entre différentes formes d'acteurs (développeurs, intégrateurs, utilisateurs) implique de comprendre leur méthode de travail, afin de tirer les meilleurs avantages d'une mise en commun. C'est donc la façon dont les équipes de développement qui ont la responsabilité de produire de nouvelles version d'ERP représentent leur application qui est au cœur de notre propos ici. Le thème est large, autant que celui de l'architecture...

6.1.1 Les critères de qualité du logiciel

La qualité peut être mesurée à partir de critères de jugement définis dans le Plan Qualité Logiciel (PQL). Ces critères sont la base d'un consensus au sein d'une population variée :

- les utilisateurs qui ont une vision externe du produit informatique, mais qui sont la référence en termes d'adéquation du produit aux exigences métiers,
- les informaticiens qui sont en charge du développement, mais aussi de la maintenance et de l'évolution du produit,
- les équipes projet en charge de la définition du besoin, du déploiement et de la mise en œuvre du produit.

Bertrand Meyer (1988) a classé les critères en deux groupes : les facteurs externes portant sur la valeur d'usage vue par des utilisateurs, et les facteurs internes du ressort des concepteurs.

Les facteurs externes sont :

- la justesse qui correspond à la capacité du logiciel de réaliser leur fonction de façon exacte, comme définie par les besoins et spécifications,
- la robustesse qui correspond à la capacité des logiciels à fonctionner même en conditions anormales,

- l'extensibilité qui correspond à la facilité avec laquelle les logiciels peuvent être adaptés aux changements de spécifications,
- la compatibilité qui correspond à la facilité avec laquelle des logiciels peuvent être combinés avec d'autres,

tandis que les facteurs internes sont :

- la modularité qui est un facteur clé pour le critère d'extensibilité,
- la lisibilité qui correspond à la compréhension des spécifications et du code,
- la réutilisabilité qui correspond à la capacité des logiciels à être réutilisés, dans leur globalité ou en partie, pour de nouvelles applications,
- la facilité de maintenance directement corrélée aux facteurs externes : la justesse, la robustesse, l'extensibilité et la compatibilité.

Nous y ajoutons d'autres facteurs tels que :

- l'efficacité qui correspond à une bonne utilisation des moyens pour obtenir le résultat,
- la portabilité qui détermine le degré d'indépendance du logiciel vis-à-vis du matériel,
- l'intégrité que nous avons déjà définie,
- la facilité d'usage ou convivialité.

Nous proposons une synthèse des différents critères de jugement sous forme d'une arborescence sur la figure 6.1. Nous avons effectué ce classement des critères par niveau, et relié les critères.

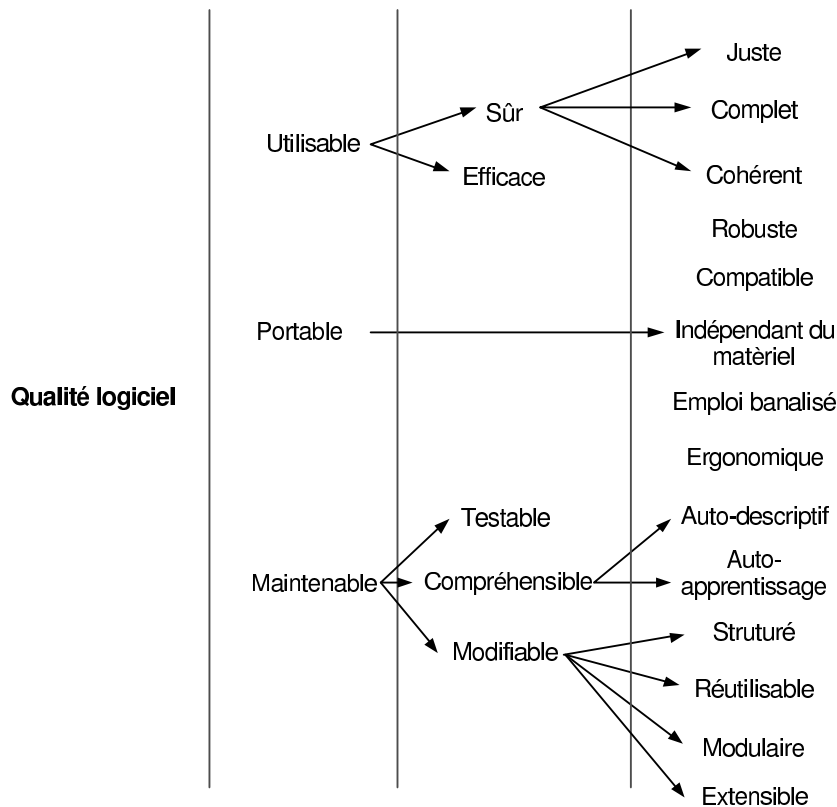


FIG. 6.1 – Critères de jugement de la qualité du logiciel

L'obtention de la qualité sensibilise toute la chaîne de production du logiciel, depuis la définition du produit jusqu'à la mise en service et l'exploitation. La palette de critères couvre tout le cycle de vie du logiciel. Là encore, comme dans toute démarche qualité, l'existence d'un référentiel qui permette de porter un diagnostic sur les éléments contribuant à la satisfaction

de ces critères, est un point clé. Il n'est donc pas étonnant que beaucoup de démarches de projet revendiquent un soutien par la modélisation pour constituer ce référentiel.

Si nous mettons ici en avant la qualité comme principal élément de jugement de succès du projet, il ne faut pas oublier le respect des coûts et des délais, deux facteurs qui sont étroitement liés dans ce type de projet où le coût de main d'œuvre est élevé. Toutefois, cette efficacité des équipes projets en charge du développement sera reléguée au second plan, dans cette partie, du fait de notre centre d'intérêt.

6.2 Méthodes de conception

Depuis une vingtaine d'années, les cycles de développement des Systèmes d'Information ont fortement évolué vers des démarches plus itératives et incrémentales afin de tenir compte plus correctement du point de vue de la maîtrise d'ouvrage (et en particulier des utilisateurs). Il est en effet indispensable que l'utilisateur puisse vérifier au plus tôt que le résultat de la conception, même partiel, correspond à ses besoins. C'est le critère de justesse évoqué plus haut. On touche là un point essentiel du dialogue MOA/MOE, et des relations entre architectures fonctionnelle et logique. De nombreuses méthodes (et parfois les outils aidant à leur mise en œuvre) sont proposées pour accompagner ces cycles de développement, et parfois au sein même des outils de développement du génie logiciel. Dans sa philosophie, la méthode cherche souvent à transformer une spécification floue, et réduite, en une description complète, consistante et compréhensible par les différentes parties.

6.2.1 Cycle de base des projets

On parle de cycle parce que la réalisation du travail est effectuée par des itérations sur une série des tâches élémentaires, qui forment le cycle de base. Ce cycle de développement d'applications informatiques doit permettre de :

- bien comprendre les demandes des utilisateurs finaux,
- tenir compte des changements du cahier des charges,
- empêcher la découverte tardive de défauts sérieux dans le projet,
- traiter au plus tôt tous les points critiques du projet,
- bien communiquer avec le client,
- bien maîtriser la complexité,
- favoriser la réutilisation,
- définir une architecture robuste et évolutive ou capable de s'adapter,
- faciliter le travail en équipe.

Nous pouvons distinguer deux catégories de cycles : les cycles dits « classiques » et les cycles liés à l'émergence de nouvelles méthodes et à la banalisation de la programmation orientée objet.

Les principaux cycles, dits « classiques », sont :

- cycle de vie en V (Barry W. Boehm, 1981),
- cycle de vie en spirale (Barry W. Boehm, 1986),

Parmi les cycles récents, nous examinerons plus précisément le cycle USDP (Unified Software Development Process).

6.2.1.1 Cycle en V

Le principe de ce modèle de décomposition en tâches (fig. 6.2) est qu'avec toute décomposition doit être décrite la recombinaison, et que toute description d'une partie de code est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description. Des liens forts sont établis pour préparer les dernières phases (validation-vérification) à partir des premières (construction du logiciel), et permettre ainsi d'éviter un écueil bien connu de la spécification du logiciel : caractériser une exigence du client qu'il est impossible de vérifier objectivement après la réalisation (ce qui sous-entend qu'on ne peut assurer la justesse).

On distingue donc deux sortes de dépendances :

- l'enchaînement des activités : se déroule essentiellement de gauche à droite (de la définition du cahier des charges, jusqu'au codage, finissant par la recette de l'application) sachant que chacune des activités est validée et doit respecter son vis-à-vis (cahier des charges et recette, spécifications générales et tests d'intégration, ...);
- la préparation des phases ultérieures. Pour passer à l'activité suivante, l'activité en cours doit être complètement terminée.

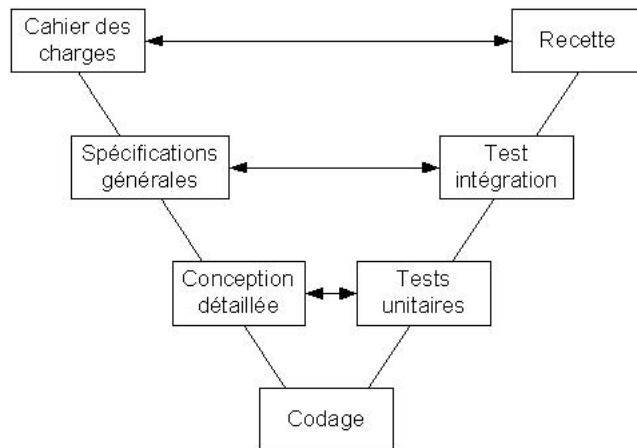


FIG. 6.2 – Cycle en V

Le cycle en V permet de mettre en correspondance les activités du début de cycle (phase descendante jusqu'au code) et les activités de fin de cycle (phase ascendante jusqu'à la recette). Le principal inconvénient de ce cycle est l'impossibilité de prendre en compte les évolutions possibles du cahier des charges lorsque cette activité est terminée, il n'est pas possible de modifier ces spécifications en cours de réalisation des programmes. Enfin, un autre défaut de ce cycle est la difficulté de proposer, de faire tester ou de montrer à l'utilisateur (le client) des maquettes ou des prototypes lors des premières activités du cycle.

6.2.1.2 Cycle en spirale

Proposé par Boehm en 1986, ce modèle (fig. 6.3) est beaucoup plus général que le précédent. Il tire profit d'une démarche d'analyse des risques dont la prise en compte induit un caractère itératif dans le déroulement des tâches. Chaque cycle de la spirale se déroule en quatre phases :

1. détermination, à partir des résultats des cycles précédents ou de l'analyse préliminaire des besoins, des objectifs du cycle, des alternatives pour les atteindre et des contraintes;
2. analyse des risques, évaluation des alternatives et, éventuellement maquettage;

3. développement et vérification de la solution retenue, un modèle « classique » (cascade ou en V) peut être utilisé ici ;
4. revue des résultats et vérification du cycle suivant.

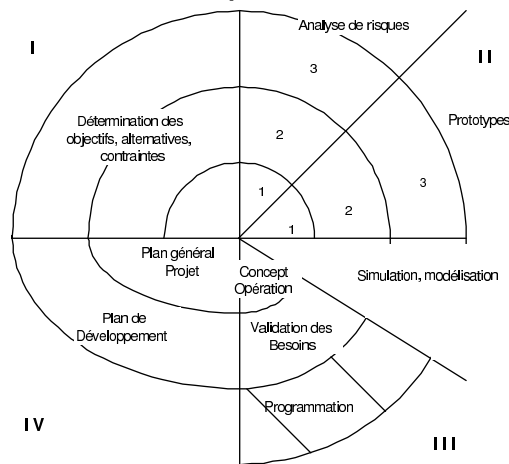


FIG. 6.3 – Cycle de vie en spirale

Le cycle en spirale est basé sur la mise à disposition de la MOA d'un prototype évolutif, exécutable et donc évaluable. À chaque évolution du prototype, il est possible d'identifier les problèmes, et de prendre en compte les corrections nécessaires lors de la prochaine itération. Les risques sont alors identifiés, évalués et gérés beaucoup plus tôt dans le projet. Cette démarche que l'on peut assimiler à des essais/erreurs, apporte une contribution progressive à la qualité de l'expression des besoins en l'étalant sur une période plus longue du projet. Mais cette évaluation permanente autour des prototypes impose une communication forte entre l'ensemble des acteurs du projet (client, utilisateur, développeur...), afin de vérifier et de valider le prototype à chaque itération. Il faut un effet mémoire, ou capital de connaissances, sur le produit de la part des équipiers, qui ait une certaine pérennité, pour assurer une progression efficace.

6.2.1.3 Cycle émergent

Au cours des années 90, de nouvelles méthodes de conduite de projet ont vu le jour, sous l'impulsion des technologies orientées objet qui connaissaient un succès notable. La méthode USDP est une des premières méthodes de développement qui utilise explicitement le modèle comme trait d'union (les diagrammes d'UML).

L'approche d'une solution obéit aux principes du cycle en spirale. Des tâches d'un cycle de base sont répétées pour acquérir une connaissance progressive dans l'espace de formulation du problème, et dans celui de la proposition de solutions. Les différents états d'avancement du projet sont modélisés par quatre temps (*phases* en anglais) : de la modélisation du contexte et de la capture des besoins jusqu'à la mise en place d'une version chez le client. A chaque temps (les colonnes « phases » de la figure 6.4), la charge de travail du cycle de base se distribue différemment entre les tâches.

Nous retrouvons l'enchaînement des différentes phases de USDP dans la figure fig. 6.4.

L'objectif de cette nouvelle méthode est d'améliorer les points suivants :

1. un développement itératif du logiciel : il est important de montrer à la MOA l'état d'avancement du projet, des livrables. Ceci doit permettre de mieux maîtriser les risques inhérents au développement en les levant au plus tôt ;

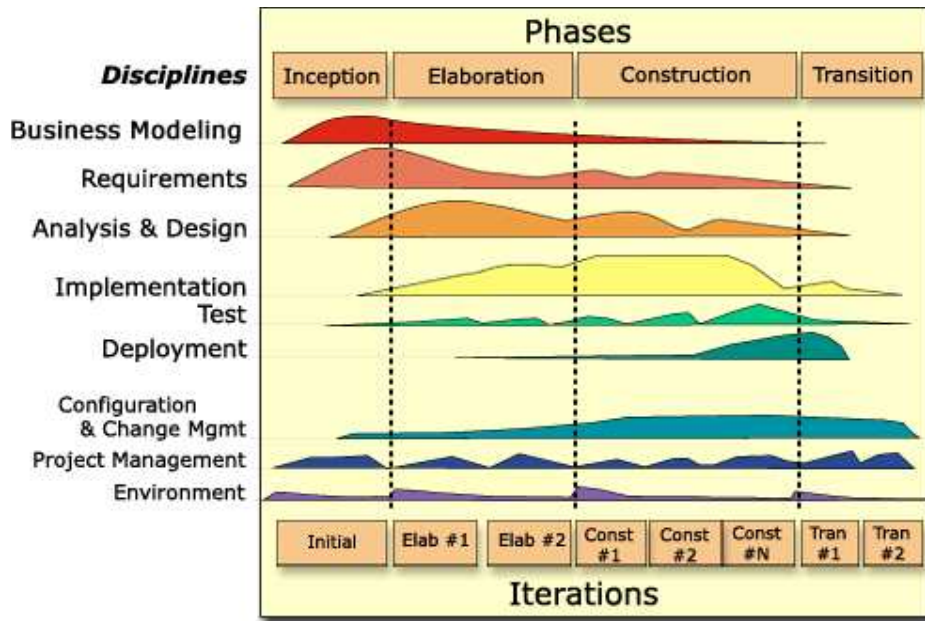


FIG. 6.4 – Démarche projet : méthode USDP

2. gérer les exigences : toutes les fonctionnalités du résultat souhaité doivent faire partie d'une ou plusieurs exigences ;
3. une architecture logicielle adaptée : il faut privilégier une construction du logiciel par assemblage de composants ;
4. adopter une modélisation graphique : il est nécessaire d'utiliser les diagrammes UML afin de mieux communiquer et de manière plus rigoureuse entre développeurs et utilisateurs ;
5. vérifier la qualité en continu : le respect des exigences doit être effectué tout au long du projet de développement ;
6. gérer les demandes de changement.

Cette méthode orchestre donc différents éléments :

- les disciplines : analyse & design, développement, tests, gestion de projet, etc.,
- les phases et les livrables de chaque phase : inception, élaboration, construction et transition,
- les acteurs du projet : analystes, développeurs, testeurs, managers, etc.,
- les artefacts : les différents modèles de document comme support de chaque discipline (modèle de document pour identifier et détailler les exigences du futur par exemple).

Le déroulement de chaque phase (inception, élaboration, construction et transition) se solde par un livrable :

- la phase d'inception établit la faisabilité du projet : à partir de la liste des exigences, nous évoquons les possibilités de réussite ou d'échecs de chacune. Il s'agit de répondre aux questions suivantes :
 - que va faire le système, définition du périmètre fonctionnel ?
 - quelle sera l'architecture générale du système ?
 - quels seront les principaux rôles au sein du projet ?
 - quels vont être les délais, coûts, moyens, ressources à déployer ?
- la phase d'élaboration s'intéresse à la capacité de production. Elle permet de :

- réaliser les premières analyses du système,
- planifier les principales tâches de la construction du système,
- éliminer les risques liés à cette phase.
- la phase de construction bâtit le système : il est temps de finaliser le développement et de vérifier la conformité du code aux exigences ;
- la phase de transition aborde l’environnement utilisateur : il s’agit de préparer les phases de déploiements, de mise en service du système chez le client.

Les deux dimensions de la méthode USDP permettent de suivre un processus intégré étayé par les modèles, les artefacts et les acteurs, afin d’obtenir le produit souhaité.

6.2.2 La modélisation

La modélisation fait partie de la démarche épistémologique de l’ingénieur (les plans dans la mécanique ou le bâtiment, des modèles mathématiques pour simuler l’écoulement d’un fluide en hydraulique, ...). La mise en œuvre de modèle est un moyen pour diminuer les risques dans la construction du système par l’expression des connaissances. Grâce au modèle, il est plus aisé de comprendre et de raisonner, donc de déceler des failles du futur système.

En Système d’Information, le modèle est une forme de représentation de l’architecture. Cette représentation en facilite la compréhension. Deux lignes directrices fortes sont à la base de ce qu’on pourrait nommer les « bonnes pratiques » de modélisation :

- la prise en compte des multiples des points de vue. Les différentes déclinaisons de l’architecture doivent être représentées afin de supporter efficacement des échanges entre plusieurs catégories d’acteurs. Il est hors de question de tout mettre à plat sur une seule représentation graphique. Chaque déclinaison (point de vue) est exprimée par une représentation graphique appropriée. Les architectures constituent une réponse à une compilation de besoins et de contraintes, qui représente un capital de connaissances sur l’ensemble du projet, et il est fondamental de pouvoir parcourir cette connaissance de manière simple et efficace. C’est donc une série de formalismes différents qui est utilisée pour la représentation dans un cadre de modélisation qui assure une cohérence d’ensemble. Le modèle se consulte en naviguant entre les différentes représentations à l’intérieur du cadre de référence.
- Le principe de structuration des connaissances dans la représentation du système : la quantité de connaissances consignée dans le modèle implique une structuration dans la construction du modèle. Les approches orientées objet ont cette propriété de permettre une représentation générique (par le méta-modèle) « à géométrie variable », c’est-à-dire pouvant être raffinée jusqu’au degré de granularité souhaité. Chaque élément du modèle peut être décrit avec plus de détail, sans surcharger inutilement la représentation.

6.2.2.1 Les cadres de référence pour la modélisation de Système d’Information

Il existe de nombreuses approches méthodologiques pour ces cadres de référence. Rolande Marcinak et Frantz Rowe (1997) proposent la classification suivante :

- les méthodes fonctionnelles (structurées ou cartésiennes) fondées sur un principe de décomposition modulaire de système et de hiérarchies de représentation,
- les méthodes systémiques : séparation des données/traitements, niveaux conceptuels, niveaux organisationnels,

- les méthodes objets.

Les méthodes fonctionnelles adoptent généralement une approche de décomposition descendante et progressive. Les modèles représentent des diagrammes de flux de données d'un système vu par les fonctions qu'il doit remplir. Le principal formalisme utilisé pour représenter ce type de modèles est le modèle IDEF0 (Icam DEFinition), plus connu sous le nom de SADT (Structured Analysis Design Technique).

Les méthodes systémiques sont apparues dans les années 1975-1980. Ces méthodes appréhendent de manière plus globale la modélisation des Systèmes d'Information et introduisent la multiplicité des points de vue. Nous citerons :

- le framework de Zackman,
- la méthode MERISE.

Les méthodes objets datent de la même époque, les cadres de référence associés sont, eux, un peu plus récents.

6.2.2.2 SADT

SADT est une marque déposée de SofTech (USA) et d'IGL Technologie (France). Cette méthode a été développée aux USA par Doug Ross en 1977 et introduite en Europe à partir de 1982 par Michel Galiner.

SADT (Michel Lissandre, 1990) permet non seulement de décrire les tâches du projet et leurs interactions, mais aussi de décrire le système que le projet vise à étudier, créer ou modifier, en mettant notamment en évidence les parties qui constituent le système, la finalité, le fonctionnement de chacune, et les interfaces entre les diverses parties qui font qu'un système n'est pas qu'une simple collection d'éléments indépendants, mais une organisation structurée pour une finalité précise.

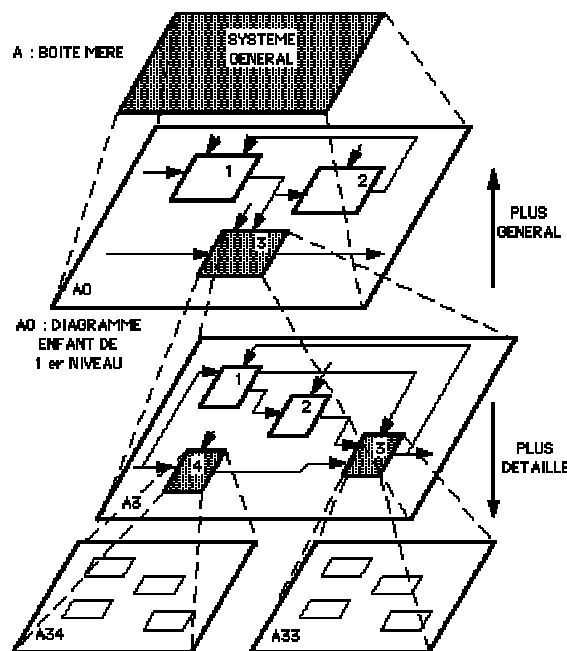


FIG. 6.5 – Décomposition hiérarchique d'un système

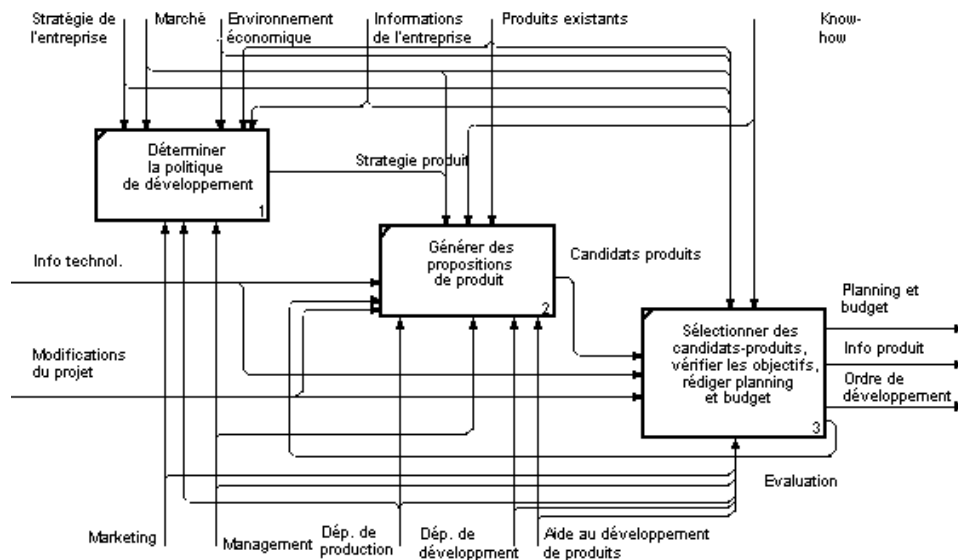


FIG. 6.6 – Enchaînement d'activités avec le formalisme SADT

Ce formalisme permet d'adopter une approche descendante, modulaire, hiérarchique et structurée d'un système. SADT décrit mal les éléments constituant le système. Par contre, il permet de comprendre ce que fait le système, les éléments entrants et les éléments sortants de chaque fonction.

6.2.2.3 Les travaux de Zachman

Le framework¹ de John A. Zachman (1987)² a été défini à la fin des années 80. Cet auteur a basé son raisonnement sur une analogie avec le métier d'architecte. Il a reconnu le caractère progressif de la définition du besoin par les acteurs avec une logique plutôt centrée sur les objectifs au départ et un niveau de détail qui se précise au fur et à mesure que l'on s'approche de la fin de l'ouvrage. Cette dimension du cadre de référence est une dimension d'abstraction des modèles qui fait passer du conceptuel à la réalité physique. Appliquant son raisonnement au Système d'Information, Zachman a affirmé que le niveau conceptuel est en relation directe avec la stratégie de l'entreprise. Sur cet axe, on recense :

- un niveau global : le contexte,
- un niveau d'entreprise : le conceptuel,
- un niveau système : la logique,
- un niveau physique,
- un niveau très détaillé : les spécifications.

De plus, ce cadre de référence conserve l'idée de différents points de vue comme une seconde dimension définissant sa structure. L'auteur en définit d'abord trois dans un premier temps, puis trois autres peu après :

- les données : le quoi,
- les fonctions : le comment,

¹On parle de framework lorsque l'ensemble de ces composants forme un squelette ou un cadre de travail prêt à être adapté ou modifié par rapport au domaine d'application. Cet ensemble constitue un moule commun à tous les modèles constituant notre système.

²<http://www.zifa.com>

- la distribution spatiale : le où,
- les acteurs : le qui,
- le temps : le quand,
- la portée : le pourquoi.

Dans ce cadre, à chaque intersection nous retrouvons une forme simple de représentation, souvent l'emploi d'un graphe à sémantique connue, décrivant le Système d'Information.

Zachman n'hésite pas à qualifier son approche « d'architecture d'entreprise », faisant l'amalgame entre le fonctionnement de l'organisation et celui du Système d'Information. Si cette mise en correspondance peut apparaître excessive, dans une certaine mesure, l'auteur anticipait un mouvement général qu'on ne remet pas en cause aujourd'hui : l'alignement du SI sur la stratégie d'entreprise passant par un modèle pour capitaliser les savoirs.































	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>	
SCOPE (CONTEXTUAL)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events Significant to the Business 	List of Business Goals/Strat 	SCOPE (CONTEXTUAL)
<i>Flavor</i>	Entity = Class of Business Thing	Function = Class of Business Process	Node = Major Business Location	People = Major Organizations	Time = Major Business Event	End/Motiv = Major Bus. Goal/Critical Success Factor	<i>Flavor</i>
ENTERPRISE MODEL (CONCEPTUAL)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Logistics Network 	e.g. Work Flow Model 	e.g. Master Schedule 	e.g. Business Plan 	ENTERPRISE MODEL (CONCEPTUAL)
<i>Owner</i>	Ent = Business Entity Reln = Business Relationship	Proc = Business Process IO = Business Resource	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	Ent = Business Objective Means = Business Strategy	<i>Owner</i>
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model 	e.g. "Application Architecture" 	e.g. "Databases System Architecture" 	e.g. Human Interface Architecture 	e.g. Processing Structure 	e.g. Business Rule Model 	SYSTEM MODEL (LOGICAL)
<i>Designer</i>	Ent = Data Entity Reln = Data Relationship	Proc = Application Function IO = User Views	Node = IS Function (Processor, Storage, etc.) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	Ent = Structural Assertion Means = Action Assertion	<i>Designer</i>
TECHNOLOGY CONSTRAINED MODEL (PHYSICAL)	e.g. Physical Data Model 	e.g. "System Design" 	e.g. "System Architecture" 	e.g. Presentation Architecture 	e.g. Control Structure 	e.g. Rule Design 	TECHNOLOGY CONSTRAINED MODEL (PHYSICAL)
<i>Builder</i>	Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	Proc = Computer Function IO = Screen/Device Formats	Node = Hardware/System Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	Ent = Condition Means = Action	<i>Builder</i>
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. "Program" 	e.g. "Network Architecture" 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>	Ent = Field Reln = Address	Proc = Language Stmt IO = Control Block	Node = Addresses Link = Protocols	People = Identity Work = Job	Time = Instant Cycle = Instant Cycle	Ent = Sub-condition Means = Step	<i>Sub-Contractor</i>
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

FIG. 6.7 – Le cadre de référence d'une architecture d'entreprise selon Zachman

6.2.2.4 MERISE

Très réputée sur le territoire français, la méthode MERISE est née vers la fin des années 1970, sous la tutelle du ministère de l'industrie, soucieux de concevoir et de définir une méthode d'intérêt national.

6.2.2.4.1 Origine

Deux courants sont à l'origine de la création de MERISE :

- L'approche systémique du SI appliquée à l'organisation d'entreprise et de son Système d'Information,

- La modélisation des données et des relations qui les lient, centrée sur l’usage des bases de données, en particulier par les travaux de E. F. Codd (1969) et Peter P. Chen (1976) dans le début des années 70, puis Hubert Tardieu et al. (1998).

À partir de ces deux fondements, un groupe de travail, placé sous la tutelle du ministère de l’industrie, s’est constitué et a entrepris une synthèse qui :

- réactualise les acquis sur la spécification des traitements,
- intègre les nouvelles approches par les données,
- propose une démarche, fruit de l’expérience, qui garantit la rigueur de la méthode et sa facilité d’application sur le terrain.

La méthode MERISE naît de ces travaux et « ambitionne de répondre aux problèmes posés par la conception d’un SI. » (Dominique Nanci et Bernard Espinasse, 1996)

6.2.2.4.2 Trois composantes de la méthode

La méthode MERISE s’inscrit dans trois dimensions exprimant :

- la démarche ou cycle de vie,
- le raisonnement ou cycle d’abstraction,
- la maîtrise ou cycle de décision.

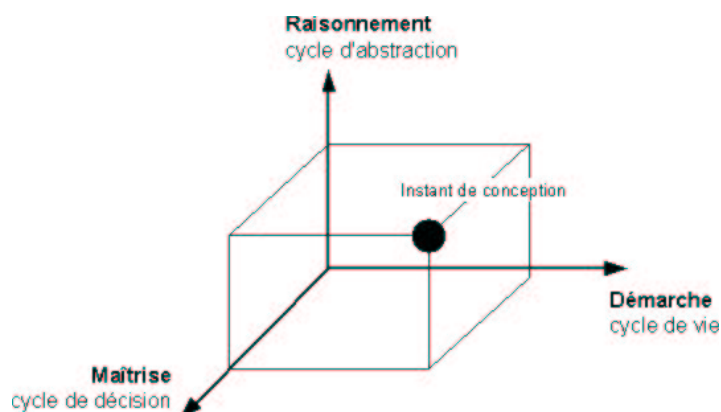


FIG. 6.8 – Repérage dans les trois dimensions

La démarche ou cycle de vie peut s’exprimer par l’exécution de différentes étapes permettant la conception, la réalisation et la maintenance du Système d’Information. C’est directement la vision de la décomposition du travail à l’intérieur de l’espace du projet :

- Tout commence par le **schéma directeur** qui définit le cadre général du système, principalement en termes d’objectifs et de contraintes. Suivant le schéma directeur, **l’étude préalable** permet d’élaborer différentes solutions et d’en évaluer les conséquences. Par la suite, une **étude détaillée** spécifie complètement le futur Système d’Information. Ces trois étapes définissent la conception du système.
- Pour décrire la partie réalisation du système, les étapes sont :
 - **l’étude technique** qui est la traduction informatique des spécifications issues de l’étude détaillée,
 - ensuite, la **production** de logiciel consiste à traduire, dans les langages appropriés, les spécifications exprimées dans les étapes précédentes,

- pour conclure, la **mise en service** consiste à installer les logiciels réalisés et à mettre progressivement l'ensemble du Système d'Information au service des utilisateurs.
- La dernière étape est la **maintenance** qui consiste à prendre en compte les évolutions apparaissant après le lancement opérationnel.

Le cycle d'abstraction permet de traiter différents problèmes (gestion, organisation, techniques, matériels, etc.). Il est dans son essence très proche de la dimension d'abstraction de Zachman. La méthode MERISE est divisée en quatre niveaux d'abstraction (tab. 6.1) :

- niveau conceptuel : exprime les choix fondamentaux de gestion ;
- niveau organisationnel : exprime les choix organisationnels des ressources humaines et matérielles ;
- niveau logique : exprime les choix de moyens et de ressources informatiques ;
- niveau physique : traduit les choix techniques.

Selon Dominique Nanci et Bernard Espinasse, les deux premiers niveaux sont adaptés à la conception du Système d'Information Organisationnel (SIO), les deux derniers à la conception du Système d'Information Informatisé (SII). Le SIO correspond plus au point de vue de l'utilisateur, tandis que le SII correspond au point de vue du réalisateur.

Le tableau 6.1 montre que le cadre de référence de modélisation par MERISE introduit deux points de vue : celui des données et celui des traitements. Les acteurs et la distribution spatiale des entités sont liés avec les niveaux d'abstraction. C'est une différence vis-à-vis des travaux de Zachman, qui est peut être compensée par le pouvoir d'expression des modèles MERISE à chaque niveau d'abstraction. Dans MERISE, les modèles sont développés pour supporter la phase d'analyse détaillée.

La maîtrise du projet ou cycle de décision est, selon Dominique Nanci et Bernard Espinasse, une succession de choix permettant de contrôler la performance du projet et de définir un système en harmonie avec les objectifs généraux de l'entreprise.

6.2.2.4.3 Transition vers l'objet

Dans les années 1990, un ensemble de langages de programmation et des méthodes de conception ont promu le concept d'objet. L'objet repose sur les notions fondamentales suivantes :

- les objets : entités concrètes ou abstraites présentes dans un univers de discours et comportant une structure statique (données) et dynamique (traitements), ainsi que des mécanismes de contrôle d'accès (encapsulation) ;
- les classes : déclaration d'un ensemble d'objets ayant même structure et même comportement ;
- les hiérarchies de classes : une classe peut être incluse dans une hiérarchie de classes et de fait, peut hériter de caractéristiques de sa (ses) surclasse(s) (attributs et méthodes) ;
- les compositions d'objets : les objets peuvent être complexes et être composés d'autres objets.

La méthode OOM (Mokrane Bouzeghoub et Arnold Rochfeld, 2000), a essayé d'évoluer vers le concept d'objet en utilisant la base principale de MERISE. Parallèlement, plusieurs courants de « pensée objet », préconisaient des approches originales et relativement complémentaires. UML est née de ces courants par un processus convergent vers un langage unique formant un cadre de référence avec, aujourd'hui, une reconnaissance internationale. Le langage UML sera présenté plus en détail dans l'annexe B.

Dans le choix d'une technique de représentation, l'approche orientée objet a retenu notre attention pour deux raisons fondamentales :

	Niveau de description	Concepts manipulés	
		Données	Traitements
SIO Système d'Information Organisationnel	Niveau Conceptuel	MCD Modèle Conceptuel de Données <i>Signification des informations sans contrainte technique ou économique</i> Concepts : objet, relation et propriété	MCT Modèle Conceptuel des Traitements <i>Activité du domaine sans préciser les ressources ou leur organisation</i> Concepts : processus, événement et synchronisation
	Niveau Organisationnel	MOD Modèle Organisationnel des Données <i>Signification des informations avec contrainte organisationnelle et économique</i> Concepts : objet, relation et propriété	MOT Modèle Organisationnel des Traitements <i>Fonctionnement du domaine avec les ressources utilisées et leur organisation</i> Concepts : procédure, phase et tâche
	Niveau Logique	MLD Modèle Logique de Données <i>Description des données tenant compte de leurs conditions et des techniques de mémorisation</i> Concepts : table et attribut	MLT Modèle Logique des Traitements <i>Fonctionnement du domaine avec les ressources et leur organisation informatiques</i> Concepts : processus, phase, fonction et module
	Niveau Physique	MPD Modèle Physique des Données <i>Description de la ou des bases de données dans la syntaxe du logiciel SGBD</i> Concepts : fichier, rubrique, objet, relation et propriété	MPT Modèle Physique des Traitements <i>Architecture technique des programmes</i> Concepts : application, unité de traitement
SII Système d'Information Informatisé			

TAB. 6.1 – Les modèles du Système d'Information pour MERISE

- la coexistence naturelle et native entre données et traitements dans les objets fournit de bonnes qualités : réutilisation des objets, maintenabilité accrue de la solution. Leur représentation par un langage standard UML diffuse progressivement dans les communautés académique et industrielle pour l'analyse et la conception,
- le cadre de référence reconnaît la notion de point de vue par neuf diagrammes permettant de couvrir un large spectre de représentation,
- la dimension de généralité est naturellement présente dans le concept d'héritage, et la méta modélisation est en passe de devenir une bonne pratique de modélisation. Ainsi, les nouvelles tendances architecturales évoquées précédemment en font un passage obligé. En particulier, un des objectifs du standard MDA (Model-Driven Architecture) de l'OMG (Open Management Group) est la possibilité de proposer une approche par modélisation indépendante du choix du langage et de la plate forme d'exploitation, et reposant sur quatre niveaux déclarés. Il s'agit d'utiliser un formalisme puissant, utilisable sur tout le cycle du projet et qui est outillé pour faciliter la description des communications. Nous ne pouvons rester insensibles à cette évolution dans le cadre de notre sujet,
- la dimension d'abstraction est prise en charge par le passage progressif de l'analyse à la conception en suivant un processus « unifié », de type USDP.

6.3 Synthèse

L'architecture d'un Système d'Information comprend trois volets :

- Un espace de description des fonctions remplies par le système, dit architecture fonctionnelle, qui est propice à la mesure de l'adéquation du système aux besoins des utilisateurs. Dans le cadre d'un projet ERP, cet espace est aussi celui de l'expression des besoins : la description des processus de l'entreprise, le rôle des différentes catégories d'acteurs, la définition et la description des ressources. C'est le terrain sur lequel se développe l'analyse.
- L'architecture logique va permettre de décrire l'organisation de l'application logicielle, c'est un espace de caractérisation du système applicatif qui doit faire écho à des exigences de traitement. Puisque l'ERP est un logiciel à large spectre de fonctionnalités, la question de la structuration de l'application est essentielle. Aujourd'hui, les architectures logiques sont centrées sur des sous systèmes principalement dédiés aux métiers, c'est le domaine de l'expertise des acteurs utilisateurs qui est le critère principal de l'architecture. Si cette orientation facilite la modularité de la couverture fonctionnelle de l'application, cela pose aussi clairement le problème de la logique d'enchaînement des traitements pour les processus à caractère transversal (qui sont nombreux), c'est à dire qui utilisent des relations d'interactions entre les métiers au sein d'une organisation. La réalisation effective de ces processus sera alors traitée par une séquence d'appels à des modules différents selon une politique de suivi des flux.
- Dans un contexte technologique où l'ouverture des systèmes par les communications est de plus en plus facilitée, la compatibilité des Systèmes d'Information peut être exploitée. Cela n'est pas sans conséquence sur les choix à réaliser en conception de systèmes. En effet, de nouvelles modes architecturales exploitent cette ouverture, et cette capacité à échanger, pour rendre l'espace des solutions applicatives plus apte à la diversité des choix. L'ERP ne peut plus constituer l'essentiel du système, mais doit pouvoir fonctionner à « voilure réduite » en s'insérant comme un sous système parmi d'autres, et qui est parfaitement intégré aux autres.

Un cheminement logique et classique consiste à utiliser les éléments de l'architecture fonctionnelle pour définir les architectures logique et physique. C'est effectivement normal de formuler un problème avant de la résoudre. Seulement, un projet ERP, fondé par un acte d'achat d'une solution préconçue de la part de l'entreprise, rend ce cheminement beaucoup moins linéaire. Il est nécessaire d'établir une relation forte entre les fonctions, les processus et les modules pour obtenir un bon degré d'adéquation. Le caractère itératif reconnu des projets de Système d'Information, et qui participe d'une démarche qualité, garde une légitimité dans les projets ERP. L'argument de la difficulté du dialogue MOA/MOE est, à ce niveau, un argument de poids...

Il est patent que les méthodes de développement de Systèmes d'Information ont évolué en qualité. Nous venons de le dire, la place de l'utilisateur au centre du projet et du système, est un premier facteur de progrès. L'usage intelligent de la modélisation pour supporter cette complexité de la démarche de conception est un second facteur de progrès.

Les cadres de référence pour le modélisation des Systèmes d'Information sont riches. Ils mettent en pratique la notion de points de vue et la déclinent par des moyens de représentations graphiques propres à chaque point de vue, avec un formalisme ad hoc. Il est ainsi possible de travailler sur les fonctions, sur les informations, sur les rôles ou sur le temps, sur la base de modèles « réduits » en accord avec le modèle général. Parcourir le modèle, c'est savoir naviguer entre les points de vue. Par ailleurs, la structuration des connaissances se fait par les deux dimensions d'abstraction (aller du concept à la réalisation en fixant progressivement les caractéristiques du produit), et de généralité (aller du général au particulier, en partant d'une base stable et condensée qui sera raffinée progressivement). Ces deux dimensions participent mutuellement à la prise en charge de la granularité évoquée dans cette partie du mémoire.

L'état de l'art en la matière est, de notre point de vue, bien traduit dans les outils et les pratiques fondés sur l'approche par objets. L'usage grandissant du langage UML permet de passer d'un savoir à un savoir-faire, qui le classe au rang d'outil de base de l'architecte. Les démarches des projets de génie logiciel sont résolument bâties autour des modèles, « basées modèles » pour traduire brutalement la philosophie MDA (annexe D).

Nous sommes convaincus qu'il s'agit d'une ouverture opportune qui offre un terrain fertile pour imaginer des liens forts entre architecture fonctionnelle de Système d'Information et architecture logique d'ERP.

Chapitre 7

La modélisation d'entreprise comme base de l'expression des besoins

7.1 Introduction

La modélisation d'entreprise consiste « en la description de l'organisation et des processus opérationnels d'une entreprise, soit dans le but de simuler ces processus pour comparer divers scénarios, soit dans le but de les analyser et de les restructurer pour améliorer la performance de l'entreprise » (François Vernadat, 1996). Elle propose un cadre formel où chaque processus est identifié, et où les interactions entre les processus sont décrites. La modélisation d'entreprise cherche à n'omettre aucun point de vue dans la représentation du fonctionnement de l'entreprise, et nous verrons que cette notion de points de vue est centrale. Grâce au modèle, nous pouvons décrire les flux d'information, de matière... ; ainsi que les interactions qui existent entre les diverses entités de l'entreprise (processus, information, acteur, ...). C'est la concrétisation des travaux sur l'approche systémique (§ 1.1) d'une part, et l'apport des sciences de gestion de production (§ 2.1) d'autre part, qui convergent vers ces nouveaux modèles des flux physiques et d'information¹.

Les objectifs et les attentes (François Vernadat, 1999) justifiant la définition d'un modèle d'entreprise sont les suivants :

1. comprendre et analyser la structure et le fonctionnement de l'entreprise,
2. prévoir le comportement et les performances des processus opérationnels avant leur implantation,
3. choisir la (ou les) meilleure(s) alternative(s) d'implantation(s),
4. identifier les risques d'implantation à gérer,
5. justifier les choix d'implantation sur des critères liés aux ressources et aux coûts,
6. bâtir une vision commune du fonctionnement de l'entreprise et la communiquer facilement au plus grand ensemble possible du personnel.

Les points 1 et 2 relèvent de la qualité dans l'entreprise, en cela, ils sont une forme de prérequis pour nos travaux. Les points 3 à 5 sont directement en phase avec le Système d'Information en tant qu'entité en charge de l'automatisation d'une partie de ces processus. Le point 6 rejoint une préoccupation de notre sujet quand on le réduit au dialogue entre les architectes de Système d'Information et le reste de l'entreprise.

¹Groupe de Travail de modélisation d'entreprise du GRP.<http://www.univ-valenciennes.fr/GRP/>

En décrivant les processus et l'organisation, ces modèles d'entreprise recouvrent donc une part des connaissances nécessaires à l'architecture fonctionnelle du Système d'Information. Il est légitime d'examiner les formalismes préconisés, et de mesurer leur pouvoir d'expression. Il est aussi essentiel de savoir quels sont les rapports entre modèle d'entreprise et modèle de Système d'Information.

Les modèles d'entreprise sont des cadres de représentation basés sur des formalismes graphiques et des syntaxes littérales. Dans les prochaines sections, nous retraçons l'historique des méthodes et modèles, ainsi que les principes sous-jacents. L'exposé détaillé sur certaines méthodes est disponible en annexe de ce mémoire (annexe C).

7.2 Processus de l'entreprise

Il existe de nombreuses définitions des processus, autant qu'il y a d'emploi de ces mêmes processus : en qualité, en réingénierie, en Système d'Information, en management, ...

Pour Patrick Burlat (2001), c'est un élément, le processus, permettant de modéliser les activités de l'entreprise auxquelles participent différentes ressources : acteurs humains, logiciels, matériels, robots et automates... Ces processus d'entreprise, très nombreux, et pour être utilisés, ils doivent être structurés en une hiérarchie de domaines fonctionnels.

D'autres auteurs, comme Mickael Hammer et James Champy, ont défini un processus opérationnel comme une suite d'activités qui contribue à la réalisation des objectifs de l'entreprise, qui produit un résultat (output) représentant une valeur pour un client (Mickael Hammer et James Champy, 1993). De manière générale, un processus peut être défini comme un enchaînement d'activités, déclenché par un événement, à exécuter pour atteindre un but donné. Cet enchaînement forme ce qu'il est convenu d'appeler le flux de contrôle du processus, c'est-à-dire sa logique d'exécution. Un événement est un fait instantané qui signale un changement d'état et nécessite une action (François Vernadat, 1999). Cette action est le déclencheur du processus. Un événement peut prendre une des trois formes suivantes :

- sollicité par un acteur interne à l'organisation,
- sollicité par un acteur externe à l'organisation,
- stimuli périodique (temporel).

L'activité est l'accomplissement d'un ensemble de tâches. Il s'agit d'une série d'opérations devant être exécutée en totalité par une ou plusieurs ressources, et en respectant des contraintes de temps d'exécution. La figure 7.1 décrit ces différents éléments ainsi que leurs relations.

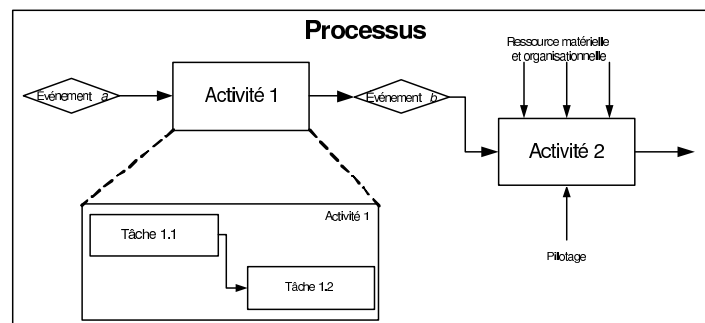


FIG. 7.1 – Décomposition d'un processus d'entreprise

Dans le cadre de l'analyse d'un Système d'Information, Chantal Morley (2002) a identifié trois types de processus :

- le processus métier a pour but d'accomplir une mission dans un domaine. Il est déclenché par un événement extérieur reconnu ou bien par le besoin périodique de produire un résultat direct en ligne avec la mission. Il met souvent en jeu plusieurs fonctions de l'entreprise. Des acteurs externes au domaine peuvent avoir une visibilité sur ce type de processus. Par exemple, pour le domaine de « Gestion commerciale », on peut définir un processus métier de gestion de commande, permettant de gérer les commandes et les livraisons ;
- le processus support n'est pas au cœur du métier : l'accomplissement de son but n'est pas une mission du domaine. Mais le résultat produit est nécessaire pour le processus métier. Il peut être déclenché par l'un des trois types d'événement. Par exemple, pour le domaine « Gestion client », on peut définir un processus de support de gestions des informations clients avec les activités de création, de mise à jour et de visualisation ;
- le processus de pilotage est déclenché par un événement interne ou temporel. Son but est d'organiser les activités de pilotage à l'intérieur d'un domaine. Par exemple, pour le domaine « Suivi », on peut définir un processus de pilotage, permettant de croiser les informations clients et commandes, afin de connaître les statistiques sur les commandes clôturées, en-cours de commandes par client, etc.

Ce terme de métier est souvent mentionné dans la littérature. Il nous semble qu'il ne prend pas toujours la même acception selon des auteurs. Tel que le perçoit Chantal Morley, le processus métier est une forme opérationnelle d'activités en droite ligne avec la notion de chaîne de valeur de l'entreprise. Cette approche est en contradiction, par exemple, avec celle de CIM-OSA où le domaine est formé de processus métiers sans aucune restriction, incorporant ainsi les processus support et de pilotage évoqués ci-dessous.

7.3 Normes de la modélisation d'entreprise

David Chen et François Vernadat (2001) ont réalisé un travail de synthèse afin de lister les organismes de normalisation, ainsi que les normes de la modélisation d'entreprise. Cette synthèse est représentée sur la figure 7.2.

Il y a trois formes d'organisations intéressées par cette activité de normalisation :

- les organismes industriels (IEEE, ISA),
- les organismes institutionnels (CEN, ISO TC) ,
- les organisations à but non lucratif (OMG, OAG, Open Group).

Les normes portent sur trois aspects :

- le modèle d'entreprise,
- les outils de modélisation/ingénierie (framework et langage),
- les technologies de l'information.

Nous avons porté notre attention sur les travaux de synthèse relatifs à un cadre de référence (framework) et à un langage pour la modélisation de l'entreprise avec les normes (ENV 40003 et ENV 12204).

La figure 7.3 est une représentation selon un ordre chronologique de l'évolution des méthodes (PERA, GRAI, CIMOSA, GERAM) vers des normes de la modélisation d'entreprise. Cette tendance à l'unification des différents acteurs de la modélisation d'entreprise au moyen d'un cadre de représentation partagé s'est concrétisée au début des années 90. C'est particulièrement vrai au niveau de l'harmonisation du vocabulaire utilisé en modélisation d'entreprise.

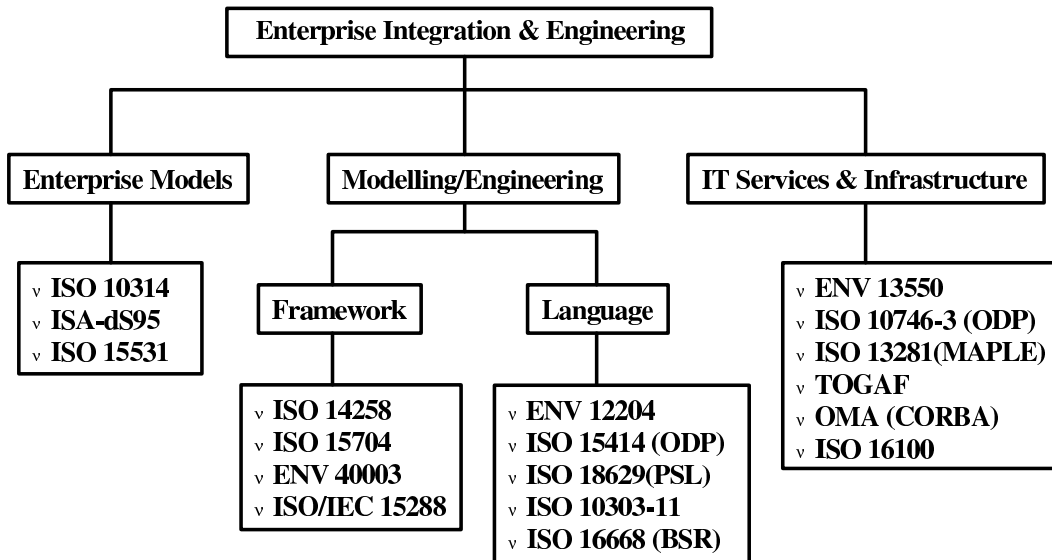


FIG. 7.2 – Classification des normes par domaine

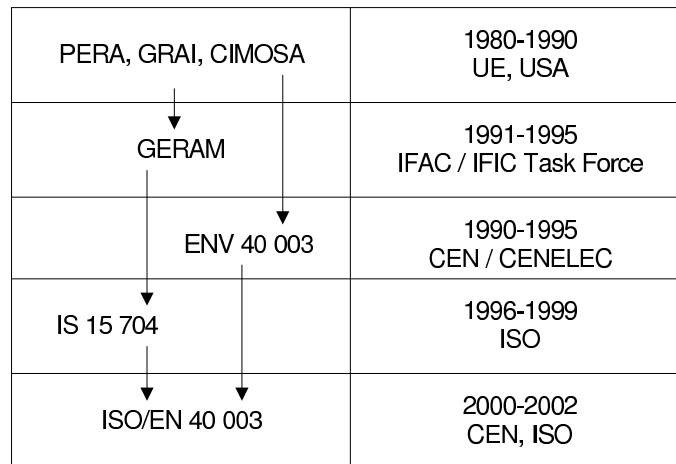


FIG. 7.3 – Évolutions des normes et des méthodes

Référence	Titre
ENV 40003	Computer Integrated Manufacturing (CIM) – Systems Architecture – Framework for entreprise Modeling
ENV 12204	Advanced manufacturing technology – System architecture – Constructs for entreprise modeling

TAB. 7.1 – Normes européennes

7.4 Cadre conceptuel

La norme ENV 40003 (ENV 40003, 2001) introduit deux notions importantes de la modélisation d'entreprise :

- le glossaire qui fournit une définition des termes de la modélisation d'entreprise,
- un cadre normatif qui fournit une approche descriptive et discriminante des modèles.

La norme est postérieure à bon nombre de méthodes et de modèles publiés. Comme dans beaucoup de cas, elle a pour but de rationaliser le domaine d'études, et de partager des connaissances entre plusieurs écoles de pensée. Ce cadre de conception, ou architecture, a permis de rationaliser les différentes contributions et « fournit un cadre général et des points de repère aux utilisateurs en leur indiquant quels aspects de l'entreprise doivent être pris en compte, les relations qui existent entre eux et la terminologie communément admise dans le domaine » (François Vernadat, 1999).

7.4.1 Dimension d'abstraction

La dimension de l'abstraction fait référence à l'utilisation du modèle en introduisant trois types de modèles¹ répartis de la manière suivante :

- le modèle de spécification définit les processus opérationnels de l'entreprise pour chaque domaine métier. Les processus sont décrits suivant une terminologie du domaine, liée à des informations, des ressources et une organisation, mais sans aucune référence à une implémentation particulière,
- le modèle conceptuel décrit plus précisément l'enchaînement des activités de chacun des processus opérationnels. Ce modèle renseigne les éléments du modèle : il décrit les attributs des entités manipulées, il précise les capacités des ressources... Le modèle conceptuel est dérivé du modèle de spécification en précisant les conditions d'implémentation dans un environnement d'exploitation connu (contraintes liées au matériel et au logiciel d'exploitation),
- le modèle d'implémentation décrit une réalisation concrète adaptée à un besoin (le codage dans un langage particulier).

7.4.2 Dimension de généricité

Cette dimension permet de décliner plusieurs niveaux de détail dans la représentation du système par un modèle. Il y a trois niveaux déclarés :

- le niveau générique représente les primitives de base du langage de modélisation, appelées aussi constructs (§ 7.4.2.1) de base de la modélisation,
- le niveau partiel représente les modèles partiels suivant une typologie d'entreprise (entreprise de production de biens, entreprise de service...),
- le niveau particulier décrit le fonctionnement spécifique d'une entreprise. Il s'agit d'une spécialisation du modèle générique.

7.4.2.1 Constructs

Un construct est un élément à caractère générique qui permet de représenter, d'une manière textuelle ou graphique, les éléments du modèle.

La norme ENV 12204 (ENV 12204, 1996) explicite ces constructs. Elle fournit une première définition des primitives de base du modèle, de leur mise en place et des possibilités d'application. Ces travaux trouveront une suite logique dans le programme européen Unified Enterprise

¹La norme ENV 40003 fait référence à sept types de modèles (identification domaine, concepts, besoins, conception, implémentation, réalisation et démantèlement). Dans le cadre de nos travaux et pour la suite du mémoire, nous préférons nous limiter à seulement trois types de modèles (spécification, conception et implémentation).

Modeling Language qui produira un méta modèle plus élaboré et que nous présenterons ultérieurement.

Sur la figure 7.4, nous avons reporté les « constructs » spécifiques à la modélisation d'entreprise (ENV 12204, 1996).

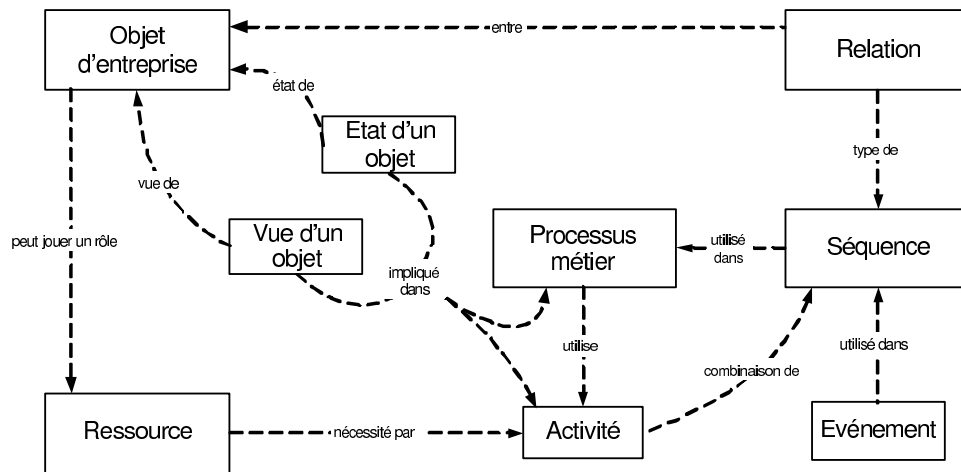


FIG. 7.4 – Relation entre « constructs »

Plusieurs concepts de première importance, et qui font force de loi aujourd'hui en modélisation d'entreprise, sont entérinés par cette norme :

- un objet d'entreprise est une entité élémentaire qui structure la description,
- un objet est doté d'une vue qui met en évidence une partie des informations qui le décrivent et certains types de comportement,
- un objet est qualifié par un cycle de vie constitué d'un ensemble d'états,
- le processus métier met à contribution la vue et l'état des objets,
- le processus est composé d'activités qui engagent des ressources et qui obéissent à une logique de changement d'état fixée par une séquence d'événements,
- les objets de l'entreprise sont en relation, ils échangent des flux de toute sorte.

À partir de ces concepts, il est possible de raffiner la description avec, par exemple :

- la définition d'une typologie de ressources,
- le renseignement des classes d'objets d'entreprise,
- la déclinaison des formes de relations en fonctions de ces types de classe,

7.4.3 Points de vue

Un point de vue de modélisation, comme son nom l'indique, est une perception particulière de l'entreprise qui met en évidence certains aspects et rend transparent les autres. C'est donc une perspective particulière pour représenter, puis observer, une même entreprise au moyen du modèle.

La norme ENV 40003 reconnaît quatre points de vue :

- la vue fonctionnelle qui fournit une représentation hiérarchisée des fonctions, des flux de l'entreprise grâce aux entrées et sorties de chaque fonction, ainsi que la logique d'enchaînement de celles-ci dans le temps ;

- la vue informationnelle qui fournit une représentation structurée d'un ensemble d'entités de l'entreprise, qui sont définies par des informations et des relations de dépendance entre ces informations ;
- la vue des ressources qui fournit une représentation de l'organisation des ressources de l'entreprise, c'est à dire de l'ensemble des moyens nécessaires pour mettre en œuvre les activités ;
- la vue de l'organisation qui fournit une représentation de l'organisation structurelle de l'entreprise : les responsabilités des individus et les unités de service au sein de l'entreprise.

Nous pouvons synthétiser ces points de vue, ainsi que les dimensions évoquées avant, en présentant le cadre conceptuel de modélisation d'entreprise sous forme d'un cube (ENV 40003). Ce cadre est effectivement doté de trois dimensions (dimension d'abstraction, dimension de généricité, dimension des vues). Chacune des 36 parties qui composent ce cube et qui décrivent les effets des modalités retenues sur chaque axe forment un élément du modèle.

Chacune des méthodes ne couvre pas l'ensemble du périmètre imposé avec ce cube.

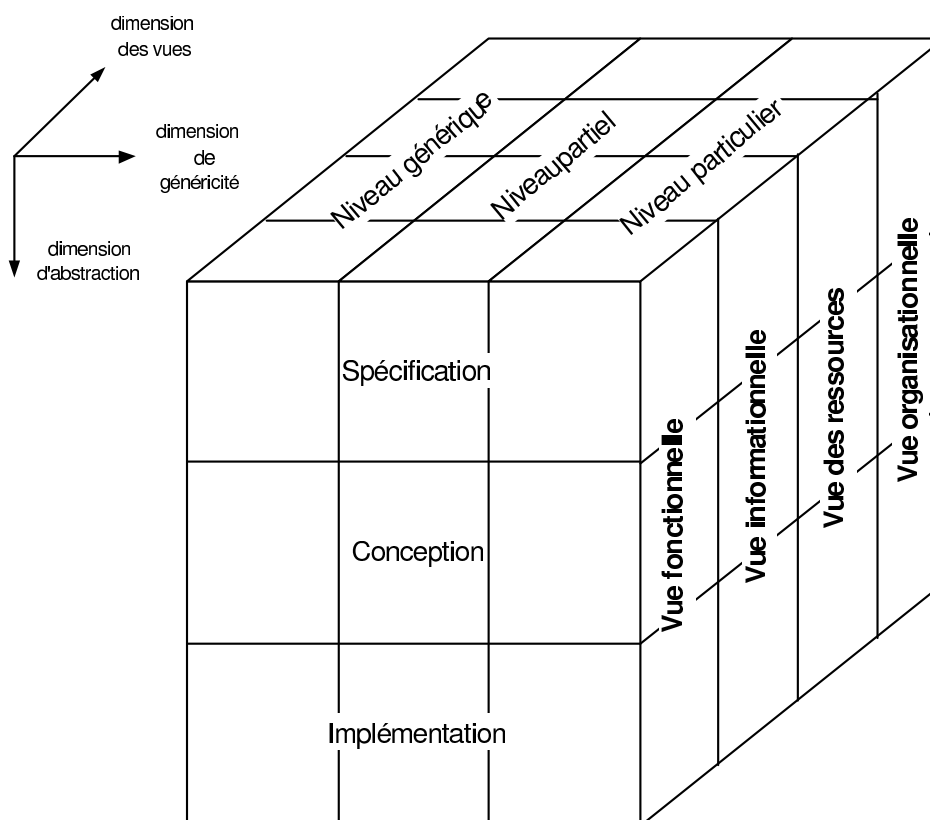


FIG. 7.5 – Cadre de conception de la norme ENV 40003

7.5 Présentation des méthodes

Nous présentons dans les sections suivantes un panorama des méthodes de modélisation d'entreprise. Ces méthodes seront plus détaillées en annexe C de ce mémoire.

7.5.1 La suite IDEF

La suite IDEF (Icam DEFinition) a été développée dans les années 70 par l'US Air Force dans un projet nommé ICAM (Integrated Computer Aided Manufacturing). Ce projet a permis la définition des trois vues suivantes :

1. IDEF0 : Utilisée pour obtenir un « point de vue fonctionnel ». Le point de vue fonctionnel est la représentation structurée des fonctions, activités ou process du système modélisé ou du sujet d'étude ;
2. IDEF1 : Utilisée pour obtenir un « point de vue informationnel » ;
3. IDEF2 et IDEF3 : Utilisées pour obtenir un « point de vue dynamique ». Le point de vue dynamique représente les caractéristiques comportementales par rapport au temps du système modélisé ou du sujet d'étude.

7.5.1.1 IDEF0

IDEF0, plus connue sous le nom SADT en France, est une méthode très employée pour la modélisation du point de vue fonctionnel. Elle repose sur un graphe simple où :

- les nœuds sont les activités du système. Ces activités permettent la transformation d'un flux intrant en un flux extrant (réalisation d'une tâche). Cette transformation se fait à partir de directives de contrôle (conditions exigées pour produire la sortie correcte) en s'appuyant sur des mécanismes (moyens mis en œuvre pour exécuter l'activité) ;
- les flèches sont les flux intrants et sortants des activités précédemment décrites.

Le résultat de cette représentation (fig. 7.6) s'organise par une suite de diagrammes, textes, et glossaires imbriqués.

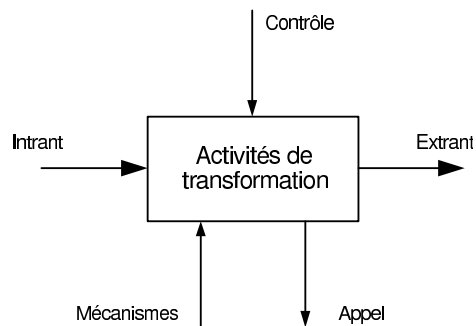


FIG. 7.6 – Le modèle SADT/IDEF0

7.5.1.2 IDEF1

La méthode IDEF1 a été conçue pour développer des modèles reflétant l'intégration de l'ensemble des informations de l'entreprise. L'approche d'IDEF1 est la suivante :

1. établir un modèle informationnel,
2. concevoir la base de données du modèle,
3. implémenter et installer la base de données ainsi que les fonctions et procédures associées.

La méthode IDEF1 offre un ensemble de règles et procédures pour la création des modèles informationnels.

Elle incorpore les graphiques, le texte et les formes nécessaires pour la meilleure description du modèle. Il existe deux composants fondamentaux dans cette représentation :

- les diagrammes : les caractéristiques du modèle informationnel, représentées selon un ensemble de règles et de procédures,
- le dictionnaire : chaque élément du modèle est décrit textuellement pour obtenir une définition explicite.

7.5.1.3 IDEF2

IDEF2 est un langage de modélisation du comportement d'un système basé sur le principe des files d'attente. C'est une méthode complémentaire à SADT/IDEF0 qui vise à répondre aux lacunes du point de vue analyse des aspects dynamiques d'un système. Cette méthode est basée sur quatre modèles : système physique, flux des entités, contrôle du système et gestion des ressources. IDEF2 aborde donc de façon privilégiée les vues informationnelles et ressources sur des niveaux d'abstraction proches de l'exécution.

7.5.1.4 IDEF3

La méthode IDEF3 est proposée en 1992 pour dépasser les limites d'IDEF0 en matière de modélisation du comportement de l'entreprise, donc la représentation des processus opérationnels.

IDEF3 modélise, comme IDEF0, les processus sous forme d'un enchaînement d'étapes, appelées unités de comportement, connectées par des boîtes de jonction et des liens. L'avantage, par rapport à l'utilisation d'IDEF0, est la possibilité de décrire un flux de contrôle du processus de manière plus complexe. Par exemple, les boîtes de jonctions permettent de représenter, à l'aide de connecteurs logiques (AND, OR ou XOR), les différents flux, les phénomènes de séquençement, de parallélisme ou de synchronisation d'activités.

7.5.2 La méthode GRAI

La méthode GRAI (Graphes à Résultats et Activités Interreliés) impose une démarche originale, puisque le point d'entrée dans la conception du modèle est la partie décisionnelle d'un système (Guy Doumeings, 1984) et (Michel Roboam, 1993).

La méthode GRAI définit un centre de décision comme le croisement d'une fonction de l'entreprise et d'un horizon/période de décision (fig. 7.7).

La grille GRAI permet de différencier les liaisons de dépendance fonctionnelle (double flèche, transmission d'une consigne ou d'un objectif) des liaisons informationnelles (simple flèche, transmission d'un flux informationnel) entre centres de décisions.

Pour compléter la grille, la mise en œuvre de la décision est détaillée en réseau d'activités. Ce réseau (fig. 7.8) permet de différencier les activités « d'exécution » de celles de « décision ».

Au début des années 90, la méthode GRAI a donné naissance à la méthodologie GIM (GRAI Integrated Methodology). GIM propose un cadre conceptuel basé sur la norme ENV 40003 (dimensions d'abstraction et de généralité) et utilise les méthodologies suivantes pour représenter différents points de vue :

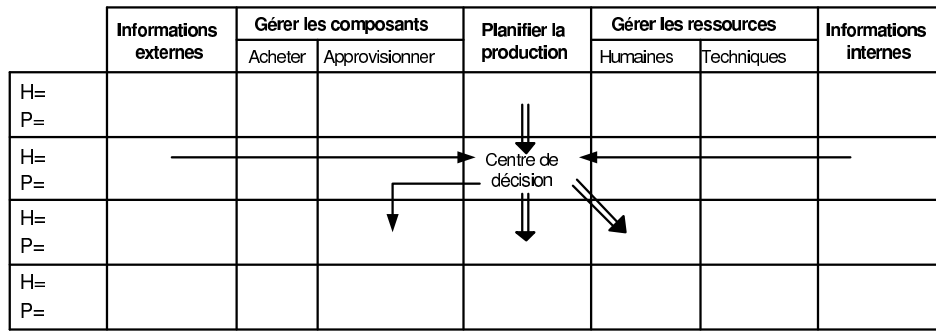


FIG. 7.7 – Exemple de grille GRAI

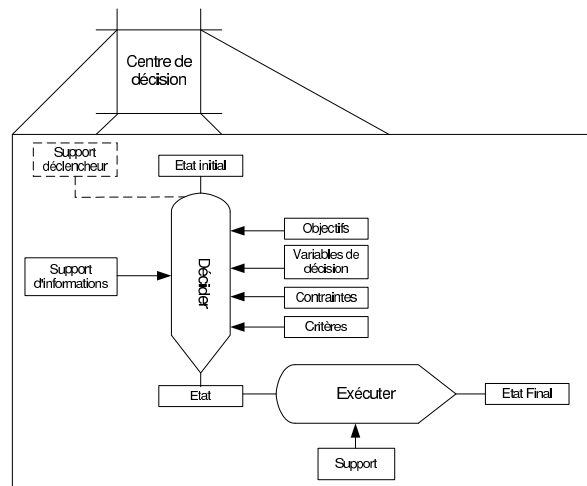


FIG. 7.8 – Réseau d'activités

- les formalismes MERISE (entité-relation) pour la vue informationnelle,
- la méthodologie IDEF0/SADT pour la vue physique et la vue fonctionnelle,
- la grille et le réseau GRAI pour la vue décisionnelle.

7.5.3 OLYMPIOS

L'objectif du modèle OLYMPIOS, (Franck Theroude et al., 2001) et (Christian Braesch, 2002), est de construire le Système d'Information d'un exploitant donné de l'entreprise. À cette fin, l'analyse de plusieurs systèmes est proposée :

- le Système de Traitement de l'Information, qui est un système de gestion d'un processeur capable de mettre en œuvre trois fonctions : utilisation, réalisation et fourniture ;
- le Système d'Information Fournisseur/Utilisateur qui a pour rôle la gestion d'échange entre deux Systèmes de Traitement de l'Information (Fournisseur et Utilisateur) ;
- le Système de Gestion des Objectifs, qui définit les missions de chaque Système d'Information Fournisseur/Utilisateur dans le but de réaliser un objectif global ;
- le Système de Gestion des Ressources, qui est chargé de qualifier les ressources (perçues sous un aspect informationnel) et de piloter les modes de partage ;
- le Système d'Activation, qui produit un graphe d'activités (plan d'actions) fonction des aspects temporels de l'entreprise.

OLYMPIOS permet de travailler sur plusieurs étapes du cycle de vie d'un Système d'Information : la spécification, la conception, l'implémentation, la mise en œuvre (et la validation) et la maintenance. Pour l'étape de spécification, le modèle OLYMPIOS s'appuie sur des Spécifications Algébriques de Type Abstrait des données (SATA) pour construire une représentation cohérente de l'existant. Dans le modèle OLYMPIOS, l'étape de conception vise à construire une organisation d'entités structurantes indépendantes d'un langage d'implémentation : les Olympios Structured ENTity (OLSEN). L'implémentation, la mise en œuvre et la maintenance sont pour leur part réalisées sur la base de langages de programmation classiques (C++) et de bibliothèques (interfaces, bases de données, etc.).

Une démarche complète d'audit est définie pour permettre la mise en œuvre de la méthodologie OLYMPIOS.

7.5.4 MECI

MECI est une méthode adaptée à l'analyse et la conception des systèmes complexes que sont les entreprises (Claude Pourcel et Didier Gourc, 2001).

La démarche de modélisation se décompose en trois étapes :

- identifier le système à modéliser : l'identification des processus principaux ainsi que les objets industriels primaires ;
- décomposer le système à modéliser : la décomposition des processus principaux en sous-processus ou activités, bien sûr sans oublier de définir les objets industriels liés aux processus et activités ;
- identifier et caractériser les activités d'un processus : l'identification et la spécification des activités et des objets ;

Pour modéliser le système et suivre cette démarche, nous avons à notre disposition un ensemble de formalismes répartis en quatre points de vue ou aspects suivants :

- fonctionnel : mise en évidence des fonctions de l'entreprise dont les objectifs et les relations seront parfaitement identifiés,
- flux : mise en évidence des liens logiques entre deux activités ou deux processus qui permettent la circulation des objets industriels,
- décisionnel : mise en évidence de la nature et du niveau de la décision qui influencent plus ou moins directement la performance des activités,
- organisationnel : mise en évidence de la structure de l'établissement par la définition des composantes de l'organisation.

7.5.5 Synthèse

Nous avons classé les différentes méthodes suivant une des dimensions de la norme ENV 40003, la dimension des points de vue. Nous n'avons pas présenté les méthodes de modélisation d'entreprise suivant les deux autres dimensions, d'abstraction et de généricité car nous n'aurions obtenu aucune information supplémentaire.

Dans la suite de nos travaux, nous faisons le choix de la méthode GRAI et de la norme ENV 40003. Tout d'abord, parce que la méthode GRAI nous sert de point d'entrée pour la modélisation des centres de décision de l'entreprise. Grâce à sa vision macroscopique, nous pouvons appréhender les relations entre les fonctions de l'entreprise, mais aussi les relations entre les

Méthode		Point de vue				
		Fonc.	Infor.	Ress.	Orga.	Décisio.
Suite IDEF	IDEF0	Oui	Non	Non	Non	Non
	IDEF1	Non	Oui	Non	Non	Non
	IDEF2	Oui	Oui	Oui	Non	Non
	IDEF3	Oui	Non	Non	Non	Non
Grille GRAI		Non	Non	Non	Non	Oui
GIM	MERISE	Non	Oui	Non	Non	Non
	IDEF0	Oui	Non	Non	Non	Non
	Grille GRAI	Non	Non	Non	Non	Oui
OLYMPIOS		Oui	Oui	Non	Non	Non
MECI		Oui	Oui	Non	Oui	Oui
ENV 40003		Oui	Oui	Oui	Oui	Non

TAB. 7.2 – Positionnement des différentes méthodes de modélisation d'entreprise

centres de décision. Les fonctions de l'entreprise identifiées peuvent être rapprochées de l'organisation. Pour chacune des fonctions, un service organisationnel lui sera affecté. Nous pouvons rapprocher ces points de vue des types d'architectures présentées (fonctionnelle, logique et physique) dans les chapitres précédents. La méthode GRAI nous permet de modéliser l'architecture logique de notre système. La méthode nous permet d'avoir une meilleure gestion du périmètre, des domaines (métiers, pilotage et support) et une première vision des données à l'aide des informations internes et externes reliées aux centres de décision.

Dans un deuxième temps, nous souhaitons utiliser la norme ENV 40003. Cette norme permet d'avoir une approche structurante dans la modélisation, tout d'abord par la mise en place d'un cadre conceptuel, et par la décomposition du modèle suivant différentes dimensions (généricité, abstraction et points de vue). La norme ENV 40003 apporte un niveau d'abstraction plus important dans la construction des modèles. Nous pouvons utiliser le point de vue fonctionnel pour représenter l'enchaînement des activités que compose un centre de décision. Nous pouvons aussi utiliser le point de vue informationnel pour construire les relations entre les différents objets de l'entreprise.

7.6 Les travaux sur UEML

Des logiciels facilitent la conception et la maintenance des modèles d'entreprise. Ils implémentent en général une méthode, et le langage de modélisation est adapté à cette méthode.

Nous citerons, à titre d'exemple :

- FirstStep (Interfacing Technologies¹) : basé sur les principes de CIMOSA (Meir H. Levi et Marios P. Klapsis, 1999),
- Aris ToolSet (IDS Scheer²) : développé par le Professeur August-Wilhelm Scheer. Cet outil est basé sur une décomposition très proche de CIMOSA, il utilise un formalisme EPC (Event Process Chain) (August-Wilhelm Scheer, 1999).
- eMagin (GraiSoft³) : outil de représentation pour la méthode GRAI.

UEML¹ est le fruit d'un programme de collaboration à l'échelle européenne qui a pour objectif

¹<http://www.interfacing.com>

²<http://www.ids-scheer.com/>

³<http://www.graisoft.com>

¹<http://www.ueml.org>

de définir une interface standardisée entre les outils de la modélisation d'entreprise basés sur des modèles différents. Ce travail conduira in fine à un certain degré de compatibilité entre les outils.

Dans ce programme, il y a deux axes de travail. Le premier axe définit des constructs en langage UML (Unified Modeling Language). Le deuxième axe consiste à développer l'interopérabilité par la définition d'un schéma au format XML (eXtended Markup Language), à partir des constructs, permettant d'échanger des informations entre outils et donc entre langages de modélisation d'entreprise.

Le groupe de travail européen a fourni une première démonstration des capacités d'échange et de dialogue entre les outils de modélisation eMagin, MO2GO² et Metis³. Nous décrirons le méta modèle d'UEML dans la dernière partie.

7.7 Synthèse et conclusions

L'entreprise a, aujourd'hui, de fortes incitations à structurer son organisation autour de processus, définis aussi clairement que possible, mais surtout formalisés et autour desquels les différents métiers communiquent.

Il y a plusieurs courants de travaux scientifiques qui accompagnent cette forme de structuration et des langages de modélisation, souvent graphiques, sont proposés pour obtenir une représentation formelle. La modélisation d'entreprise est certainement l'une des formes les plus accomplies.

Un langage est construit sur une syntaxe et une sémantique. La syntaxe fixe les éléments (grammaire) et les règles qui fixent l'emploi du langage. La sémantique est directement liée au pouvoir d'expression, elle dicte l'interprétation qu'il faut faire du langage pour trouver tout le sens d'une proposition.

La syntaxe et la sémantique des langages de modélisation font partie des bases de comparaisons entre les différents modèles.

Dans un premier temps, nous avons mis en évidence les dimensions qui régissent la conception et l'utilisation d'un modèle. Les dimensions d'abstraction et de généralité sont essentielles pour ne pas se noyer dans la masse d'informations et de connaissances à traiter. Elles sont identifiées et employées dans les modèles d'entreprise et dans les modèles de Systèmes d'Information avec une même philosophie. Le cadre de Zachman, la décomposition des niveaux de représentation de MERISE, les mécanismes d'instanciation (pour l'abstraction), de généralisation ou de spécialisation (pour la généralité) de l'approche objet sont autant d'illustrations.

Ces deux dimensions sont à la base du cube de la norme CEN 40003 fixant le cadre conceptuel de la modélisation d'entreprise. La plupart des méthodes se réfèrent aujourd'hui à ce cube pour mesurer leur capacité de représentation. Les outils logiciels qui permettent de modéliser mettent en œuvre les mécanismes qui autorisent la navigation sur ces deux dimensions.

Mais c'est la dimension des points de vue qui regroupe les éléments qui fondent le langage de modélisation. La représentation partielle, par le concept de point de vue, est mise en œuvre. C'est donc au niveau de chaque point de vue qu'il devient possible de faire une analyse du langage.

²<http://www.um.ipk.fhg.de/mogo/bprhome.htm>

³<http://www.computas.com/>

Nous avons listé quelques méthodes, qui sont présentées plus en détail en annexe C. De cette étude, il ressort quelques points fort :

- le point de vue de l'information est très orienté vers la structuration des données. Les modèles réutilisent des formalismes proches du modèle de base « entité-relation ». Lorsque l'on chemine du conceptuel vers le physique en se concentrant sur les bases de données, on intègre progressivement les contraintes techniques de la base (base relationnelle, base objet, ...). Il n'y a pas de divergence majeure entre modèle d'entreprise et modèle de Système d'Information sur cette partie des langages. La suite IDEF intègre directement le modèle de données de MERISE, on trouve dans le logiciel eMagin, qui implante la méthode GRAI, un diagramme de classe UML, par exemple ;
- le point de vue de l'organisation est représenté par des organigrammes, ou par des matrices dans le cas de structure plus complexe. Ce sont en général des formes graphiques simples où les personnes se voient attribuer un ou des rôle(s) dans l'organisation. Il y a naturellement une difficulté à descendre au niveau de l'acteur dans le processus, car il s'agit d'un rôle particularisé (au sens de la dimension d'abstraction) et qui ne relève pas nécessairement d'une logique hiérarchique (la notion de chef de projet par exemple). En cela, l'acteur peut être considéré comme une ressource particulière dans le point de vue fonctionnel. Mais c'est souvent une représentation insuffisante lorsque l'application rentre dans le champ des connaissances et des compétences ;
- le point de vue des ressources comprend deux formes de connaissances. La première forme est relative à la description des ressources, il peut s'agir d'une typologie, par exemple, fixant quelles activités sont autorisées à mobiliser des ressources. La seconde forme est relative à l'expression des contraintes d'affectation des ressources qui peuvent être exclusives ou cumulatives. Dans tous les cas, c'est par référence à l'état de la ressource, qui est un attribut variable, que la décision d'affectation peut être mise à exécution. Si ces contraintes sont évoluées, il peut être pratique de recourir à une représentation de type machine à état.
- le point de vue fonctionnel a déjà été évoqué parce qu'il s'agit de la modélisation des processus. Il y a deux formes de représentation, selon que l'on se place :
 - au niveau de l'analyse fonctionnelle avec une logique d'énumération des fonctions et de propagations des flux. Un modèle de type IDEF0 y ajoute des objectifs et des contraintes de fonctionnement vu au niveau de chaque *fonction*.
 - au niveau du comportement où l'exécution des processus doit être décrite dans le temps en ayant le souci de prendre en charge des propriétés structurelles telles que le parallélisme, la synchronisation *d'activités*.

Dans les langages de modélisation de Système d'Information, toutes les primitives permettent de couvrir les points de vue informationnel et fonctionnel. Bien souvent, les modèles d'entreprise sont plus riches parce qu'ils sont restreints à un domaine et qu'ils fixent les éléments de connaissances (les ontologies) à manipuler. Ainsi, dans la méthode GRAI, la grille fonctionnelle ou la grille de conduite utilise un cadre qui décompose les décisions de l'entreprise en sous-ensembles caractérisés par la fonction impactée et l'horizon de temps. C'est le centre de décision qui indique les personnes habilitées à prendre ces décisions dans l'organisation.

Cette grille n'appartient à aucun des points de vue évoqués dans la norme CEN 40003, mais on retrouve sur les colonnes une logique de fonctions et de processus. On retrouve également dans la grille de conduite une relation aux informations. Enfin, dans les mécanismes de propagation de décisions entre centres, il y a une notion de responsabilité des acteurs qui est sous-jacente.

De notre avis, il n'y a pas lieu d'opposer les langages de modélisation d'entreprise et de Système d'Information. Les approches reposent sur des concepts similaires et partagés. La modélisation

d'entreprise offre un cadre plus spécialisé et plus contraignant où les interactions entre les différents éléments (entités) sont décrites avec des syntaxes plus évoluées. La sémantique des modèles d'entreprise est plus riche et certainement très adaptée à une représentation de l'architecture fonctionnelle. Néanmoins, à un niveau de représentation qui est celui des points de vue, rien n'interdit d'exploiter un langage comme UML, par exemple pour traduire les connaissances. C'est effectivement sur :

- des structurations de données,
- des logiques de changement d'état,

que reposent les besoins d'expressions. Nous nous attachons à démontrer cela dans la dernière partie du mémoire.

Troisième partie

Utilisation de la modélisation

Chapitre 8

Utilisation du langage UML pour la modélisation d'entreprise

Le langage UML peut être utilisé pour représenter des points de vue d'un modèle d'entreprise. Nous allons débiter ce chapitre par une proposition qui explique cet usage : comment les constructs de la modélisation d'entreprise sont-ils traduits en langage UML ? Quels diagrammes UML pour quel point de vue ?

Puis, nous dépasserons le stade syntaxique pour entrer dans le champ sémantique. Il faut tirer profit de la richesse des modèles d'entreprise. Nous étudierons une variation du modèle GRAI par un usage de la grille comme chef d'orchestre de la cartographie des processus (Franck Darras et al., 2003c). A partir de cette carte, il est possible de cheminer dans les points de vue en naviguant entre les diagrammes, en conservant les pratiques habituelles de la modélisation UML.

8.1 Proposition d'un méta-modèle

Un méta-modèle est un langage formel ou semi-formel permettant de modéliser des systèmes particuliers, à savoir des modèles (Chabane Oussalah, 1997). Le méta modèle est devenu un point d'entrée obligatoire pour la modélisation d'entreprise. En effet, il permet de dresser une liste, à un niveau générique, des éléments qui seront utilisés dans la représentation sous une forme plus instanciée (particularisée). Cela s'apparente effectivement à un langage puisqu'il fournit les mots (les constructs) et les relations entre les mots (la grammaire).

De plus, le méta-modèle permet de tisser des liens entre les différentes vues du modèle d'entreprise. Les présentations qui suivent montrent qu'il est possible de classer les composantes du méta-modèle dans des cadres qui sont les points de vue. Les relations qui traversent les cadres expliquent alors les passerelles entre les points de vue. En cela, le méta-modèle est un fil rouge qui permet à un niveau très macroscopique de conserver une cohérence à l'ensemble des graphes.

8.1.1 Les apports du méta-modèle de CIM-OSA

La figure 8.1 montre la répartition des constructs entre les différentes vues. Ce méta-modèle est différent de celui de la norme ENV 12204 présenté à la figure 7.4. Nous allons expliquer ces différences.

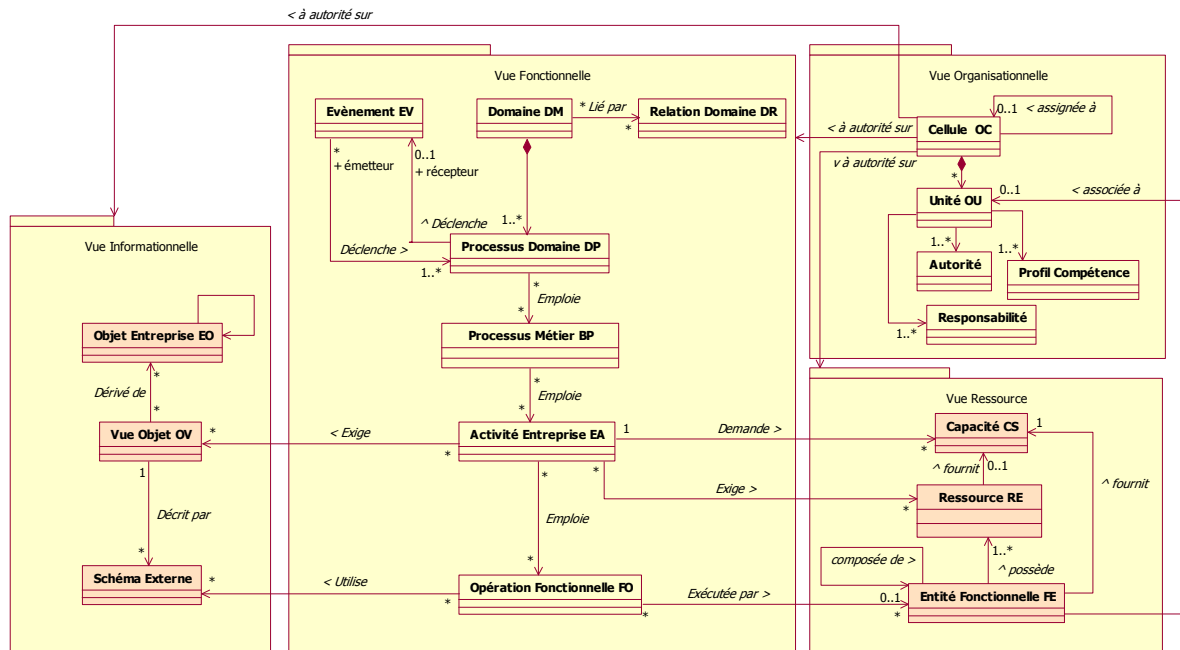


FIG. 8.1 – Méta-modèle de CIM-OSA

Lorsqu'on examine cette représentation, il apparaît que la vue fonctionnelle est centrale, et que les autres points de vue gravitent autour d'elle. Cela correspond à la définition donnée par CIM-OSA du système entreprise décrit comme « un grand ensemble :

- de processus concourants,
- d'agents ou d'acteurs ou de ressources communicantes

que l'on peut organiser en domaines (ou zones fonctionnelles) ayant des interactions en termes d'échanges (événements ou résultats) et qu'il faut coordonner. »

Le domaine est donc l'élément structurant de l'approche conceptuelle du modèle. Un domaine est défini comme un regroupement de processus en module indépendant permettant de gérer la complexité. Nous aurons à creuser ultérieurement autour de cette notion de domaine, car l'établissement d'une cartographie de processus ne peut pas nous laisser indifférent dans le cadre de l'architecture fonctionnelle du Système d'Information. En effet, si les modules d'un ERP peuvent constituer des domaines au sens de CIMOSA, nous avons déjà signalé que les processus transversaux traversent les modules de l'ERP et qu'il faut conserver une capacité à communiquer entre les modules qui soit adaptée. Ce cheminement est facilité par la cartographie des processus. D'ailleurs, le méta modèle de CIM-OSA découpe les domaines en processus du domaine, qui sont liés à des événements d'une part, puis à des processus métier d'autre part. Ce découpage permet bien d'imaginer des logiques d'activation de processus métiers ou d'activités qui, mis bout à bout, forment un processus transversal.

L'objet d'entreprise subsiste dans la vue informationnelle. Il a toujours une vue qui correspond à un ensemble d'états possibles de l'objet. Mais la notion d'état d'objet a disparu. La notion de schéma vient compléter la vue d'objet pour expliciter la structure des données.

La notion de séquence est également absente de ce méta-modèle. En effet, c'est le lien bilatéral entre processus et événement qui est parcouru itérativement pour construire la séquence d'activités.

Les ressources sont décrites par leur capacité. L'entité fonctionnelle permet de distinguer les ressources qui participent directement à la dynamique du système opérant comme les

hommes, les machines et les applications informatiques, des autres ressources à caractère plus « organique ». C'est pour cette raison que l'entité fonctionnelle contribue à l'exécution de l'opération fonctionnelle (la tâche).

Enfin, le point de vue de l'organisation introduit la notion d'unité qui est un rôle, ou une position, généralement tenu par une personne. Une cellule correspond à un groupe d'unités qui a une reconnaissance particulière au sein de l'organisation (service, département, direction, équipe projet, ...).

8.1.2 Les apports du méta-modèle de UEML

Ce méta-modèle est extrait d'une référence de François Vernadat (2001). On y trouve également une répartition des constructs en différentes vues. Les liens entre les vues sont de nature moins centralisée que dans le modèle CIMOSA. Il y a également une similitude plus forte avec la norme ENV 12204.

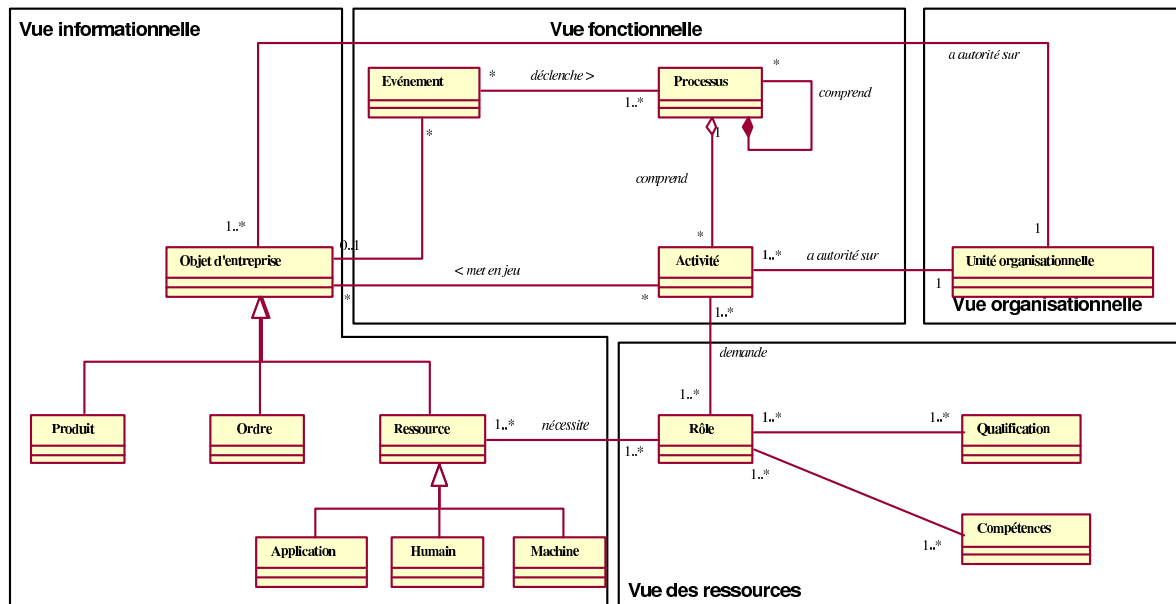


FIG. 8.2 – Méta-modèle de UEML

La notion de domaine n'apparaît pas dans UEML. Le processus est mis en relation de composition avec lui-même pour permettre une hiérarchisation. Le lien direct avec l'événement est tissé, sans utiliser de séquence. La vue organisationnelle est plus pauvre, mais elle est décrite par la notion élémentaire d'unité associée à une autorité sur une activité et sur un objet. Parmi les objets, on trouve une distinction entre les entités fonctionnelles (désignée par ressource) qui sont présentées avec une relation d'héritage. Les différentes vues de l'objet sont également décrites par une relation d'héritage. Tout comme pour la vue de l'organisation, la vue des ressources est simplifiée, puisqu'on ne trouve que le rôle joué par une entité fonctionnelle.

8.1.3 Notre proposition

Les deux méta-modèles commentés ci-dessus sont assez proches. Certains éléments présentent un intérêt dans le cadre de nos travaux :

- le « domaine » : il permet de décomposer le système selon une certaine logique,

- la distinction entre « processus domaine » et « processus métiers » que nous avons décrite comme une façon d'aborder la cartographie des processus.

Nous n'avons pas cherché à conserver la finesse de description des points de vue de l'organisation et des ressources de CIMOSA, notre solution est à ces niveaux très proche d'UEML. Les relations d'héritage sont reprises dans le point de vue de l'information en conservant le lien avec la notion de rôle des ressources.

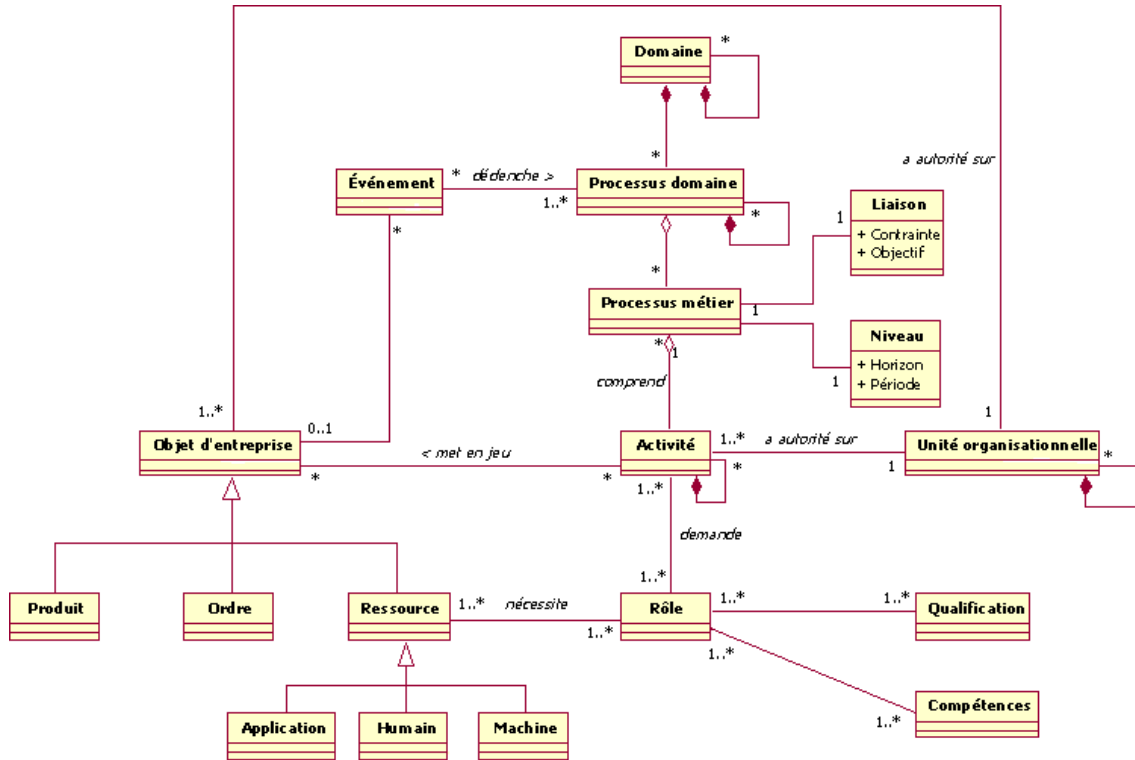


FIG. 8.3 – Proposition de méta-modèle

Toutefois, il nous a semblé intéressant de modifier quelques éléments :

- le domaine, les processus domaine, les activités et les unités organisationnelles sont dotés de relation de composition récursive, qui laissent la porte ouverte à la possibilité de construire une arborescence de domaines, de processus, d'activités ou d'unités,
- afin d'exprimer la dominance du processus domaine sur le processus métier, et du processus métier sur l'activité, une relation d'agrégation est retenue,

Enfin, avec le souci de bâtir une cartographie de processus en utilisant la grille GRAI, deux nouvelles composantes sont introduites en relation avec le processus métier. La première composante, le niveau, correspond à des informations temporelles :

- l'horizon qui correspond à la durée maximale d'obtention du résultat attendu pour un processus métier et sera ainsi en accord avec la portée de la décision dans un centre de décision de la grille GRAI,
- la période (concept hérité de GRAI) qui correspond à la durée maximale d'exécution du processus et sera ainsi en accord avec la portée de la décision dans un centre de décision.

La seconde composante correspond à des concepts qui délimitent un centre de décision dans une grille GRAI : l'objectif et les contraintes. Ces objectifs et contraintes sont transmis lors de la transition d'un processus métier à un autre. Nous aurons l'occasion d'illustrer leur emploi ultérieurement dans le mémoire.

8.2 Mise en application du méta-modèle sur un cas d'école

Le tableau 8.1 a été construit à partir d'éléments d'un chapitre du livre de Jean-Michel Tysebaert (2001) dédié à la gestion commerciale, et par une réflexion menée avec des experts du métier au sein de la société pour laquelle cette étude est menée.

Il a une valeur d'illustration, sans plus d'ambition. C'est un cas d'école qui concerne le domaine commercial avec deux processus de domaine : la vente et l'achat.

Domaine	Processus domaine	Processus métier	Activité	Évènement	Organisation	Rôle
Commercial	Vente	Prospection, Offre, Devis, Commande de vente, Livraison, Facturation...	Identification client, Identification produit, Saisir information client, Saisir information produit, Saisir Information Cmde, Contrôle disponible...	Saisir rapport visite, Saisir offre, Saisir offre via Internet, Valider offre, Valider devis, Valider Cmde...	Service des ventes	Gestionnaire des ventes
	Achat	Commande d'achat, Réception, Facturation, Règlement	Demande d'achat, Sélection fournisseur, Ordre d'achat, Contrôle réception...	Sélectionner fournisseur, Passer un contrat, Faire une demande, Passer Cmde via Internet...	Service des achats	Gestionnaire des achats

TAB. 8.1 – Spécification du processus Commercial

	Id. Client	Id. Produit	Saisir information client	Saisir information produit	Saisir information Cmde	Contrôle faisabilité
Prospection						X
Offre	X	X	X	X		
Devis	X	X	X	X		
Commande de vente	X	X	X	X	X	

TAB. 8.2 – Relation entre processus métiers et activités du processus domaine de « Vente »

	Demande d'achats	Sélection fournisseurs	Ordre d'achat	Contrôle Réception
Commande d'achat	X	X	X	
Réception				X
Facturation				
Règlements				

TAB. 8.3 – Relation entre processus métiers et activités du processus domaine d'« Achat »

Les têtes de colonne du tableau 8.1 correspondent aux constructs de notre méta-modèle. Il apparaît qu'une activité telle qu'est décrite dans ce tableau est très proche d'un traitement, c'est-à-dire d'une transformation de l'information réalisée par la machine. Le domaine concerné est très abordable, car il est commun à beaucoup d'activités et ne recouvre pas le niveau de complexité d'un domaine de production ou de conception de produits.

Afin d'illustrer les relations entre processus métier et activités, nous avons décrit dans les tableaux 8.2 et 8.3, les associations effectives entre ces deux catégories. Une activité peut être

impliquée dans plusieurs processus métier, c'est le cas de l'identification du produit vis-à-vis de l'offre, du devis ou de la commande vente.

Pour compléter la description de ce cas d'école, le tableau 8.4 dresse le bilan des objets d'entreprise impliqués dans la réalisation des processus du domaine.

Processus domaine	Activité	Objet d'entreprise				
		Produit	Ordres	Ressource		
				Application	Humaine	Machine
Ventes	Identification client, Identification produit, Saisir information client, Saisir information produit, Saisir information cmde, Contrôle disponible, ...		Offre, Devis, Contrat, Marché, Commande	Client, Produits, Contrat, Marché, Stocks, Termes & conditions, Internet, EDI	Prospects, Clients, Prospects, Offreurs, Deviseurs, Commercial, Comptable	
Achat	Demande d'achat, Sélection fournisseur, Ordre d'achat, Contrôle réception, ...	Matières premières, Pièces, Outils, Sous-traitance,	Demande d'achat, Contrat, Marché, Commande, Bon de réception, Facture, Paiement	Produits, Stocks, Fournisseurs, Suggestions, Demande, Commande, Réception, Facture, Règlement, Internet, EDI	Acheteurs, Magasiniers, Comptable	Magasins, Emplants

TAB. 8.4 – Relation entre les activités et les objets d'entreprise

8.2.1 Mise en œuvre du langage UML sur ce cas d'étude

Dans les sections suivantes, nous utilisons UML pour représenter les points de vue de la norme ENV 40003 tels qu'ils ont été présentés au chapitre 7.

8.2.1.1 Point de vue fonctionnel

Les éléments de notre méta-modèle (fig. 8.3) qui composent ce point de vue sont décrits par le schéma ci-dessous.

L'application au cas d'école conduit au diagramme de classe de la figure 8.5. Pour plus de clarté, nous n'avons représenté qu'une partie de ce diagramme de classe. Nous utilisons les stéréotypes d'UML pour apporter de la sémantique aux classes et faire plus facilement le rapprochement entre les classes de ce modèle et celles du méta-modèle.

Chacune des classes processus possède ses propres attributs et opérations. Nous avons renseigné les informations concernant :

- le niveau du processus métier « commande » suivant un horizon et une période. Dans ce cas, nous l'avons placé en temps réel. Dans le cas d'un processus métier « gestion du carnet de commande », nous aurions pu avoir un horizon d'un mois et une période d'une semaine, par exemple.
- les contraintes et les objectifs de ce processus sont exprimés par la classe LiaisonTemps-Cycle de type « Liaison » exprimant la contrainte que la commande doit être terminée en deux jours.

Dans la littérature, différents auteurs ont déjà travaillé sur la représentation des processus en utilisant le formalisme UML. Nasser Kettani et al. (1999) proposent d'identifier les processus aux cas d'utilisation, puis de compléter par des diagrammes de comportement. D'autres

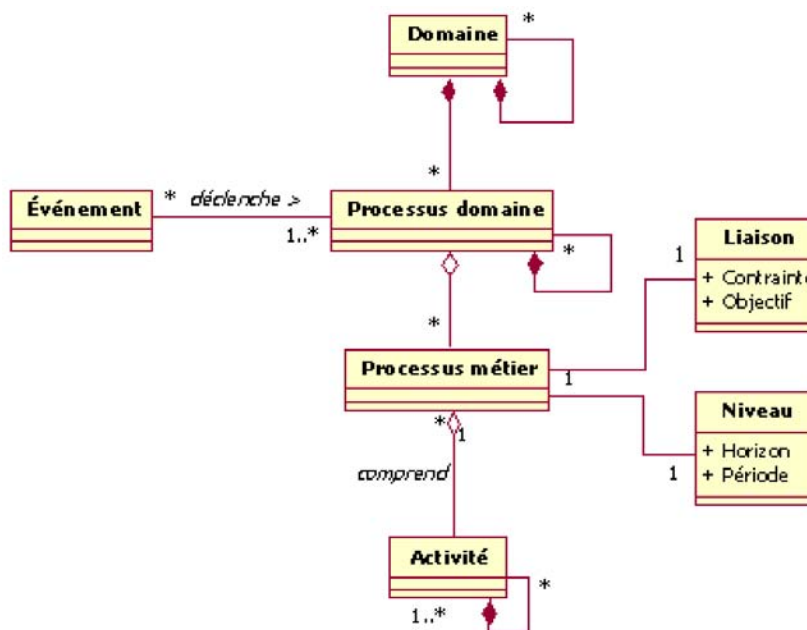


FIG. 8.4 – Méta-modèle partiel - point de vue fonctionnel

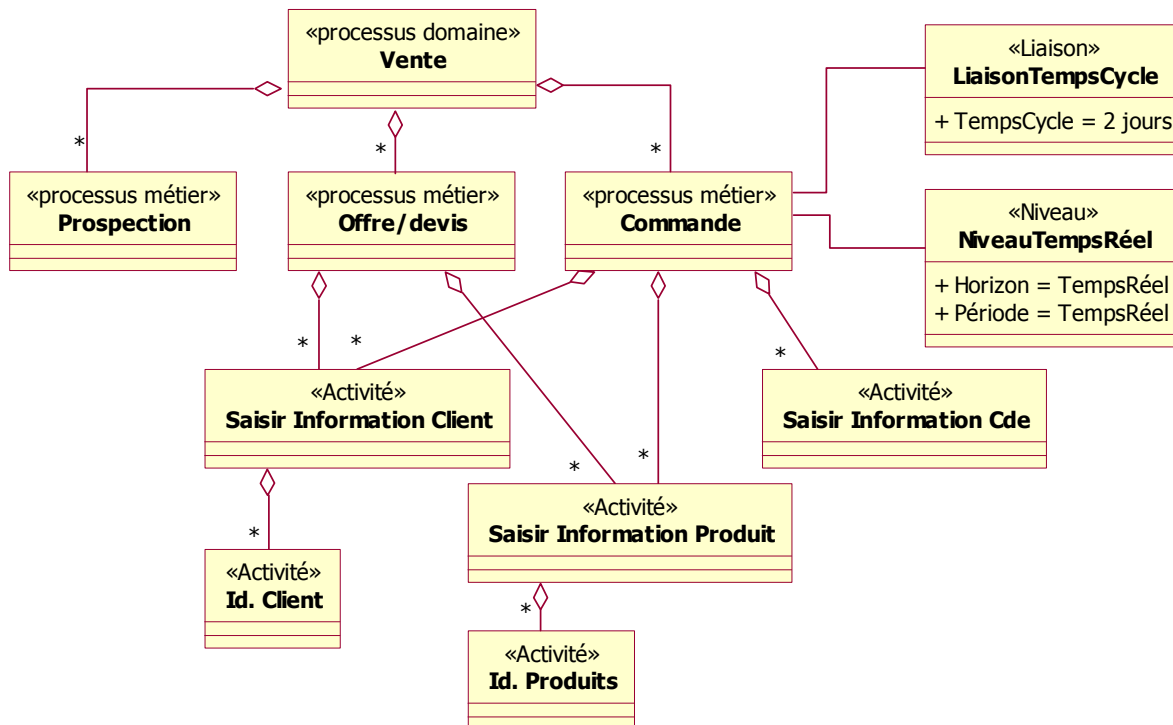


FIG. 8.5 – Diagramme de classes du processus domaine « Vente »

auteurs, Chantal Morley et al. (2000) modélisent les processus à l'aide des diagrammes de classes et des diagrammes de comportement. En fait, pour ces derniers auteurs, le cas d'utilisation se situe à un niveau supérieur, intitulé workflow, qui se concentre sur les activités de communication et de collaboration entre acteurs.

Nous pensons que la modélisation des processus doit effectivement faire appel à ces deux formes de contribution. Sur la figure 8.6, nous utilisons le diagramme des cas d'utilisation pour représenter les domaines de l'entreprise en tant que paquetages UML, et les processus domaine de l'entreprise en tant que cas d'utilisation UML. Les relations entre différents cas d'utilisation permettent de représenter la communication entre les processus, alors que les relations entre les paquetages expriment les dépendances entre les domaines. Cette dépendance sera utile lors du déploiement de la solution pour assurer une cohérence au niveau des domaines déployés dans l'entreprise.

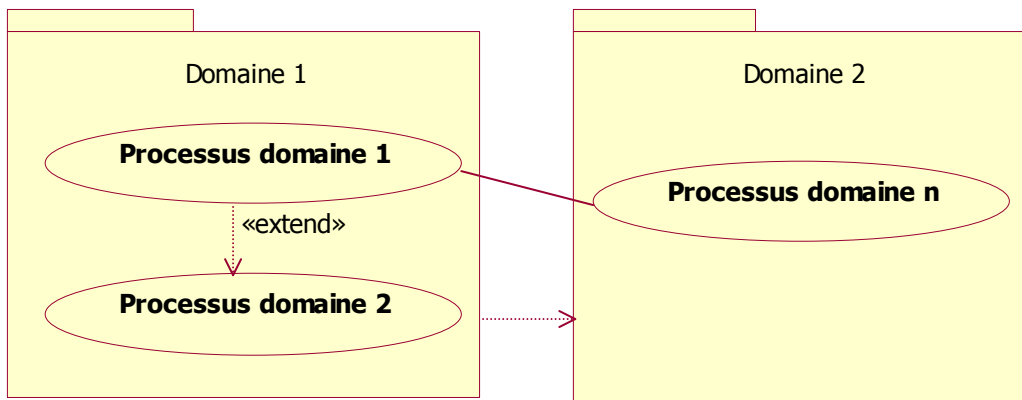


FIG. 8.6 – Cadre de description des processus domaine

Nous traduisons ce choix de représentation sur notre cas d'école (fig. 8.7). Il y a un domaine avec deux processus domaine.

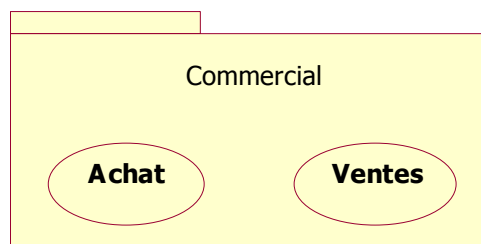


FIG. 8.7 – Représentation des processus domaine avec les cas d'utilisation

Dans la modélisation UML, les cas d'utilisation (« processus domaine ») appartiennent donc à des paquetages et ces éléments seront caractérisés uniquement par leurs noms.

Il est possible d'affiner les cas d'utilisation pour préciser les différents processus métiers ou activités qui les composent. Ces deux autres éléments du méta-modèle (processus métiers et activités) seront aussi représentés dans le diagramme d'activités, mais en utilisant la notion de stéréotype UML pour les marquer. Le diagramme d'activité est un des diagrammes de comportement. Il permet de définir les scénarios d'enchaînement des processus et des activités qui vont être exécutés avec :

- la description de l'enchaînement des processus métiers,
- la mise en évidence des étapes de validation,

- la représentation des activités détaillant le déroulement du processus métiers,
- la représentation de la logique d'enchaînement en utilisant des opérateurs logiques du langage,
- le lien avec des objets d'entreprise, instanciés à partir des classes du modèle,
- l'utilisation des opérateurs d'émission et de réception de message permettant la représentation d'un workflow.

La figure suivante est un diagramme de ce type. Les points représentent les événements de déclenchement du processus (ici plusieurs points d'entrée). Dans les cadres sont indiqués les processus métier qui sont mis à exécution avec leur intitulé précédé par le stéréotype. Les objets qui sont portés sur la gauche (commande, client, produit) correspondent aux informations en entrée et sortie d'un processus. Les flèches qui relient les objets au processus sont différentes de celles qui décrivent la logique d'enchaînement. On retrouve à ce niveau, sous une forme peu explicite si on la compare à un diagramme IDEF0, une logique de flux d'information. Enfin, comme dans un organigramme classique, les tests sont représentés par des losanges exprimant un branchement conditionnel.

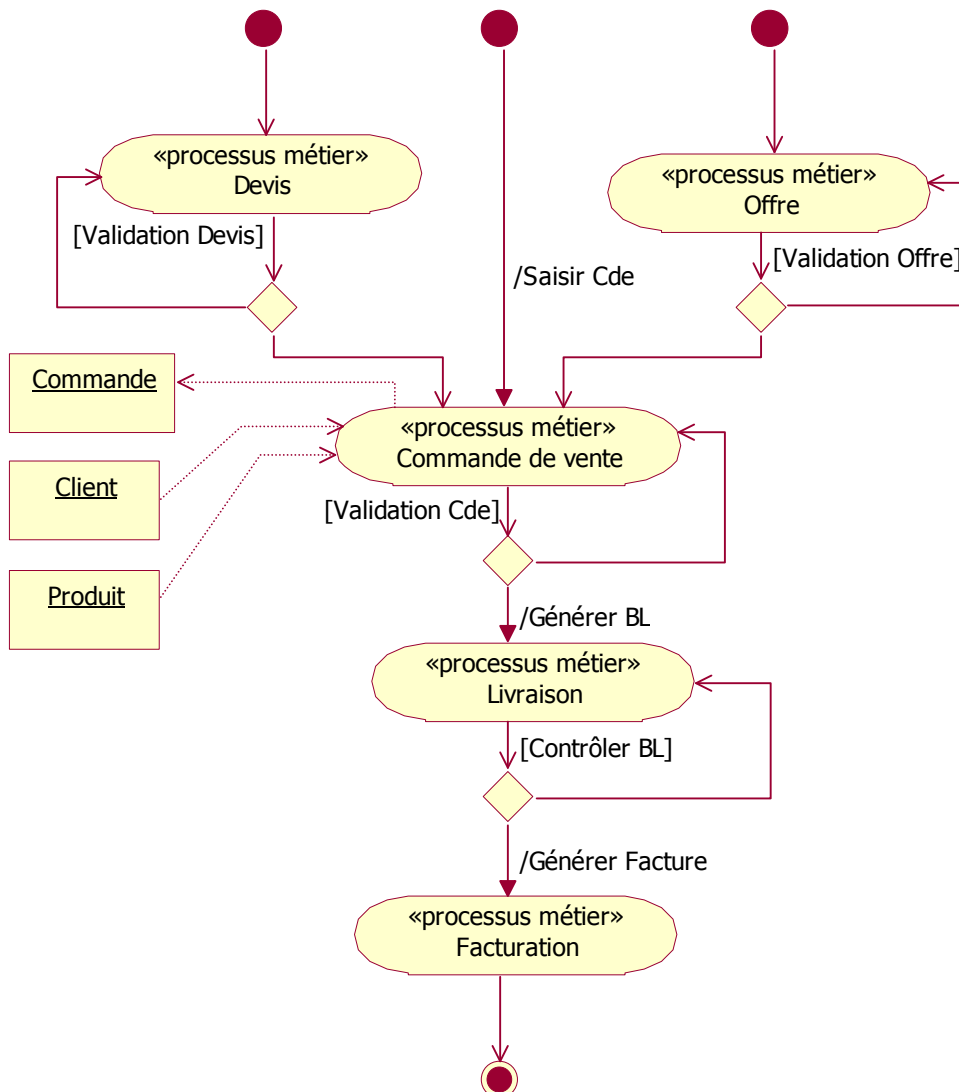


FIG. 8.8 – Représentation comportementale du processus domaine « Vente » par un diagramme d'activités

Il est possible d'invoquer un autre diagramme d'activités pour décrire l'enchaînement des ac-

tivités qui composent un processus métier. Par ce moyen, la représentation prend un caractère hiérarchique : chaque description de comportement est associée à l'objet d'entreprise ou au processus qui en est la source. Ci-dessous, nous illustrons plus en détail le comportement du processus métier « commande de vente » à partir de certaines des activités avec lesquelles il possède une relation (fig. 8.5).

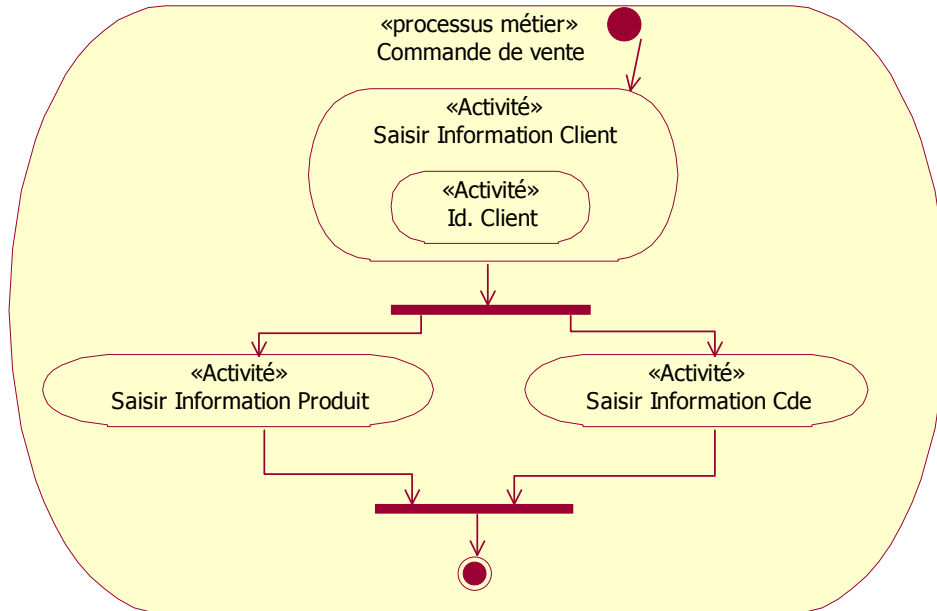


FIG. 8.9 – Enchaînement des activités pour le processus métier « Commande de vente »

Nous pouvons aussi décrire les différents états représentatifs des activités de l'entreprise. Chaque processus ou activité du système se trouve dans une situation particulière à un instant donné. Nous représentons ce point de vue en utilisant un diagramme d'états-transitions (fig. 8.10) qui permettra de décrire l'évolution d'une instance de classe (évolution d'état d'un objet en termes de comportement et de réaction à divers événements internes ou externes). Cette représentation est inspirée des travaux de normalisation menés par le WFMC¹.

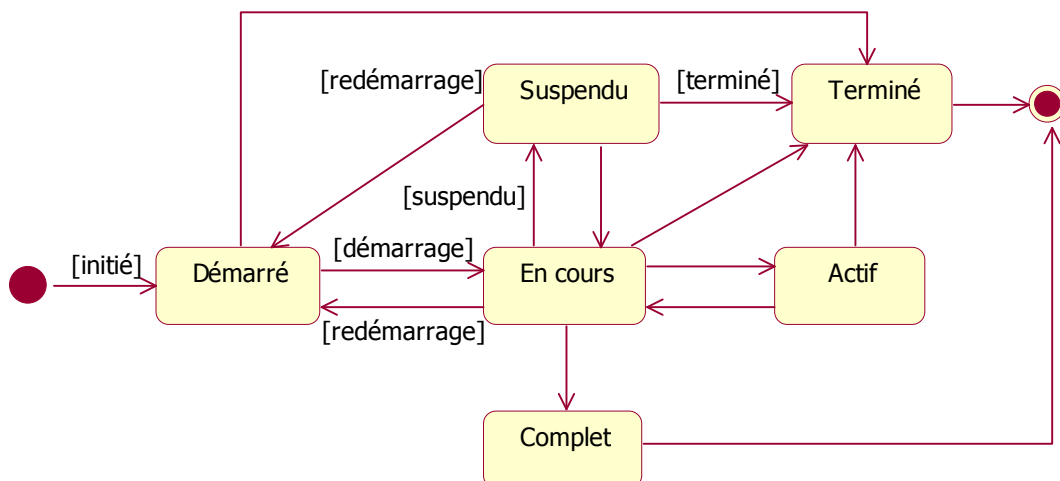


FIG. 8.10 – État d'un processus ou d'une activité

Cette représentation permet d'obtenir une trace de l'état de tous les processus et activités de

¹<http://www.wfmc.org>

l'entreprise. À partir de ces traces, nous pouvons effectuer les traitements permettant d'obtenir des tableaux de bords sur l'état des processus.

8.2.1.2 Point de vue informationnel

La partie du méta-modèle (fig. 8.3) qui est relative à ce point de vue est décrite par le schéma partiel ci-dessous.

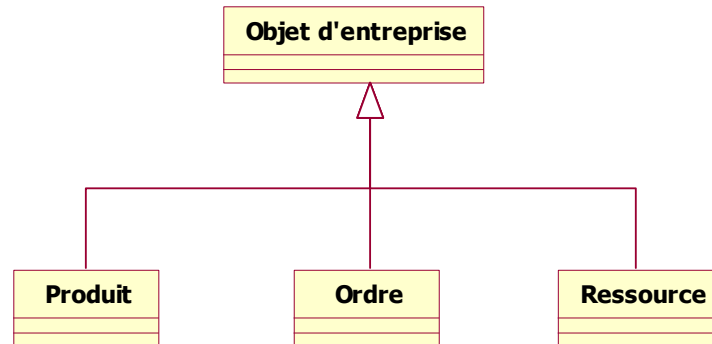


FIG. 8.11 – Méta-modèle partiel : point de vue informationnel

Les objets de l'entreprise correspondent aux éléments réels qui circulent dans l'organisation, par exemple une commande (fig. 8.12) ou un bon de livraison.

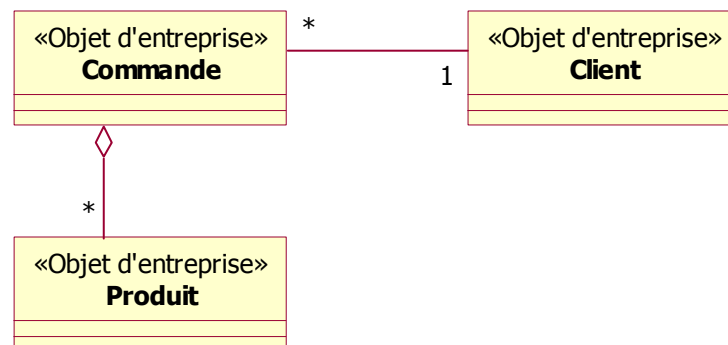


FIG. 8.12 – Diagramme de classes des objets de l'entreprise

Nous pouvons résumer la relation entre un client et le produit (au travers des commandes) à l'aide du diagramme simple ci-dessus (fig. 8.12). Nous voyons aussi qu'il existe une forme de relation entre les objets présents dans le diagramme d'activités (fig. 8.8) de la vue fonctionnelle et ce diagramme de classe. Une commande est reliée à un unique « client », tandis qu'un client de l'entreprise peut n'avoir aucune, une ou plusieurs commandes. Nous avons utilisé une relation plus forte entre le produit et la commande afin d'exprimer le fait que le produit fabriqué a été lancé pour une ou plusieurs commande(s).

Nous pouvons obtenir un niveau de détail plus élevé dans la représentation en raffinant une partie de ce diagramme pour expliquer, par exemple, les éléments d'information qui vont composer la commande (fig. 8.13).

La commande est composée de :

- conditions de commande : les conditions de commandes correspondent à l'en-tête de la commande, c'est-à-dire les références du client, les différentes adresses de livraison et facturation ;

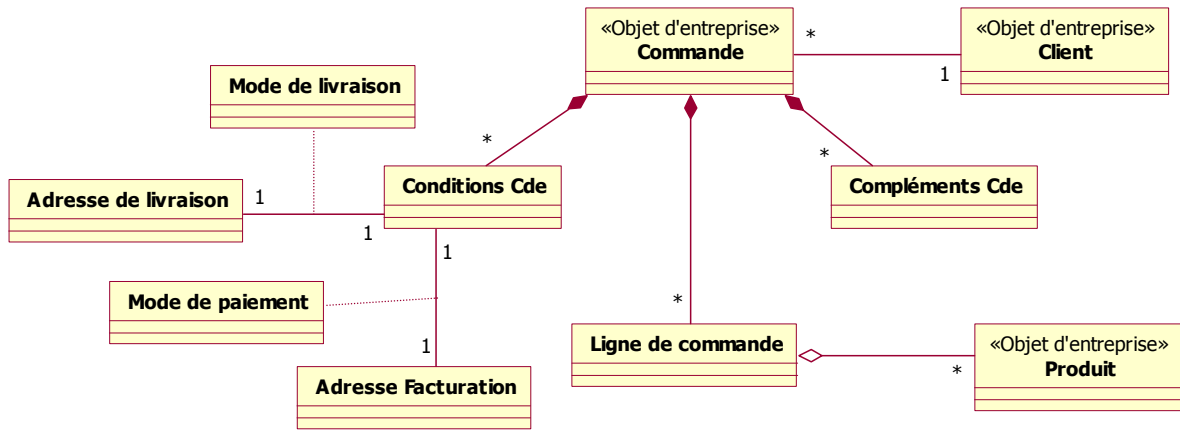


FIG. 8.13 – Diagramme de classe détaillé pour la commande

- lignes de commandes : les lignes de commandes stockent toutes les informations liées à la relation commande-produit, c'est-à-dire la quantité de produit commandée, le prix unitaire, le délai de livraison pour un produit ;
- compléments de commande : les compléments de commande correspondent au pied de la commande, c'est-à-dire, les différents frais (emballage, port, douane) ou les remises sur la totalité de la commande.

La navigation dans le cube sur la vue informationnelle en descendant vers les niveaux plus particuliers de la dimension de généralité se fait en utilisant le diagramme d'objets. Ce diagramme permet de représenter des instances particulières d'une commande, de client, de produits, ainsi que les relations entre eux.

Comme dans la vue fonctionnelle, les objets peuvent se trouver dans un état particulier à un instant donné. Par contre, il n'y a pas de moyen de caractériser les états de manière unique, ces états seront propres aux métiers de l'entreprise. Par exemple, toutes les entreprises ne gèrent pas automatiquement de la même manière l'état d'une commande. Sur le diagramme de la figure 8.14, il n'y aura pas possibilité de faire une facturation partielle tant que la livraison n'est pas totale.

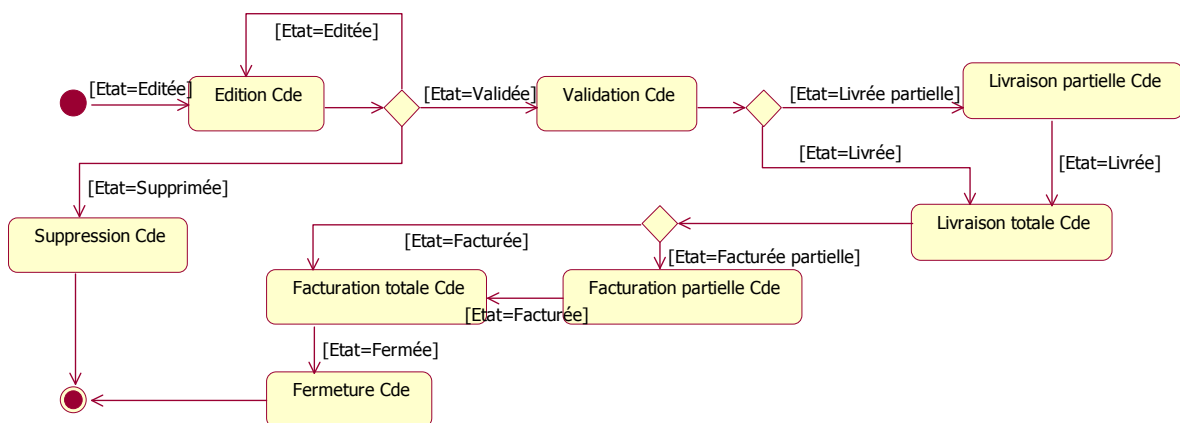


FIG. 8.14 – État de l'objet « commande » de l'entreprise - diagramme d'états transitions

Ce diagramme permet de représenter les différents états stables d'une commande. Lorsqu'une commande rentre dans un état, des actions sont déclenchées automatiquement (phase de vérification : au niveau de la programmation, il s'agit des pré-conditions). Lorsqu'une commande se trouve dans un état, il est possible d'exécuter certaines actions. Lorsqu'une commande sort

d'un état, des actions sont déclenchées automatiquement (phase de validation : au niveau de la programmation, il s'agit des post-conditions). Il est aussi possible de déclencher des actions sur les transitions (passage entre états pouvant donner lieu à des synchronisations d'activités).

Comme pour les processus, cette représentation permet de voir les différents états de l'objet (la situation de telle ou telle commande) et de définir des tableaux de bord offrant une vision globale du carnet de commande.

8.2.1.3 Point de vue des ressources

Les éléments de notre méta-modèle (fig. 8.3) qui composent ce point de vue sont décrits par le schéma partiel ci-dessous (fig. 8.15).

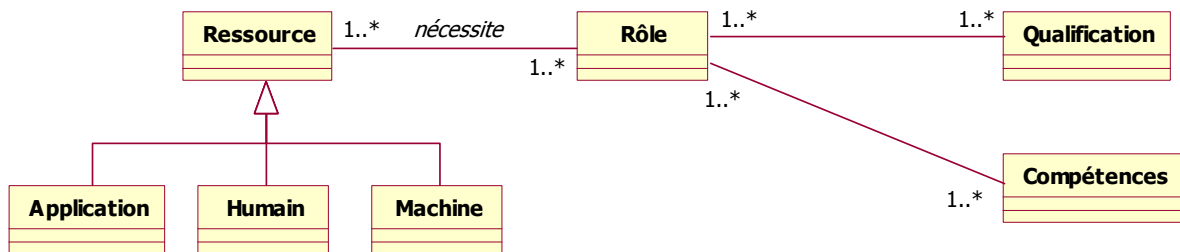


FIG. 8.15 – Méta-modèle partiel : point de vue des ressources

Selon Pierre Alain Muller (1999), « un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système ». Il existe différents types d'acteurs :

- les acteurs principaux : les personnes qui utilisent les fonctions du système ;
- le matériel externe : le matériel incontournable qui fait partie du domaine de l'application ;
- les autres systèmes : les systèmes extérieurs avec lesquels le système doit interagir.

Nous proposons donc une forme de représentation des ressources humaines et matérielles de l'entreprise à travers des classes particulières stéréotypées « acteur ». Nous utiliserons une forme particulière du diagramme des cas d'utilisation pour décrire les liens existant entre les différents acteurs. Il est possible d'utiliser la relation d'héritage entre les acteurs pour exprimer une dépendance hiérarchique (fig. 8.16). Les acteurs y sont représentés par une icône spéciale symbolisant le stéréotype. Qu'est ce que cela signifie concrètement ? Le concept d'acteur dans UML est orienté vers la description des liens entre utilisateur (ou moyen) et fonction remplie par le système (d'où le terme : cas d'utilisation). Dans notre exemple, cela signifie que la « direction » a les mêmes droits sur des cas d'utilisation que « l'administration des ventes » ou le « gestionnaire des ventes ».

Lorsque nous relierons effectivement les acteurs aux cas d'utilisation, nous pointons les rôles (ou les droits) des acteurs (fig. 8.17) dans les différents processus de l'entreprise.

Cette notion d'acteur est exploitée dans le diagramme d'activités pour concrétiser la relation entre l'acteur (dans son rôle) et l'activité au sein d'un workflow. La notion de « couloirs » inhérente aux diagrammes d'activités permet également de faire jouer les rôles. Il est en effet possible d'affecter l'exécution des processus métiers ou des activités de manière unique à des acteurs. Ces affectations définissent le droit des acteurs à déclencher les processus métiers ou activités (fig. 8.18). Le diagramme de la figure 8.18 est à mettre en regard de celui de la figure 8.8, il fournit une représentation plus poussée du système.

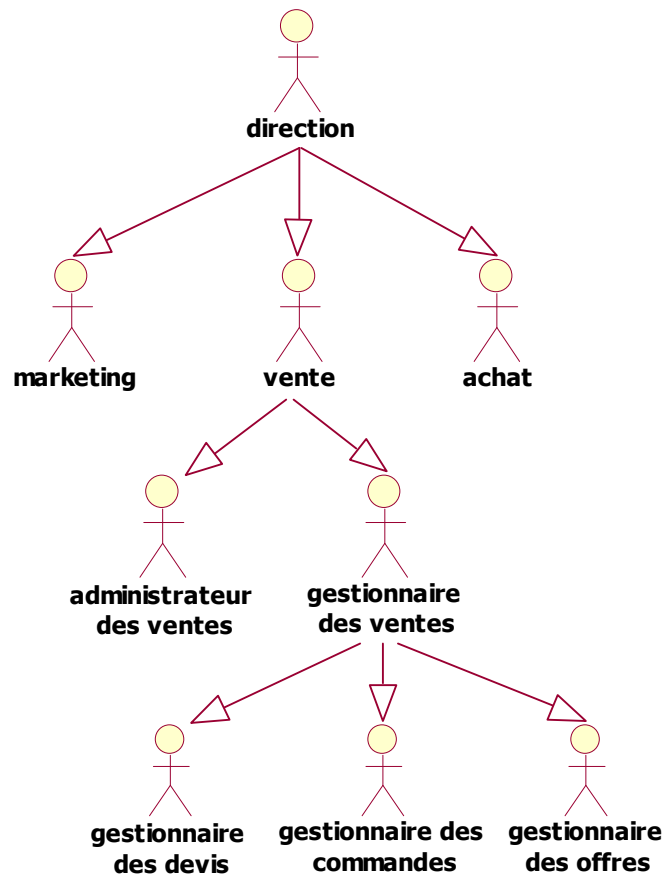


FIG. 8.16 – Hiérarchie de rôle

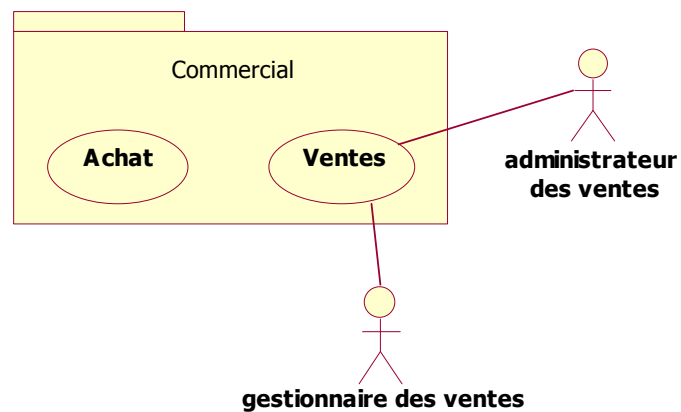


FIG. 8.17 – Relation entre les rôles et les processus domaine

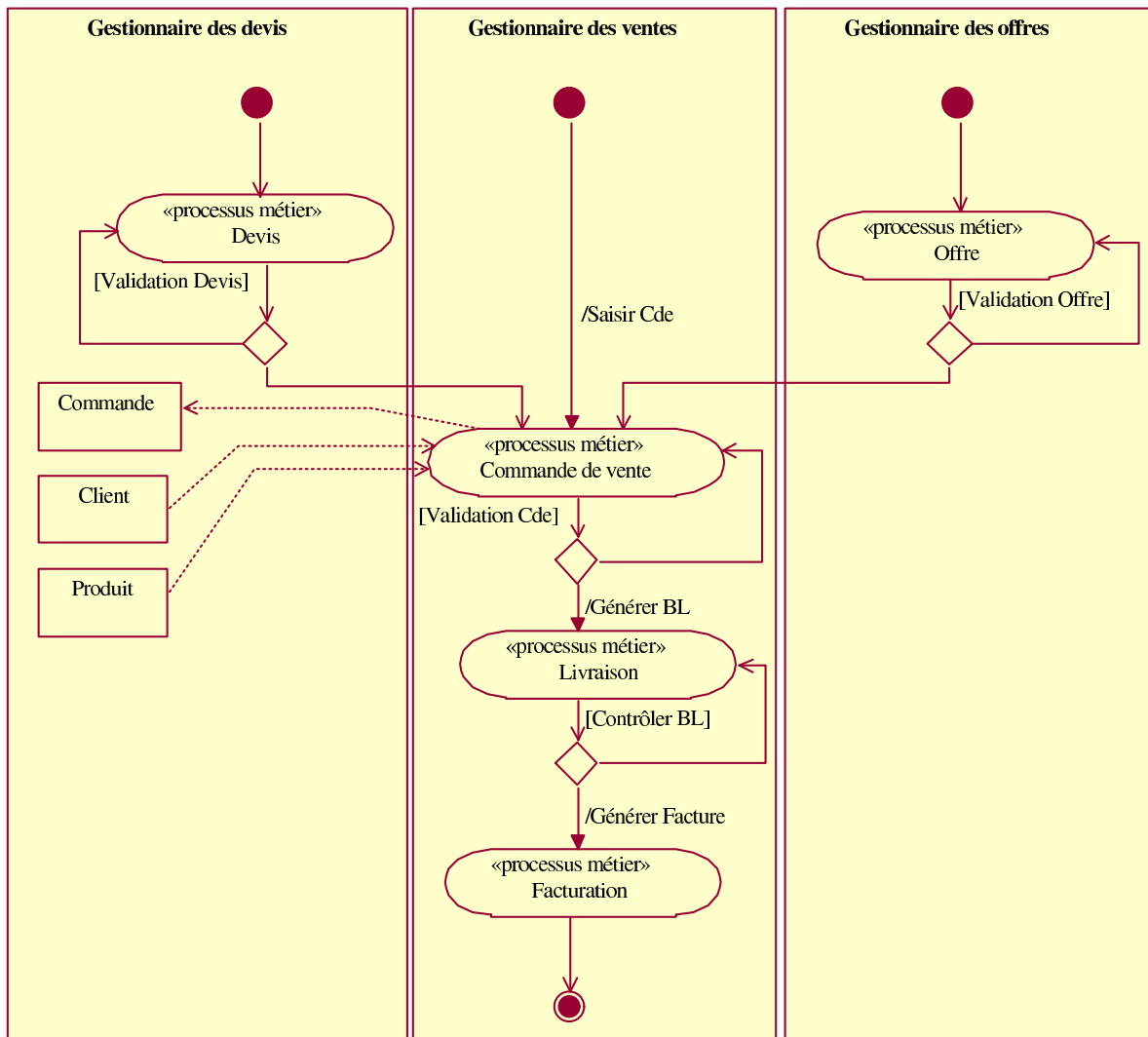


FIG. 8.18 – Relation entre les acteurs et les processus à l'aide du diagramme d'activité

Puisque l'acteur dans UML ne se réduit pas à l'homme, mais peut signifier un moyen mis en œuvre, nous complétons notre présentation par la figure 8.19 où une hiérarchie des machines est fondée sur deux catégories : les machines de production servant à la fabrication de pièce mécanique et les machines de contrôle permettant de vérifier la conformité de la pièce. Cela permet de faire une illustration plus classique du point de vue des ressources.

Ces machines devront être affectées à des activités de production. Il est pratique de leur attribuer un état et nous utilisons encore un diagramme états-transitions pour représenter les trois états possibles d'une machine particulière (fig. 8.20).

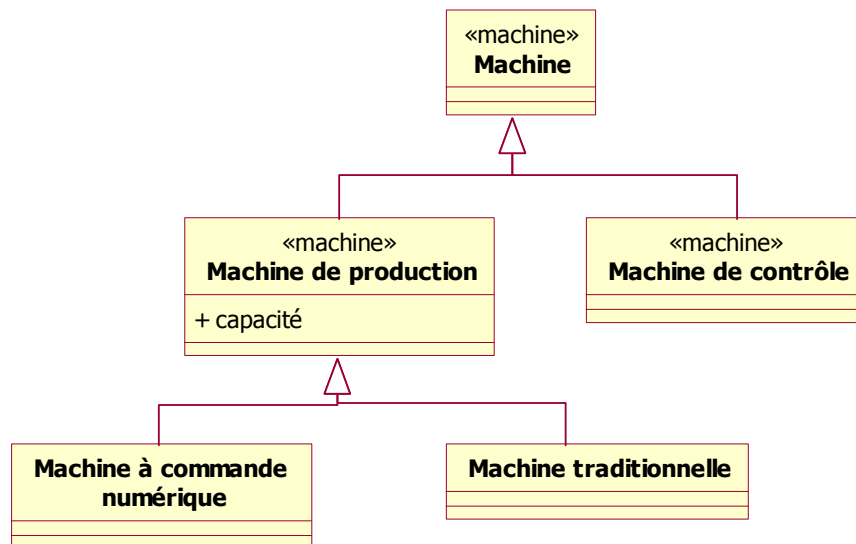


FIG. 8.19 – Hiérarchie des machines de l'entreprise

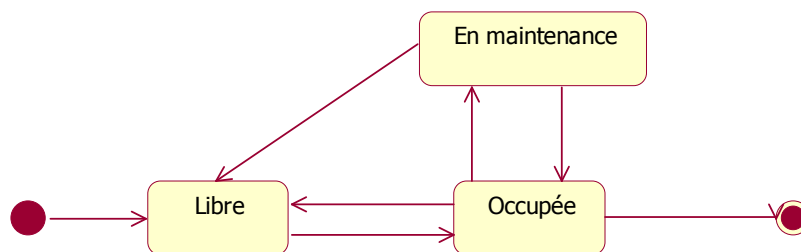


FIG. 8.20 – État d'une ressource

8.2.1.4 Point de vue organisationnel

Les éléments de notre méta-modèle (fig. 8.3), qui composent ce point de vue, sont décrits par le schéma partiel ci-dessous (fig. 8.21). Il faut mettre une relation de composition récursive sur l'unité organisationnelle pour autoriser des hiérarchies.

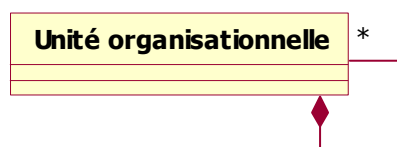


FIG. 8.21 – Méta-modèle partiel : point de vue organisationnel

La vue organisationnelle permet de décrire l'ensemble des entités représentant l'entreprise. Cette vue (fig. 8.22) permet de décrire la structure de l'entreprise, les différentes unités qui la composent (site de production ou site de vente, services commerciaux France ou export, ...).

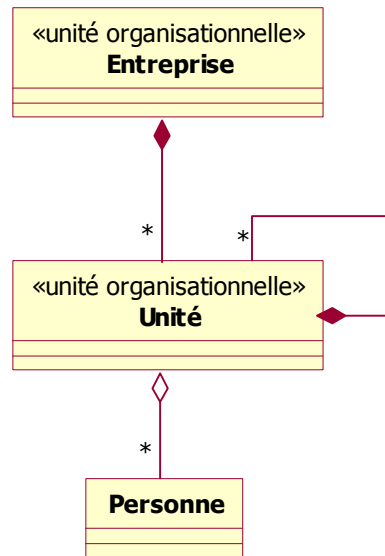


FIG. 8.22 – Structure de l'organisation

En naviguant selon la dimension de généricité, nous pouvons, à l'aide d'un diagramme d'objets (fig. 8.23), montrer une organisation particulière obéissant à cette structure.

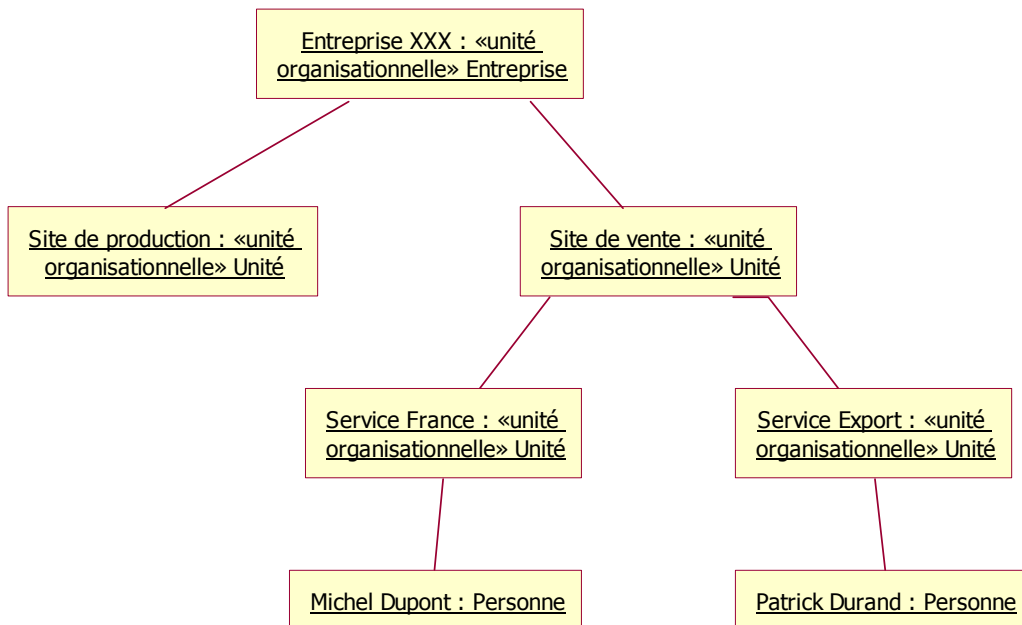


FIG. 8.23 – Exemple d'organisation tiré de la structure précédente

8.2.2 Synthèse

En parcourant chaque point de vue d'un modèle tel que préconisé dans la norme ENV 40003, nous avons traduit des constructs de notre méta-modèle en éléments du langage UML, puis nous avons élaboré les modèles eux-mêmes à l'aide de séries de diagrammes particuliers ((Franck Darras et al., 2002)). Le tableau 8.5 décrit les correspondances entre les éléments de notre méta-modèle et les éléments du langage UML.

<i>Méta-modèle</i>	<i>UML</i>
Vue fonctionnelle	Diagramme de cas d'utilisation Diagramme d'activités Diagramme d'états-transitions Diagramme de classe Diagramme d'objet
Domaine	Paquetage
Processus domaine	Cas d'utilisation, classe stéréotypée
Processus métier	Activité, classe stéréotypée
Activité	Activité, classe stéréotypée
Évènement	Évènement
Vue informationnelle	Diagramme de classe Diagramme d'objet Diagramme d'états-transitions
Objet d'entreprise	Classe et objet
Vue des ressources	Diagramme de cas d'utilisation Diagramme d'activités Diagramme de classe Diagramme d'objet Diagramme d'états-transitions
Ressource	Acteur Classe et objet
Rôle	Cas d'utilisation et acteur Couloir du diagramme d'activités Classe et objet
Vue organisationnelle	Diagramme de classe Diagramme d'objet
Unité organisationnelle	Classe et objet

TAB. 8.5 – Table de correspondance entre notre méta-modèle et les éléments du langage UML

8.3 La grille GRAI comme élément fédérateur de l'architecture fonctionnelle

La méthode GRAI est fondée sur un cadre conceptuel où la décision joue un rôle central. La vie de l'entreprise est effectivement rythmée par des décisions prises à tous les niveaux (stratégiques, tactiques, opérationnels). Les événements qui déclenchent des processus de tous ordres (métier, pilotage et support) font partie de ces décisions. Il est clair que la circulation des flux d'information est directement corrélée à l'orchestration des processus par l'intermédiaire des décisions.

Les grilles GRAI ont la réputation de fournir une cartographie générale des décisions basée sur deux paramètres : les fonctions (les colonnes de la grille) et le temps découpé en horizon (les lignes de la grille). Nous allons détourner le concept en injectant des processus métier à l'intérieur des centres de décision. Une telle manœuvre implique que les processus puissent effectivement être classés selon la portée de leur finalité en classes de processus stratégiques, tactiques et opérationnels. Grâce à cette modification, nous définissons un cadre général où les cas d'utilisation peuvent être positionnés sans équivoque dès lors qu'ils constituent le point d'entrée de la représentation du processus. Ensuite, les pratiques habituelles de lecture des diagrammes UML prennent le relais pour permettre de découvrir ou d'exploiter le modèle d'entreprise. Ce modèle est bien sûr conçu selon les règles d'utilisation d'UML présentées précédemment.

Il existe deux formes de grille. La grille de pilotage est dédiée au système de pilotage et en cela, elle se concentre sur la programmation des activités futures. Il faut dimensionner les demandes sur les produits (de la matière première aux produits finis) et les besoins en ressources productives, et réussir à planifier les activités de manière cohérente. La grille fonctionnelle est dédiée au système opérant, elle se concentre sur la mise en œuvre des plans et cherche à assurer leur exécution. Les colonnes sont directement associées à des fonctions macroscopiques qui forment la chaîne de valeur du système selon une philosophie très popularisée actuellement par les chaînes logistiques : c'est le maillon de la chaîne qui va des achats de matières premières jusqu'à la mise à disposition des produits vendus chez le client en passant par les livraisons des fournisseurs, la production, les stockages et la distribution.

Nous formulons l'hypothèse suivante : « la grille de pilotage fait référence à la catégorie des processus de pilotage, alors que la grille fonctionnelle fait référence à la catégorie des processus opérationnels et des processus supports. »

Nous allons maintenant examiner comment utiliser le langage UML dans le but de se rapprocher de la capacité d'expression de la grille GRAI dans sa forme naturelle. Cette discussion est menée sur un exemple de grille de pilotage décrite sur la figure 8.24. Les principes discutés sur cette grille s'appliquent à une grille fonctionnelle.

	Infos Externes	Gérer les produits		Planifier la production	Gérer les ressources	Gérer les ventes
		Acheter	Approvisionner			
H=1 an P=1an	Prévisions de ventes par famille	Politique d'achat	Politique d'apros	BUDGET	Plan Equipements et Effectifs	Elaborer Plan Marketing
H= 6 mois P= 1 mois	Prévision de commandes	Ajuster les marchés	Ajuster les paramètres d'apros	Plan de production	Ajuster les capacités	Recherche / Négociier marché (CRM)
H=1an P=6mois	Carnet de Commandes	Relancer	Approvisionner	Plan de charge	Répartir les effectifs	
H=1an P=1mois	Commandes Fermes			Ordonnancement	Affectation des postes	
Temps Réel		Passer Commande	Enregistrer E/S	PDV	Fabriquer / Monter	Enregistrer les commandes

FIG. 8.24 – Exemple de grille de pilotage de la méthode GRAI

Sur cette figure, chaque centre de décision est situé à l'intersection d'une ligne et d'une colonne. La seconde colonne de la grille fait exception puisqu'il ne s'agit pas d'une fonction, mais d'informations provenant de l'environnement de l'entreprise.

Un centre de décision s'apparente à un sous-ensemble de l'organisation où va être résolu un problème de calcul des valeurs de variables de décision dans le but de concourir à la performance par la poursuite d'objectifs propres au centre de décision, et sous la double influence de contraintes portant sur ces variables de décision et de critères exprimant la performance. Ces variables de décision fixent le degré d'autonomie des personnes responsables appartenant à ce sous-ensemble de l'organisation.

Les flèches simples décrivent des transferts d'informations entre les centres de décision. Les flèches doubles indiquent une relation de dépendance fonctionnelle où les décisions d'un centre (source) imposent des objectifs à un autre centre (cible). C'est la notion de cadre de décision.

Le « détournement » de la grille peut suivre deux voies distinctes :

- placer les cas d'utilisation à l'intérieur des centres de décision, un cas d'utilisation ne pouvant se situer que dans un seul centre. Cette voie privilégie l'aspect organisationnel en partant du rôle du décideur, vu comme un acteur dont le rapport au système est l'exercice de ses compétences pour prendre des décisions dans un cadre délimité. Rappelons que le déclenchement d'un processus ou d'une activité peut être considéré comme une décision particulière. La grille est un diagramme de cas d'utilisation dessiné de manière particulière,
- assimiler le centre de décision à un centre d'activités avec une logique de regroupement par colonne obéissant à la poursuite d'un objectif propre au type d'activité qu'elle inclut. Cette voie privilégie l'aspect fonctionnel en se basant sur les échanges ou interactions entre activités et suit principalement une logique de déroulement le long de la chaîne de valeur évoquée plus haut. Les informations trouvent naturellement leur place dans cette vision de la grille. La grille est un diagramme d'activités dessiné de manière particulière.

8.3.1 La grille des cas d'utilisation

Chaque colonne de la grille GRAI est un paquetage (concept de base pour ranger un ensemble d'éléments du modèle qui obéissent à une certaine unité logique) dont l'identifiant correspond à la fonction native de la grille GRAI. Les centres de décision sont représentés par des cas d'utilisation. N'oublions pas que dans UML, ces cas d'utilisation sont des classes stéréotypées et qu'en tant que classe, ils acceptent des attributs. Pour qualifier les horizons et périodes d'un centre de décision, nous avons donc utilisé systématiquement la notion d'attributs. Pour chaque cas d'utilisation, nous renseignons ces deux attributs.

Les liaisons décisionnelles sont représentées par les flèches en pointillés munies du stéréotype « cadre de décision », tandis que les liaisons informationnelles sont représentées par des flèches simples.

La figure 8.26 montre le résultat obtenu par l'application de ces règles sur la grille de pilotage de la figure 8.24. Le dessin est organisé en plaçant les cas d'utilisation :

- selon une position horizontale déterminée par l'égalité de valeur des attributs,
- selon une position verticale fixée par les paquetages et selon un ordre décroissant des valeurs des attributs,

permettant de retrouver la forme du tableau à deux dimensions de la figure 8.24.

8.3.2 La grille des activités

Dans cette version alternative, une activité correspond à un centre de décision. Le diagramme d'activités permet de représenter simplement les différents enchaînements entre les activités. Une colonne de la grille GRAI est assimilée à une macro-activité recouvrant des activités de nature comparable. A cette macro-activité, nous avons affecté le stéréotype « entreprise mission » pour exprimer cette nature comparable. Les activités circonscrites par cette macro-activité sont munies du stéréotype « entreprise activité ». Pour représenter les informations externes qui influent sur un centre de décision, nous avons utilisé directement les objets concernés en les reliant simplement aux activités.

La description des périodes et horizons liés aux centres de décision ne peut plus être faite par des attributs dans un tel diagramme. Nous avons fait le choix de les décrire explicitement dans le stéréotype de chacune des activités. Par exemple, l'identifiant « entreprise decision, H3a, P1a » correspondra à un horizon de 3 ans et à une période de 1 an.

Une activité, centre de décision, permettra d'encapsuler un réseau d'activités, décrit à l'aide du diagramme d'activité, représentant les processus dépendants afin de répondre à l'objectif du centre de décision.

La figure 8.27 montre le résultat obtenu par l'application de ces règles sur la grille de pilotage de la figure 8.24. Le dessin est organisé en plaçant les activités :

- selon une position horizontale déterminée par le stéréotype appliqué à l'activité,
- selon une position verticale fixée par les macro-activités, et selon un ordre décroissant des valeurs des attributs,

permettant de retrouver la forme du tableau à deux dimensions de la figure 8.24.

8.3.3 Bilan comparatif

Le tableau 8.6 récapitule les deux formes de modèle UML imaginés pour « détourner » la grille GRAI vers la cartographie de processus.

Formalisme GRAI	Définition	Voie 1	Voie 2
Grille GRAI	Permet de mettre en évidence les principaux liens décisionnels de l'organisation étudiée	Diagramme de cas d'utilisation	Diagramme d'activités
Colonne de la Grille	Fonction de l'entreprise	Paquetage	Méta-activité avec le stéréotype « Mission »
Lignes de la Grille	Niveau de prise de décision symbolisé par un Horizon et une Période	Utilisation des attributs affectés à chaque cas d'utilisation	Utilisation de stéréotypes de type « Hxa » et « Pxa » où 'x' représente la valeur et 'a' l'unité de temps

continue à la page suivante...

...suite de la page précédente

Centre de décision	C'est l'ensemble des activités de décision appartenant à un même niveau (H/P) pour une même fonction	Cas d'utilisation	Activité avec le stéréotype « entreprise activité »
Cadre de Décision	Ordres qui permettent d'assurer la fonction (symbolisé par une flèche à double trait). Il contient les informations nécessaires à la décision	Relation portant le stéréotype « Cadre Décision »	Transition portant le stéréotype « Cadre Décision »
Information	Représente les flux d'informations importants pour un centre de décision	Non défini	Relation à un objet

TAB. 8.6 – Bilan des choix de modélisation

Il semble qu'il n'y ait pas de perte sémantique forte en suivant chacune des deux voies. Seul le lien à l'information donne un petit avantage à l'approche par activités. Toutefois, cet avantage est partiellement détruit par la lourdeur de gestion des stéréotypes dont l'identifiant doit être adapté à chaque niveau (H/P).

8.3.4 Enchaînement des diagrammes UML pour la modélisation d'un système

Nous rappelons l'enchaînement classique, la navigation habituelle, entre les diagrammes dans la prise de connaissance d'un modèle UML. Cet enchaînement est synthétisé sur la figure 8.25.

À partir d'un cahier des charges, il est possible de déterminer les principaux cas d'utilisation du système étudié. Un cas d'utilisation peut correspondre à :

- un scénario unique. Dans ce cas, le diagramme d'activités va décrire l'enchaînement des activités qui le compose,
- un ensemble de scénarios possibles. Un diagramme de séquence correspond à chaque scénario.

Les informations ou éléments de diagrammes, sont relatives à des objets :

- les objets nécessaires pour atteindre les objectifs des cas d'utilisation,
- les objets définis dans les diagrammes d'activité (objet résultant d'une action, d'un acteur ou du système),
- les objets identifiés pour la réalisation des différents scénarios des cas d'utilisation. Ces objets sont représentés par le diagramme de séquence. Nous pouvons aussi identifier les opérations de ces mêmes objets.

À partir de ces différentes catégories, et en appliquant des principes de classification, il est possible de construire le diagramme de classes.

Compte tenu de cette démarche où le cas d'utilisation précède le diagramme d'activités, d'une part, et de la bonne équivalence constatée entre les deux formes de modélisation, d'autre part, notre préférence va à la représentation par les cas d'utilisation (Voie 1).

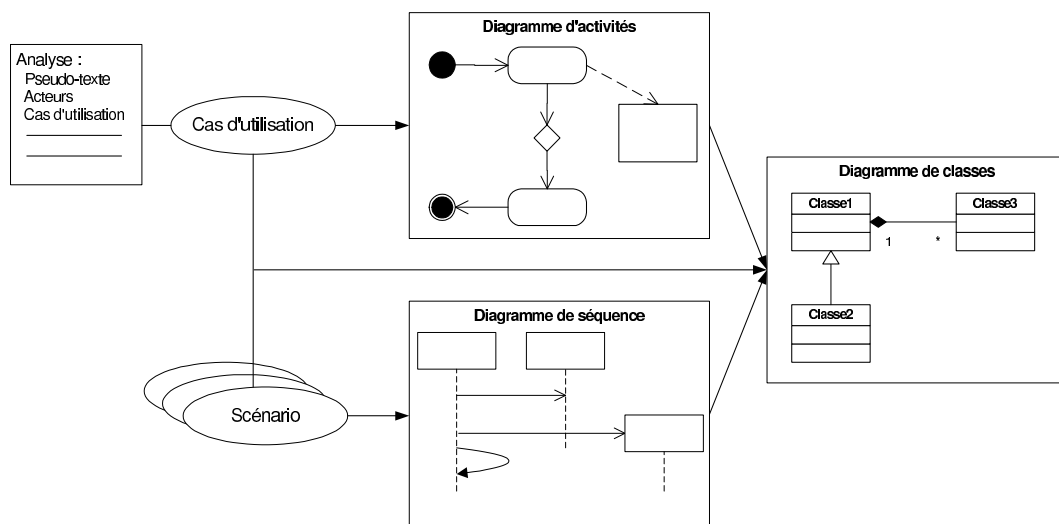


FIG. 8.25 – Ordre de consultation des diagrammes UML

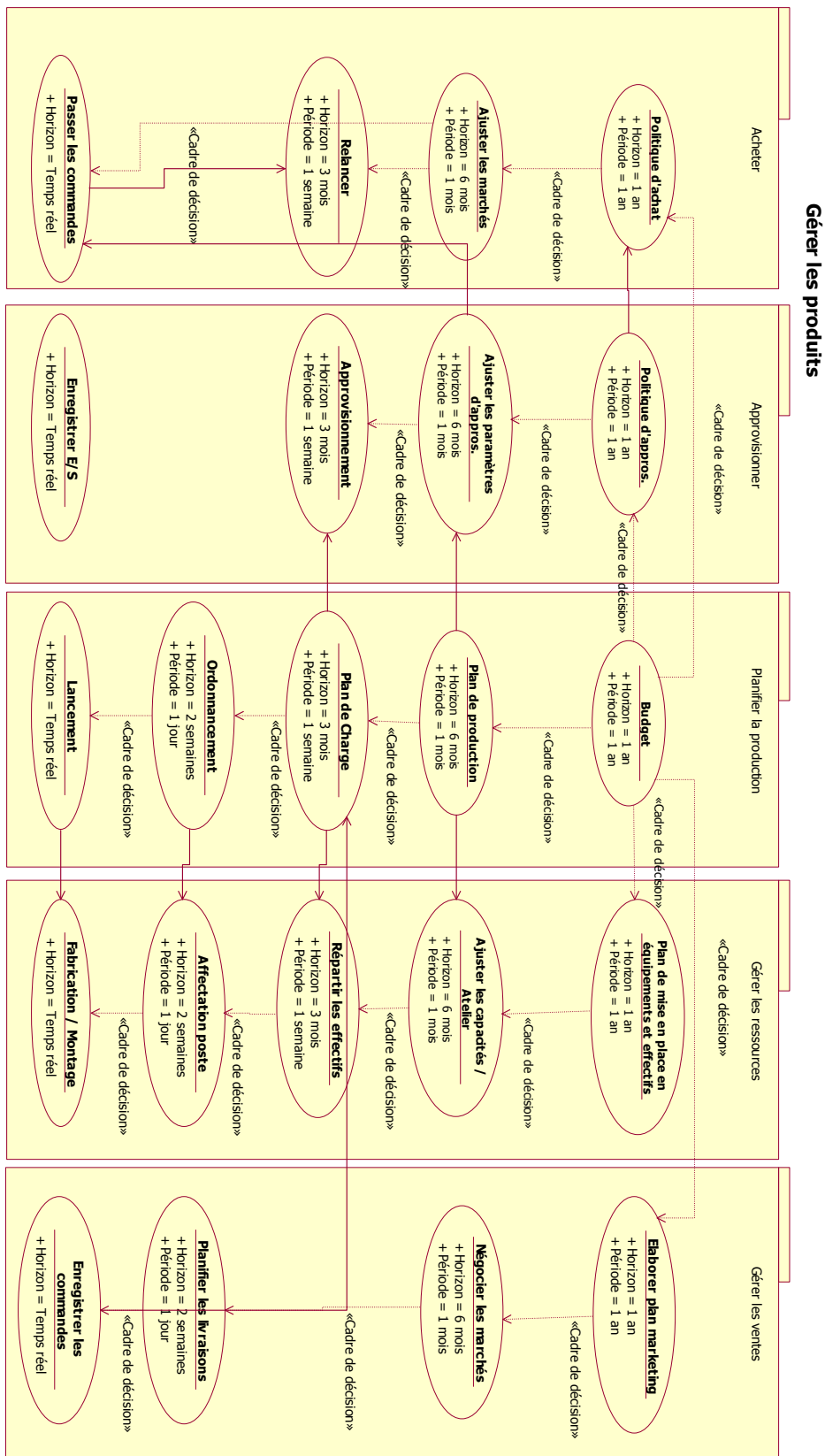


FIG. 8.26 – Grille GRAI à l'aide du diagramme des cas d'utilisation organisé en packages

8.4 Conclusions

Le langage UML est doté de neuf diagrammes dont la syntaxe est définie par la norme 1.4. Chaque diagramme apporte une vision partielle du système. Il est nécessaire de grouper ces diagrammes dans une structure qui guide leur emploi. Philippe Kruchten (1995) parle d'un modèle d'architecture dit des 4+1 vues.

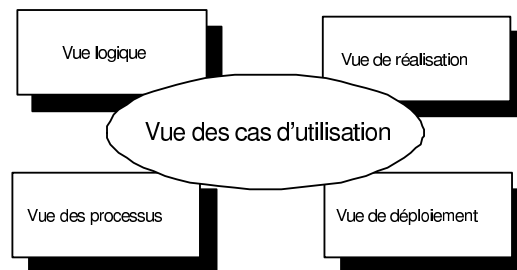


FIG. 8.28 – Représentation du modèle d'architecture 4+1 vues

Dans ce modèle d'architecture, nous retrouvons les notions d'architecture logique (§ 5.2.1) et fonctionnelle (§ 5.2.3.1), représentées par la vue logique et vue des processus, et la notion d'architecture physique (§ 5.2.3.2), représentée par la vue de réalisation et la vue de déploiement. Tous ces points de vue sont reliés par la vue des cas d'utilisation qui assure la cohérence.

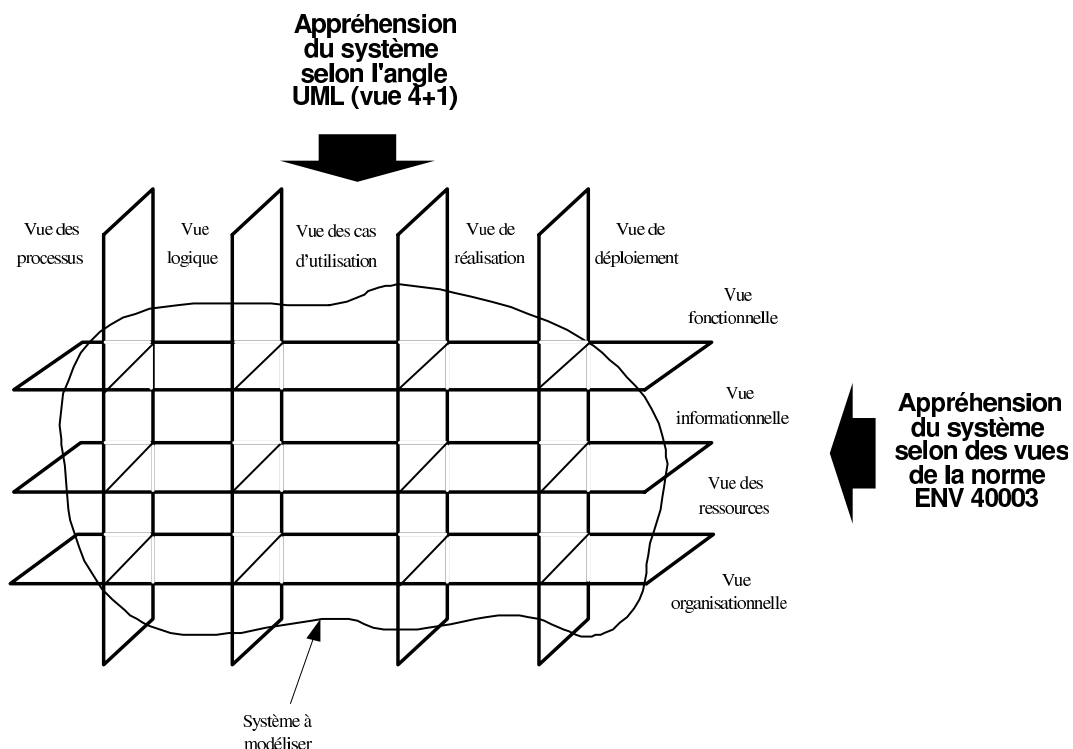


FIG. 8.29 – Intersection entre le modèle 4+1 vue et les quatre points de vue de la norme

Dans ce chapitre, nous nous sommes attachés à montrer qu'il était possible d'utiliser le langage UML pour modéliser les points de vue dans les méthodes de modélisation d'entreprise sur trois vues parmi les cinq du modèle architectural 4+1. Partant d'un méta-modèle, nous avons donc utilisé les capacités syntaxiques de ce langage pour expliquer un modèle sur un cas d'école. Ensuite, et afin de renforcer la sémantique du modèle UML, nous avons utilisé intensivement

le concept de stéréotypes et avons essayé de trouver une démarche assimilable à celle de la méthode GRAI dans les premières étapes de la conception du modèle.

L'utilisation de stéréotypes appliqués systématiquement à des éléments des diagrammes UML conduit à la définition d'un « profil » en UML. L'OMG a déjà standardisé cinq à six profils dans divers domaines. Ces profils sont des « constructs » (éléments de base du formalisme UML) qui ont été adaptés à un univers spécifique. Il serait intéressant de pousser notre raisonnement pour mesurer la pertinence de nos choix vis-à-vis des initiatives de nouveaux profils adaptés à l'entreprise et supportés par l'OMG.

Chapitre 9

Cadre de référence

9.1 Mise en adéquation des modèles

Le modèle du cycle de vie d'un Système d'Information identifie invariablement la séquence des tâches suivantes :

- l'analyse et la spécification,
- le développement,
- l'exploitation et la maintenance de l'application.

Il nous semble original de décliner ces trois temps par rapport à deux points de vue ; celui du client et celui de l'éditeur. Un client cherche à acquérir une solution informatique pour faciliter la gestion de son entreprise. Il doit délimiter le périmètre d'action du logiciel et expliquer ses méthodes de gestion à l'intérieur de ce périmètre afin de mettre le logiciel en adéquation avec ses besoins. Toutefois, l'expérience montre que cette recherche d'adéquation est un exercice délicat : les méthodes de gestion de l'entreprise (§ 2.1.1) ne sont pas forcément facilement explicables, ou encore l'examen des méthodes standard codées dans le logiciel peuvent être considérées comme une meilleure façon de procéder pour l'industriel. Dans ce dernier cas, c'est alors l'entreprise qui doit s'adapter à de nouvelles méthodes et se mettre en adéquation avec le logiciel. Une autre source de difficulté peut provenir d'un manque de couverture du logiciel vis-à-vis des besoins de l'entreprise, ce que l'on désigne sous le terme de « trou fonctionnel ». Ce type de situation, qui est connu, conduit l'éditeur à proposer le développement d'un code spécifique intégrable à la solution standard pour être en adéquation avec le besoin.

La complexité des problèmes rend délicate l'écriture du cahier des charges. La part d'incertitude doit être minimisée dans la spécification détaillée du développement pour le client avant le prototypage. Les projets de ce type sont réputés non linéaires dans le sens où plusieurs itérations entre la phase d'analyse et de spécification, et celle de développement, peuvent être nécessaires pour atteindre un bon niveau de qualité.

Un éditeur de logiciel doit maîtriser le développement de son produit. Le cycle de vie du produit se renouvelle suivant les trois phases évoquées (début du § 9.1) pour chaque version du logiciel.

Il faut distinguer, d'une part, le cycle de vie de l'application cliente, et d'autre part, le cycle de développement de l'application par l'éditeur (§ 6.2.1). Ce dernier correspond à un ensemble d'étapes (analyse, développement, documentation, contrôle/test et validation) permettant de développer une application « globale » répondant à un ensemble de besoins des clients installés

(étant dans un contexte de maintenance évolutive) ou à installer (étant dans un contexte de déploiement d'une nouvelle version du produit). Pour l'éditeur, le caractère générique du logiciel est primordial. Chaque application client doit être déduite de l'application générique avec un minimum de développements spécifiques. C'est ce que l'on appelle le paramétrage qui doit constituer l'essentiel de la charge des services techniques.

Chaque nouvelle version doit assurer une non régression pour les applications installées. Elle doit aussi offrir des services nouveaux pour couvrir des attentes de clients sur des besoins non satisfaits auparavant. Ces nouveaux services sont perçus de plusieurs façons :

- factorisation des développements spécifiques,
- effets de mode en gestion,
- adaptation de méthodes existantes.

	Spécification	Développement	Exploitation
Cahier des charges	ME	ME	ME
Outil	MR	MR	MR

TAB. 9.1 – Correspondance Modèle d'Entreprise - Modèle de Référence

Sur le tableau 9.1, nous émettons l'hypothèse que les deux protagonistes sont en phase sur une période correspondant à la gestion d'une version stabilisée pour un client donné. Chaque client peut avoir une valeur de période différente. Nos travaux étudient les apports d'une coexistence entre un modèle de l'entreprise côté client et un modèle de l'application côté éditeur. Cette correspondance de modèles est utile sur les trois étapes du déroulement du projet et de la relation client/éditeur. Pour chaque étape, nous allons mettre en évidence la spécificité des apports d'une telle démarche, sur la base de l'expérience acquise auprès de notre partenaire industriel.

9.1.1 Analyse et spécification

L'étape d'analyse et de spécification est une phase de collecte d'informations. Ces informations sont nécessaires pour proposer une solution adaptée aux besoins du client.

9.1.1.1 Côté client

Actuellement, la démarche d'analyse de l'éditeur auprès du client est la suivante :

- description du mode de fonctionnement de la structure client,
- expression des besoins,
- étude de l'adéquation des besoins avec la solution logicielle,
- récupération des données existantes.

Nous souhaitons standardiser cette démarche et soutenir les chefs de projet à l'aide de documents normalisés. Le modèle d'entreprise propre au client sera développé au moyen d'un formalisme complet et adapté. Par capitalisation et retour d'expérience, il sera possible d'extraire des classes de modèles représentant des profils types de clients.

9.1.1.2 Côté éditeur

Nous considérons que le modèle de l'application existe chez l'éditeur. Si, de plus, ce modèle est basé sur un formalisme compatible avec le modèle d'entreprise, la pré-configuration de l'application se trouve facilitée. Cela signifie que :

- les descriptions des produits, des coûts et des procédés sont possibles par le logiciel,
- certains processus de gestion peuvent être repérés comme des fonctions programmées du logiciel.

Il y a donc naturellement possibilité d'imaginer des procédures de transition du modèle de référence de la solution logicielle vers le modèle d'entreprise pour assister la pré-configuration. De manière symétrique, des procédures de transition du modèle d'entreprise vers le modèle d'application mettront en évidence les Trous Fonctionnels Potentiels (TFP) et poseront les bases du développement d'applications spécifiques.

À la suite de cette analyse détaillée, il faut bâtir le paramétrage de l'application et coder les développements spécifiques. Cette pré-configuration de l'application cliente va permettre d'alimenter la grille de paramétrage et de faire le développement.

9.1.2 Développement

La généralité de l'application côté éditeur se concrétise par une facilité de paramétrage (obtenue par une base de données des paramètres). En d'autres termes, un certain nombre d'options permettant de caractériser les méthodes programmées, et donc les fonctionnalités disponibles, sont accessibles pour les techniciens chez l'éditeur. La partie du développement relative au paramétrage consiste à renseigner les valeurs de ces paramètres pour compléter la définition du modèle de l'entreprise cliente.

Puisque le développement est aussi une phase de travail sur les spécifiques, le modèle de référence permet de faciliter le travail d'intégration du nouveau code dans le code existant.

À l'issue de cette activité de développement, nous allons pouvoir alimenter la grille de validation. Cette grille de validation permet de tester la solution logicielle, incluant les développements spécifiques, avec les jeux de test du client.

9.1.2.1 Côté client

Un client possède son propre mode de fonctionnement, son univers de discours. La collecte d'informations pour le paramétrage doit être menée avec une dépense minimale d'énergie. Le modèle d'entreprise, lorsqu'il est mis en regard du modèle de référence, permet de faire la correspondance entre les données et les méthodes de gestion, et les paramètres de la solution logicielle.

L'utilisation de certaines fonctionnalités du logiciel peut amener le client à modifier son organisation et ses habitudes de gestion. La transition du modèle de référence vers le modèle d'entreprise permettra d'exprimer ces contraintes, et de faciliter la prise de décision du client sur la base d'un dialogue entre le client et l'éditeur.

Outre les valeurs des paramètres, l'existence du modèle d'entreprise permet de faciliter la mise au point de scénarios de tests qui serviront de validation pour la recette de l'application chez le client.

9.1.2.2 Côté éditeur

Le modèle de référence du futur système généré à partir du modèle d'entreprise facilite la compréhension du projet par le développeur. L'équipe projet de l'éditeur a donc un moyen de médiation à travers le modèle de référence. La distribution des tâches se trouve normalement facilitée.

Si, au cours du développement, une difficulté surgit, le modèle d'entreprise peut être utilisé pour chercher à modifier le modèle de référence de manière transparente pour le client et en minimisant la casse.

9.1.3 Exploitation et maintenance

La phase d'exploitation et de maintenance concerne plus particulièrement les services personnalisés qui peuvent être fournis au client par l'éditeur (maintenance hotline, maintenance préventive, maintenance évolutive).

9.1.3.1 Côté client

À l'aide de la grille de test et validation, nous avons une vision du modèle de la solution logicielle qui a été installée. Il est donc possible d'utiliser ces connaissances pour la formation des utilisateurs sur les modules installés, incluant les spécifiques du client.

Nous pouvons aussi, à l'aide de ces informations, fournir et maintenir une documentation personnalisée pour chaque client.

9.1.3.2 Côté éditeur

La grille de test et validation est une base de connaissance faisant office de référence dans le cas d'une assistance au client, ou pour une gestion de l'évolution de version plus aisée.

9.1.4 Synthèse

Nous synthétisons sur la figure 9.1 la coexistence des modèles d'entreprise et de référence de la solution, des deux côtés de la relation client/éditeur, fondement du projet.

Cette coexistence des deux modèles permet :

- de structurer le travail du projet en exploitant les possibilités de transition entre le modèle d'entreprise et le modèle de référence,
- d'obtenir un meilleur dialogue entre les experts métiers et les experts SI (Système d'Information),
- d'améliorer la qualité (délais, coûts et performance) globale de la relation entre les deux parties.

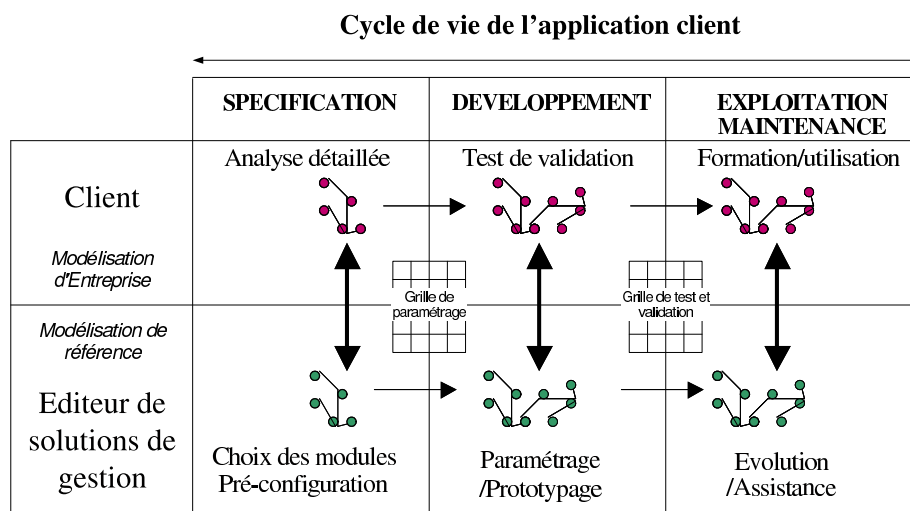


FIG. 9.1 – Mise en relation du modèle d'entreprise (côté client) et modèle de référence (côté éditeur)

9.2 Adéquation dans le cycle de vie

Nous avons vu dans le précédent chapitre qu'il était possible de modéliser les différents points de vue de la modélisation d'entreprise en utilisant le langage UML (Unified Modeling Language). Or nous avons également insisté sur l'usage des modèles dans les outils de développement logiciel ((Franck Darras et al., 2002)). Nous allons revenir sur ce point ci-dessous.

9.2.1 Décomposition d'un SI et utilisation d'UML

Nasser Kettani et al. (1999) ont proposé une démarche de modélisation en UML, au sein du projet SI, décrite de manière synthétique par le schéma de la figure 9.2.

Ils proposent d'aborder le projet de la manière suivante :

1. la description de l'environnement et des objectifs dans la mise en place d'un Système d'Information ;
2. la description du domaine (avec une hiérarchie de domaines si nécessaire) : les acteurs et les cas d'utilisation ;
3. la description de la structure statique : les classes et les objets nécessaires au fonctionnement du Système d'Information.

La représentation de la structure statique se fait à partir :

- des informations élémentaires nécessaires pour répondre aux différentes spécifications des deux phases précédentes ;
- des informations nécessaires pour la représentation de l'ensemble du modèle dynamique du Système d'Information.

Les différentes classes devront être réparties dans des catégories ou packages. Ce découpage établit une relation forte entre les classes et le domaine dans lequel elles sont principalement utilisées. Différentes dépendances seront créées pour pouvoir mettre en relation les différentes classes avec d'autres domaines où elles jouent un rôle.

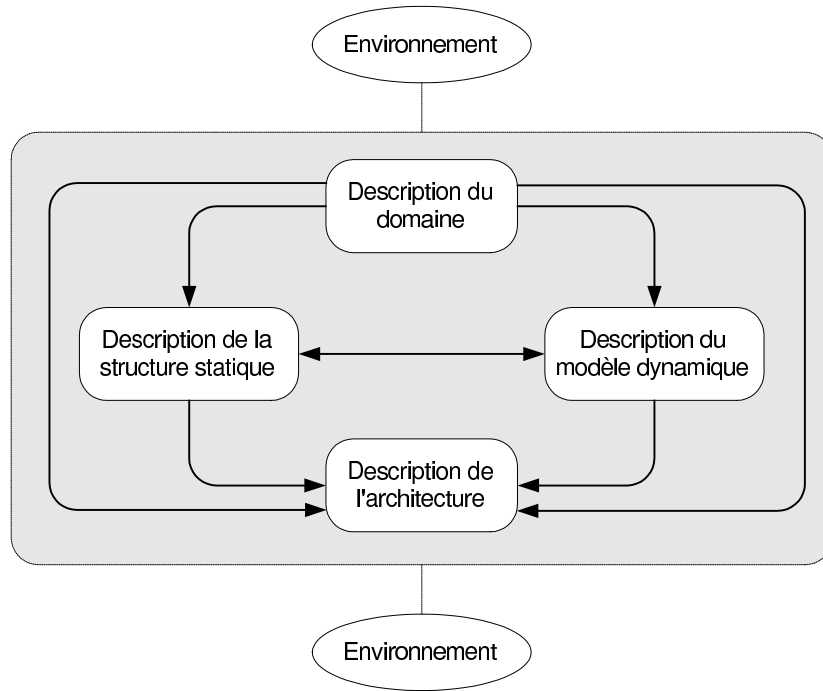


FIG. 9.2 – Organisation des différentes étapes de représentation via UML

Il est possible d'avoir des itérations entre la description de la structure statique et celle du modèle dynamique afin de permettre une optimisation des informations représentées et utilisées.

5. la description du modèle dynamique : le Système d'Information réagit et déclenche des actions de traitement de l'information par rapport à des sollicitations représentées par des événements. C'est la description d'un comportement temporel.

En UML, il faudra utiliser les formalismes des diagrammes d'interactions, et désigner l'état des objets par des diagrammes d'états.

6. la description de l'architecture du Système d'Information : l'implémentation et la distribution physique du système peuvent être décrites et organisées à l'aide de deux diagrammes d'architecture (composants et déploiement).

Nous pouvons synthétiser ce découpage à l'aide du tableau suivant (tab. 9.2).

Représentation	Diagramme UML
Besoins, domaine et environnement	Diagramme des cas d'utilisation
Structure statique	Diagrammes d'objet et de classe
Comportement dynamique	Diagrammes d'activité, de collaboration, d'états-transitions et de séquence
Architecture	de Diagrammes composants et de déploiement

TAB. 9.2 – Répartition des neuf diagrammes en classes de représentation du SI

9.2.2 Synthèse

L'utilisation de la modélisation d'entreprise permet de :

- normaliser ou standardiser la représentation de l'organisation,
- définir les processus standard de l'organisation,
- aider au paramétrage,
- améliorer la rapidité de mise en œuvre de l'ERP,
- réduire les coûts dus aux délais de paramétrage,
- améliorer l'analyse des besoins (adéquation entre les processus de l'entreprise et les fonctionnalités proposées de l'ERP),

L'utilisation du modèle de référence permet de :

- maîtriser le développement de l'application,
- maîtriser la complexité du système,
- maîtriser l'évolution du client,
- capitaliser les connaissances des besoins de l'ensemble des clients.

La réalisation du modèle d'entreprise impose à l'éditeur de prendre en compte certains éléments de l'organisation dans le modèle de référence de la solution logicielle. Si, dans chaque modèle, nous avons la présence d'éléments similaires, il doit être possible de faire une projection d'un modèle sur l'autre. Cette projection doit être réalisée au travers d'un cadre de référence (§ 9.3.2) et utiliser des mécanismes que nous détaillerons dans les prochaines sections.

Nous pouvons synthétiser la répartition des différents diagrammes par rapport aux vues de la modélisation d'entreprise dans le schéma suivant (fig. 9.3).

Cycle de vie de l'application client			
	SPECIFICATION	DEVELOPPEMENT	EXPLOITATION MAINTENANCE
Client <i>Modélisation d'Entreprise</i>	Vue fonctionnelle		Diagramme dynamique
	Vue informationnelle		Diagramme de classe et d'objet
	Vue des ressources		Diagramme de cas d'utilisation
	Vue organisationnelle		Diagramme de classe et d'objet
Editeur de solutions de gestion <i>Modélisation de référence</i>	Besoins, domaine, environnement		Diagramme de cas d'utilisation
	Structure statique		Diagramme de classe et d'objet
	Comportement dynamique		Diagramme dynamique
	Architecture		Diagramme de composant et déploiement

FIG. 9.3 – Mise en relation et utilisation des différents diagrammes

9.3 Démarche projet

Une démarche projet est une démarche :

- de bon sens,

- de communication : principes et vocabulaire,
- d'efficacité,
- d'adaptabilité.

La démarche projet fixe le cadre de communication entre le maître d'ouvrage et le maître d'œuvre au sein du projet. Notre méthode de conception cherche à tirer un maximum de connaissances de la mise en correspondance des modèles, et conduit logiquement le projet vers son objectif ((Franck Darras et al., 2003b)). Après une synthèse et une analyse critique de quelques démarches existantes, nous proposerons notre propre vision des projets ERP.

9.3.1 Présentation de l'existant

9.3.1.1 Démarche d'analyse des Systèmes d'Information

Chantal Morley et al. (2000) proposent une démarche fondée sur le langage UML pour accompagner le projet ERP. Le modèle est utilisé pendant l'étape de déploiement. Elle tisse une relation entre les diagrammes UML et les différentes tâches du déploiement, que nous avons résumé dans le tableau suivant (tab. 9.3). Cette approche montre la capacité d'UML à fournir des formalismes appropriés pour nos besoins de modélisation.

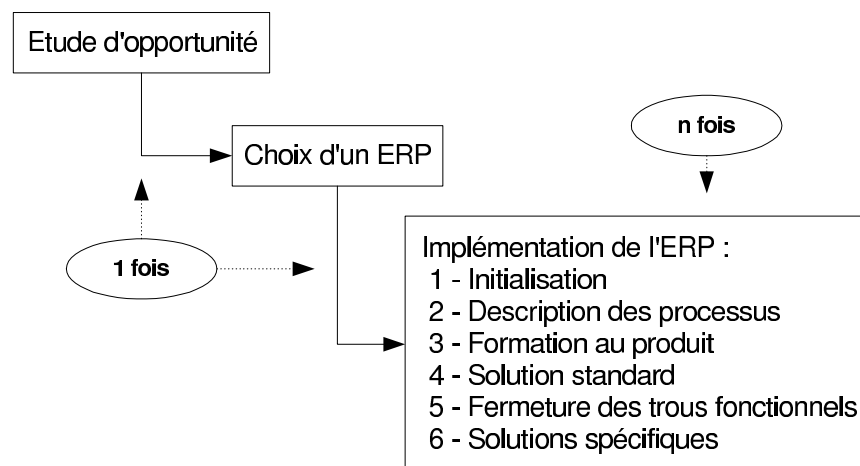


FIG. 9.4 – Démarche dans la mise en œuvre d'un Système d'Information

L'étape « étude d'opportunité » amorce le début du projet et en définit les grandes lignes. Cela induit deux choix : l'un sur les investissements, l'autre sur le périmètre à couvrir.

Dans la deuxième étape, le choix de l'ERP, il s'agit de faire la sélection d'un progiciel correspondant le mieux aux critères de l'entreprise, qu'ils soient stratégiques, fonctionnels, technologiques, techniques ou commerciaux.

Dans la troisième étape, l'implémentation de l'ERP permet de découvrir, paramétrer, adapter le progiciel retenu en vue de sa mise en œuvre. Chantal Morley et al. (2000) ont travaillé sur cette étape en utilisant les diagrammes UML comme support de modélisation des différentes phases d'implémentation. Nous résumons également ces étapes et les diagrammes appropriés dans le tableau 9.3.

Étapes, phases et tâches du cycle d'un projet ERP	Apports d'UML
Étape Étude d'opportunité	
Étape Choix d'un ERP	
Étape Implémentation de l'ERP	
Phase Initialisation	Taxonomie des entités
Phase Description des processus	Diagramme de collaboration Diagramme de classes (entités) Typologie des processus Diagramme d'états-transitions Diagramme de séquence Diagramme d'activités
Phase Formation au produit	
Phase Solution standard	
Tâche Adéquation	
Tâche Configuration	Cas d'utilisation
Tâche Prototype	
Tâche Validation	
Tâche Simulation en grandeur réelle	Cas d'utilisation Scénarios
Phase Fermeture des trous fonctionnels	
Phase Solution spécifique	
Phase Préparation de la mise en œuvre	

TAB. 9.3 – Apports d'UML aux différentes étapes, phases et tâches d'un projet ERP d'après (Chantal Morley et al., 2000)

9.3.1.2 Démarche développée lors des travaux de thèse de Boutin

Dans ses travaux de thèse, Pascal Boutin (2001), a proposé une démarche de mise en correspondance de modèles qui est décrite sur la figure suivante (fig. 9.5).

Dans la phase de modélisation, il représente l'entreprise de façon macroscopique, sans aller dans le détail des processus. Ceci doit permettre d'identifier les principaux processus et leurs éventuels dysfonctionnements, et de construire une base de modélisation suffisante pour la sélection d'un ERP (AS IS).

Dans la phase d'expression des besoins, les besoins de l'entreprise sont identifiés en termes d'évolution organisationnelle et de couverture fonctionnelle pour l'ERP. Dans la phase de définition du modèle futur (TO BE), l'évolution de l'organisation est analysée en tenant compte du potentiel de l'ERP choisi. Selon Boutin, la richesse des progiciels est telle que les contraintes ne portent plus sur ce que l'on peut faire, mais sur le « comment faire ». Dans la phase de prototypage, il s'agit de définir le fonctionnement de l'ERP suivant le modèle du futur système par rapport au modèle de l'ERP.

Les différentes propositions de Boutin associent bien l'analyse de l'entreprise et les possibilités de l'ERP par l'intermédiaire de la modélisation. Il a astucieusement placé la phase de modélisation d'entreprise (ASIS) en amont du choix de l'ERP, permettant ainsi l'utilisation du modèle du système futur (TO BE) et du modèle de l'ERP afin de déduire une solution adéquate du Système d'Information de l'entreprise.

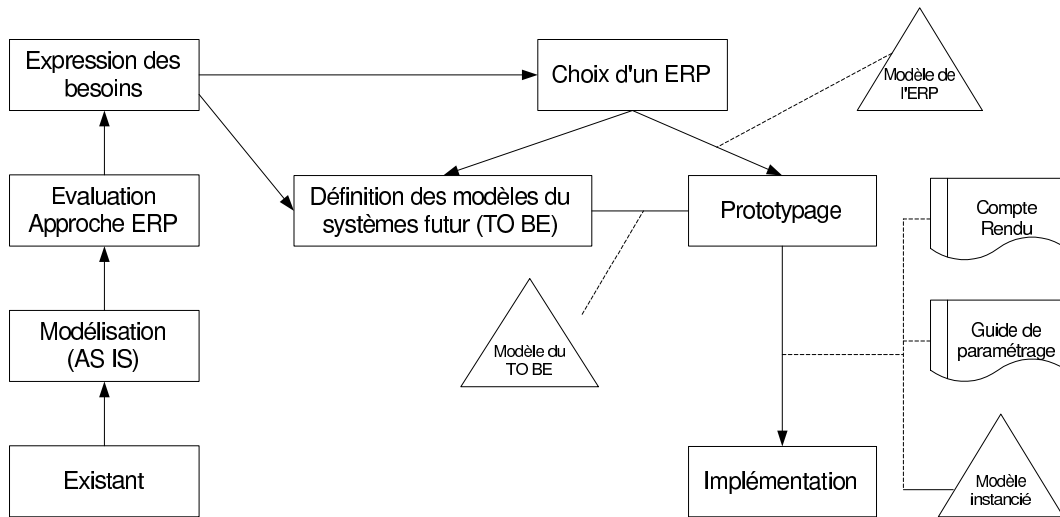


FIG. 9.5 – Démarche de mise en œuvre proposée par Pascal Boutin

Ces travaux fixent une grande partie des principes de la méthode que nous recherchons. Toutefois, dans la mise en pratique de ses idées, Boutin n'a pas proposé d'imposer les mêmes formalismes aux différents modèles pour minimiser la charge de travail de l'équipe projet. Il n'y a pas d'efficacité dans la mise en correspondance du modèle d'entreprise et du modèle de référence tant que l'unification de ces modèles n'est pas assurée.

9.3.1.3 Projection entre deux modèles

Certains travaux comme ceux de Keith A. Butler et al., (Keith A. Butler et al., 1999) et (Ali Butler et al., 2000), se concentrent sur la convergence du modèle d'entreprise et du modèle de référence pour en déduire les composants utilisables pour la conception d'un Système d'Information dans un processus RAD (Rapid Application Development).

Dans le cadre de développement de SI, ils définissent une méthode itérative pour coordonner le modèle d'entreprise et le modèle de l'application. Le chemin, décrit sur la figure 9.6, permet de faire dialoguer les experts métiers et les informaticiens.

Un modèle des processus métiers (BP - Business Process) est décliné sur la branche haute de cet arbre, alors que celui de l'application « AS-IS » est sur la branche basse parallèle. Les experts métiers font évoluer le modèle BP suivant les changements désirés. En suivant ces modifications, une correspondance est effectuée sur le modèle de l'application pour rester en phase avec le besoin. Une bibliothèque de composants est à la base de l'architecture applicative, ce qui facilite les gestions des modifications.

Cette projection a été appliquée en utilisant, dans un premier temps, le formalisme IDEF3, (formalisme de la méthode IDEF, annexe C) pour représenter le modèle d'entreprise, et dans un deuxième temps le formalisme UML pour représenter le même processus. Outre le manque d'informations techniques détaillées sur les publications connues de nous, à notre connaissance, il n'existe aucune information sur la poursuite de ces travaux et les résultats acquis.

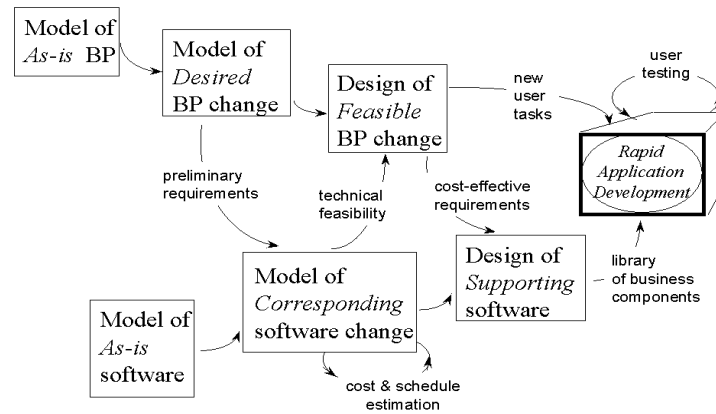


FIG. 9.6 – Convergence de modèles

9.3.1.4 Synthèse

Sur la figure 9.7, nous montrons comment ces différentes sources d'inspiration ont participé à l'ébauche de notre proposition.

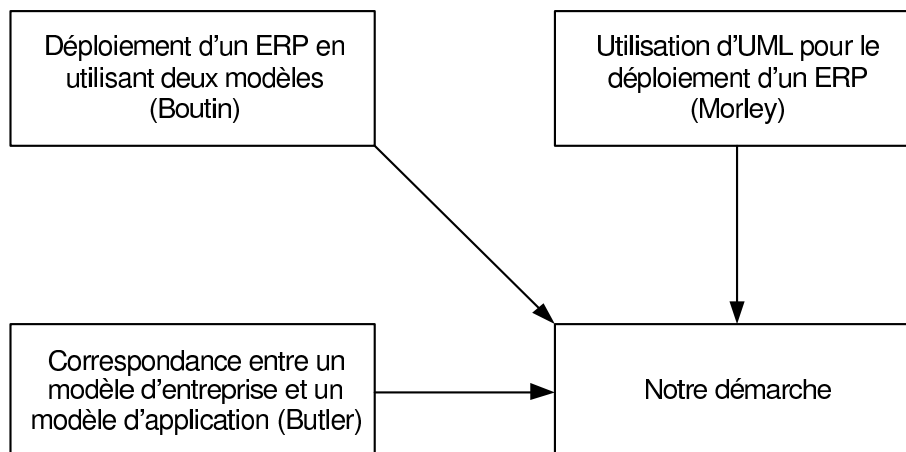


FIG. 9.7 – Positionnement de notre démarche par rapport à l'existant

9.3.2 Cadre de référence

Dans notre démarche, nous définirons les points suivants (Franck Darras et al., 2003b) :

- les dimensions utiles pour notre cadre de référence,
- le cadre de référence et le placement des différents modèles,
- les différents mécanismes mis en œuvre pour passer d'un modèle à l'autre.

9.3.2.1 La base du cadre

La dimension du « degré de généralité » correspond au niveau de détail auquel on associe le modèle. Il y a trois niveaux : le niveau générique, le niveau partiel et le niveau particularisé.

C'est la reconnaissance d'une démarche structurée dans la construction du modèle où il est possible de travailler par généralisation (du plus bas niveau vers le plus haut), ou par spécialisation (du plus haut vers le plus bas). C'est un mécanisme qui est un fondement de la démarche orientée objet, notamment avec les liens d'héritage ((Pierre Alain Muller, 1999)).

La dimension de « l'abstraction » fait référence à l'utilisation du modèle en introduisant trois niveaux :

- le modèle de spécification qui définit les opérations à exécuter dans un contexte donné d'application, en termes d'opérations, d'informations, de ressources nécessaires, de responsabilités et d'autorités sans aucune référence à des options ou des décisions d'implémentation ;
- le modèle conceptuel précise les opérations qui seront effectivement réalisées avec les entités de l'entreprise qui seront mises à contribution ;
- le modèle d'implémentation décrit les règles et les moyens pour exécuter les opérations en respectant les spécifications.

C'est une dimension qui est bien connue dans le domaine de l'ingénierie des Systèmes d'Information et qui est devenue familière aux praticiens de la méthode MERISE ((Hubert Tardieu et al., 1998)). Dans MERISE, les modèles de données et de traitement sont, en effet, déclinés en trois niveaux (conceptuel, logique, physique) qui correspondent aux trois niveaux ci-dessus. Un mécanisme de dérivation successive permet de passer du niveau le plus élevé au niveau le plus bas. Dans une approche orientée objet, c'est le mécanisme d'instanciation de classes qui permet de passer du niveau de spécification au niveau conceptuel.

9.3.2.2 Le cadre

La norme de modélisation d'entreprise CEN 40003 sert de cadre à notre proposition de conception. Nous nous sommes limités à un plan formé par les bases citées ci-dessus, l'axe de généralisation et l'axe d'abstraction, pour expliciter la démarche. L'axe des points de vue est pris en compte dans les diagrammes UML utilisés.

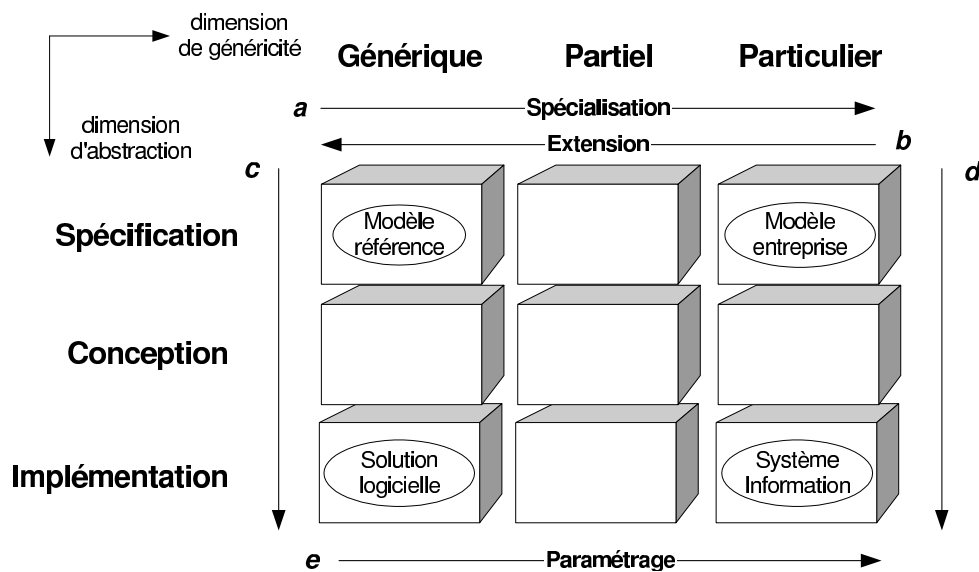


FIG. 9.8 – Notre premier cadre de conception

À partir de ce constat, nous avons défini notre cadre de référence comme le montre la figure fig. 9.8. Nous avons positionné sur les neuf faces de ce cadre nos modèles et nos applications. Nous retrouvons le modèle d'entreprise (particulier et spécification) en face du modèle de référence de l'application (générique et spécification).

Pour passer d'une face à l'autre, nous avons défini un certain nombre de mécanismes. Pour passer du modèle de référence au modèle d'entreprise, nous avons défini un mécanisme de spécialisation (**a**) permettant la particularisation et la réduction du modèle de référence pour obtenir le modèle de l'entreprise cible. Si des trous fonctionnels sont détectés, nous avons défini le mécanisme d'extension (**b**) permettant d'étendre le modèle de l'entreprise vers le modèle de référence de l'application. Si ce cas se produit, nous avons défini le mécanisme (**d**) permettant l'analyse, la conception et l'implémentation de ces trous fonctionnels.

Bien entendu, l'application logicielle a déjà été développée à partir du modèle de référence de l'éditeur en utilisant le mécanisme (**c**). Lorsque les modèles sont finalisés, nous pouvons les utiliser pour obtenir le Système d'Information de l'entreprise cible. Pour cela, la solution logicielle possède des paramétrages permettant de traduire un choix au niveau projet en une valeur au niveau du modèle, puis un (ou des) paramètre(s) au niveau applicatif.

Mais ce premier cadre de référence présente deux inconvénients :

- la démarche ne convient peut être pas à tous les types de PME/PMI pour des raisons de charge de travail en conception détaillée, il faut introduire une approche avec des solutions « pré-formatées » ;
- cette démarche ne met pas assez l'accent sur l'influence réelle de la modélisation d'entreprise sur le modèle de référence, car le modèle de référence doit posséder une architecture proche de celle du modèle d'entreprise pour réussir à établir une communication par mise en correspondance d'éléments similaires.

À partir de ce constat, nous avons décidé de modifier légèrement la représentation pour obtenir un cadre plus réaliste vis-à-vis de cette cible de clients.

9.3.2.3 Nouvelle proposition de cadre de référence

Dans ce cadre de référence (fig. 9.9), nous avons fait le choix de rajouter un niveau supplémentaire à la dimension d'abstraction. Nous avons nommé ce nouveau niveau « spécification métiers » car ce niveau se préoccupe plus particulièrement du fonctionnement stratégique, opérationnel ou organisationnel de l'entreprise.

Au niveau « spécification métiers », nous avons représenté uniquement les concepts de la modélisation d'entreprise, et à un niveau générique. Ces concepts influencent maintenant le modèle d'entreprise et le modèle de référence de la solution logicielle. Au niveau partiel, nous avons le modèle d'entreprise métier représentant une vision globale de l'entreprise et influençant la réalisation du modèle de référence. À ce niveau partiel, nous avons aussi autant de modèles d'entreprise que de typologies d'entreprise, que ce soit des entreprises de production ou des entreprises de services, des entreprises ayant une gestion point de commandes ou des entreprises ayant une gestion à l'affaire...

La démarche est maintenant la suivante :

- les mécanismes de spécialisation (**a**) et d'extension (**b**) permettent de faire évoluer le modèle d'entreprise (niveau « spécifications métiers ») ;

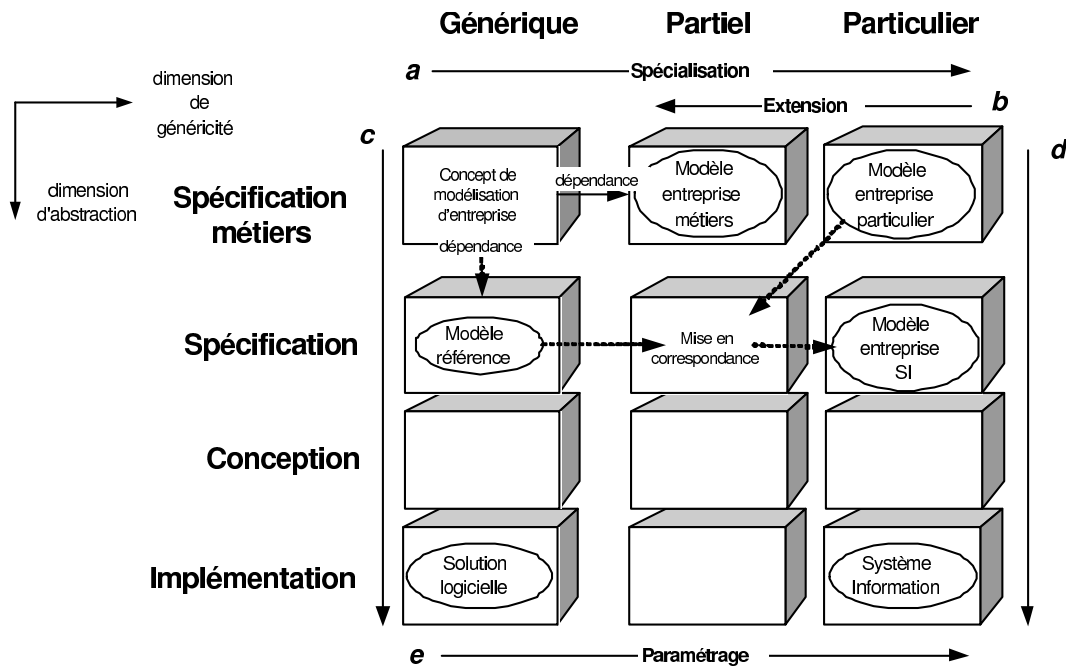


FIG. 9.9 – Notre second cadre de conception

- le mécanisme d'implémentation (**d** et **c**) permet de passer des différents modèles à une solution implémentée. Le mécanisme d'implémentation (**c**) permet le développement de l'application avant le déploiement auprès d'un client. Le mécanisme d'implémentation (**d**) permet de développement des trous fonctionnels (§ 9.6), si nécessaire ;
- le mécanisme de paramétrage (**e**) permet de passer d'une solution logicielle au Système d'Information de l'entreprise.

Nous décrirons plus précisément ces mécanismes au moyen d'exemples dans une prochaine section (§ 9.5).

9.4 Exemple

9.4.1 Démarche suivant la taille de l'entreprise

Les PME/PMI forment un spectre d'entreprises de tailles très variables (§ 3.1). Il est donc nécessaire d'adapter notre cadre de référence à des classes d'entreprises cible. Nous allons utiliser la taille de l'entreprise, généralement corrélée avec la taille du projet, pour présenter trois démarches possibles d'utilisation du cadre de référence.

9.4.1.1 Démarche pour une petite structure

Le nombre de journées d'analyse, pour exprimer correctement les besoins de l'entreprise, est réduit lors du déploiement d'un Système d'Information dans une petite PME/PMI. De plus, pour ces entreprises, le Système d'Information peut être utilisé pour organiser les services de celle-ci.

L'éditeur/intégrateur doit trouver rapidement la solution la plus en adéquation avec le fonctionnement du client. Il y a une spécialisation du modèle de référence pour déterminer le modèle d'entreprise de SI. Cette projection permet simplement d'obtenir les principaux processus spécifiques à l'entreprise. Nous utiliserons, tout de même, des connaissances, définies au niveau des spécifications métiers, basées sur une typologie d'entreprise (mécanique, électronique ou à la commande, à l'affaire, ...) pour faciliter la définition du modèle de référence du Système d'Information de l'entreprise.

L'utilisation du niveau de spécification métier n'aura pas d'influence dans la phase de déploiement du Système d'Information. Nous utilisons le cadre de référence en adaptant les mécanismes de déploiement, comme décrit sur la figure 9.10.

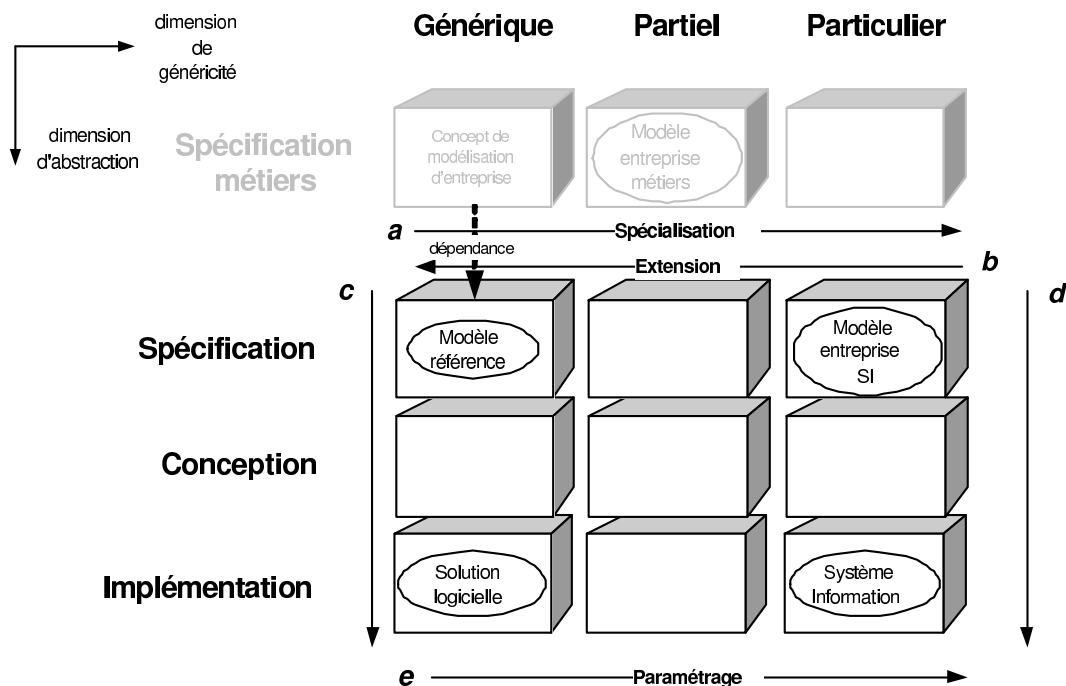


FIG. 9.10 – Démarche d'utilisation du cadre de référence pour une petite structure

9.4.1.2 Démarche pour une structure moyenne

Dans le cadre d'une PME de plus grande taille, les projets informatiques sont plus importants et les organisations plus aptes à prendre en charge une certaine complexité de projet. Suivant la taille du projet, l'intégrateur a la possibilité d'avoir plus de journées d'analyse permettant de définir plus précisément le fonctionnement de l'entreprise et par conséquent de réaliser le modèle particulier de l'entreprise.

L'utilisation du cadre va se distinguer de celle de la petite structure par la réalisation de ce modèle d'entreprise. Le travail réalisé sur les processus sera plus fin, et surtout plus spécifique. On passe en quelque sorte d'une représentation en figures imposées à une représentation en figures libres.

9.4.1.3 Démarche pour une grande structure

Dans les entreprises de type PME de grande taille, l'organisation est dotée des moyens humains requis pour ce type de projet, avec en particulier : une direction information, une assurance qualité, un contrôle de gestion.

La plupart de ces entreprises ont mis en place une certification d'assurance qualité. Les nouvelles normes d'assurance qualité ISO 9000 version 2000 sont tournées vers la définition et la représentation des processus de l'entreprise. Par conséquent, un bon nombre de processus de l'entreprise sont déjà connus et archivés. On peut croire qu'ils sont entretenus et donc représentatifs de la réalité opérationnelle au quotidien.

Lorsque l'entreprise souhaite déployer un Système d'Information, l'intégrateur a en sa possession une connaissance de base, avec un formalisme obéissant aux spécifications des normes qualité qui n'obéit pas aux bons principes évoqués au chapitre 7. Sans traduction de cette connaissance pour retrouver ces bons principes, la mise en correspondance entre le modèle d'entreprise particulier de l'entreprise et le modèle de référence de la solution logicielle sera très difficile. Par conséquent, pour mettre en correspondance ces deux modèles, un travail de conversion est entrepris. Il est nécessaire de passer du modèle ISO spécifique à l'entreprise à un modèle avec le formalisme UML. Il s'agit d'un travail long et méticuleux qui mériterait des supports méthodologiques, voire des outils adaptés, mais qui sort du périmètre de nos travaux.

9.4.2 Gestion de projet dans le déploiement d'un ERP

Nous connaissons la décomposition structurelle du travail en une arbre de tâches dans un projet ERP (§ 4.2.1). Nous avons présenté les deux principales étapes d'un tel projet, chacune des étapes étant composée de plusieurs macro tâches. Cette organisation peut être résumée à l'aide du schéma suivant (fig. 9.11) décrivant à la fois l'organisation et l'enchaînement de ces tâches.

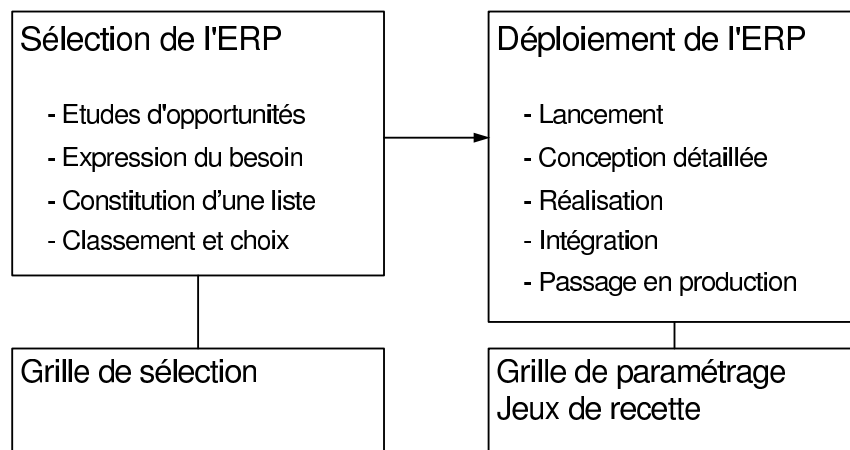


FIG. 9.11 – Macro tâche d'un projet ERP

Notre conduite de projet propose un enchaînement un peu différent des différentes tâches, principalement dans la sélection de l'ERP. Nous estimons que les grilles de sélection, fournis par le CXP ou le CETIM, ne permettent pas de mettre l'accent sur les besoins de l'entreprise et sur l'implication des fonctionnalités non couvertes de la solution logicielle. De plus, c'est l'éditeur qui répond aux questions, tandis que la MOA (Maîtrise d'OuvrAge) doit miser sur la

bonne foi de ce dernier pour les points qui ne sont pas pris en compte par le jeu de tests, puis appliquer des pondérations entre les réponses pour réaliser son classement des propositions. Cette démarche souffre d'un manque d'accès à la définition technique de chaque outil pour permettre de juger la cohérence de la réponse.

La communication, pour être efficace, doit permettre au client d'aller lui-même puiser, en utilisant un modèle, les réponses aux questions qu'il se pose à travers une représentation formelle des capacités du progiciel, d'où le recours à un modèle de référence appréhendé comme un modèle de Systèmes d'Information.

Le problème qui se pose alors au client réside dans l'utilisation d'un formalisme qu'il ne maîtrise pas forcément. Nous avons vu dans les précédents chapitres qu'il existe bon nombre de formalismes de représentation pour la modélisation d'entreprise comme pour la modélisation de solutions logicielles. La faisabilité du bon déroulement de notre conduite de projet est donc conditionnée par la réduction des formalismes de modélisation employés et par la mise à disposition d'un maximum d'informations, via le modèle de la solution logicielle, de la part des éditeurs.

Il est donc nécessaire, dans un premier temps, lors de l'étape de sélection, de proposer au client des éléments de représentation de haut niveau, compréhensibles sans être nécessairement un informaticien. Ces éléments permettent, par un mécanisme de spécialisation, de définir le périmètre fonctionnel, d'identifier les principaux processus de l'organisation, ainsi que les objets métiers jouant un rôle dans la solution logicielle (classes candidates). Une seconde passe, basée sur un mécanisme d'extension, permet de voir s'il existe des manques dans la solution déployée.

Dans un deuxième temps, lors de l'étape de déploiement, l'analyse se veut plus détaillée que dans l'étape de sélection. Le problème de la méthode reste pourtant quasiment le même. En effet, il s'agit toujours d'exprimer un besoin et de trouver les solutions dans le progiciel en proposant un bon paramétrage des fonctions standard. Nous allons donc pouvoir utiliser des éléments du formalisme UML de plus bas niveau. Pour aider le client à modéliser ses besoins, il est nécessaire de mettre en place une équipe projet. Du côté du client, nous retrouvons les personnes en charge d'exprimer ces besoins, de caractériser les processus de l'entreprise, de montrer l'organisation de celle-ci, et du côté de l'éditeur (principalement pour les PME/PMI), nous retrouvons des chefs de projet, ayant des connaissances aussi bien en génie industriel qu'en génie logiciel, capable d'assimiler ces informations et de les transformer afin de réaliser le modèle de la solution finale. C'est le chef de projet qui aura la maîtrise de la modélisation des besoins détaillés du client et qui aura la charge de faire valider les résultats par le client.

Nous proposons donc deux itérations dans l'utilisation de notre cadre de référence. La première itération permettra de mettre en concurrence les différents modèles des solutions logicielles et de choisir l'ERP adéquat. Lors de la deuxième itération, il s'agira de détailler plus précisément le modèle par rapport à l'organisation de l'entreprise.

9.4.2.1 Sélection de l'ERP

Nous proposons donc d'utiliser notre cadre de référence pour la sélection de l'ERP. Lors de cette première itération, nous utilisons les mécanismes de spécialisation et d'extension pour adapter le modèle d'entreprise au fonctionnement de l'entreprise cible par rapport à son savoir-faire (automobile, électronique, ...) et à son mode de gestion (à la commande, à l'affaire, MRP, SCM, ...). Pour accélérer l'application des mécanismes, nous utilisons une base de connaissances fournissant des parties de modèles. Ces modèles sont définis pour le niveau partiel de la dimension de généricité.

9.4.2.1.1 Première phase

La première phase (fig. 9.12) représente le premier mécanisme de spécialisation pour obtenir le modèle de l'entreprise cible (modèle entreprise particulier) à partir :

- soit des concepts de modélisation d'entreprise,
- soit des modèles partiels spécifiques à un métier.

Afin d'accélérer la conduite du projet de déploiement, nous proposons une base de connaissance fournissant un ensemble de référentiels représentant des modèles d'entreprise suivant son savoir-faire et son mode de gestion.

Nous proposons d'utiliser les grilles GRAI pour présenter les différentes typologies d'entreprise. Cette base est utilisée par l'intégrateur pour obtenir, comme le dit Boutin, l'existant de l'entreprise, le modèle AS-IS.

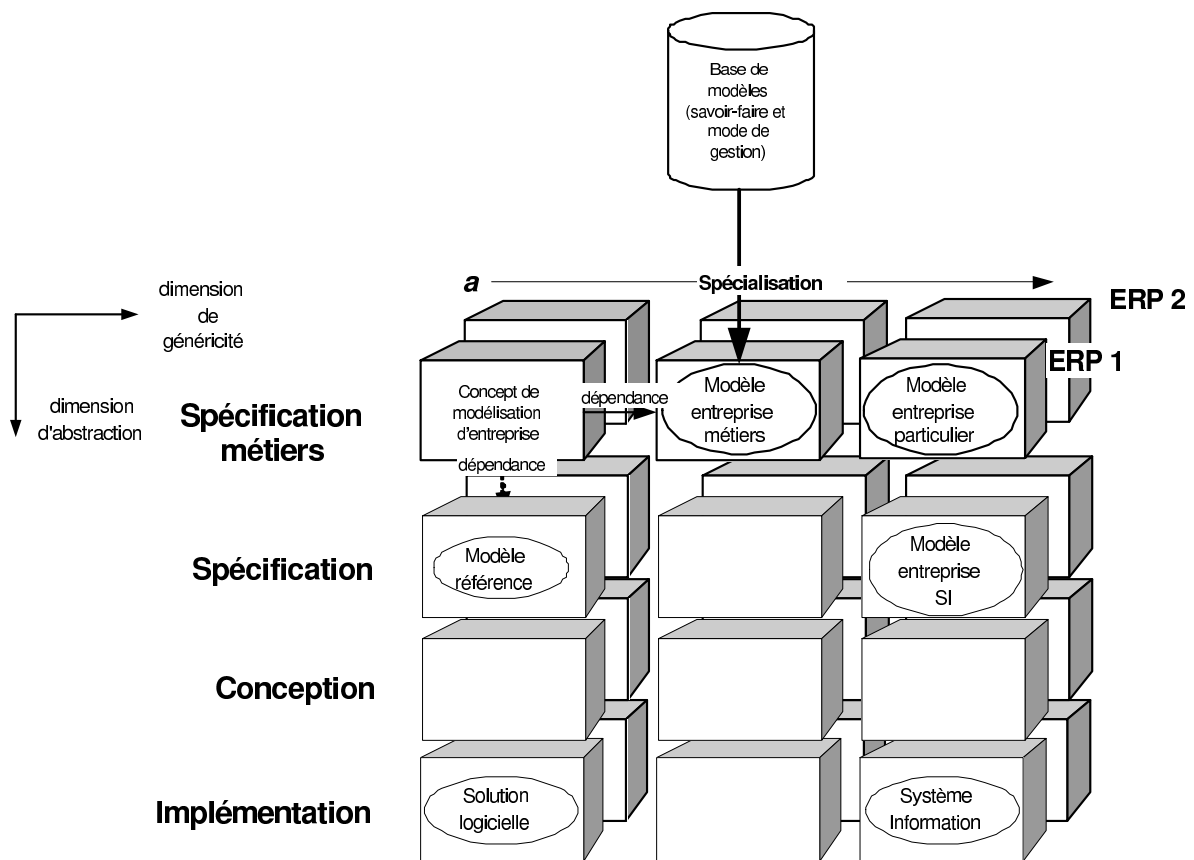


FIG. 9.12 – Conduite de projet « Première phase »

Cette démarche doit permettre de mettre en concurrence deux éditeurs (ERP1 et ERP2) de solutions logicielles distinctes. L'obtention du modèle particulier de l'entreprise doit permettre d'obtenir le taux de couverture de la solution logicielle d'un éditeur par rapport au modèle obtenu. Le travail sur les spécifications métiers est fait une fois pour tous les niveaux de spécifications, un niveau par éditeur.

9.4.2.1.2 Deuxième phase

Le déroulement complet de cette deuxième phase est soumis à certaines conditions. Le mécanisme d'extension n'est nécessaire que dans le cas où il y a eu découverte d'un besoin non

couvert (TFP) par une fonctionnalité de la solution logicielle. Le client a alors deux possibilités :

- ignorer ce trou fonctionnel et adapter son organisation,
- demander à l'éditeur ou l'intégrateur de développer un spécifique relatif à son besoin.

Le développement de spécifique a bien entendu un impact sur la solution logicielle. Nous reviendrons plus tard sur ce point (§ 9.6). Pour l'éditeur, le développement de spécifique est un avantage concurrentiel. À partir de ce constat, suivant le taux de correspondance de la solution logicielle avec les attentes de l'entreprise, le client a déjà la possibilité d'éliminer des éditeurs trop éloignés de la solution attendue (Choix de l'ERP). Pour les autres éditeurs, il est possible de mettre en place le mécanisme d'extension afin d'effectuer l'analyse des développements spécifiques à réaliser. L'utilisation d'une base de connaissances par les chefs de projet permet de voir si ce type de développement spécifique n'a pas déjà été réalisé. Cette base peut également être utilisée comme outil d'aide dans l'analyse du spécifique (étude des coûts et des ressources que cela implique pour tenir des délais de mise en œuvre).

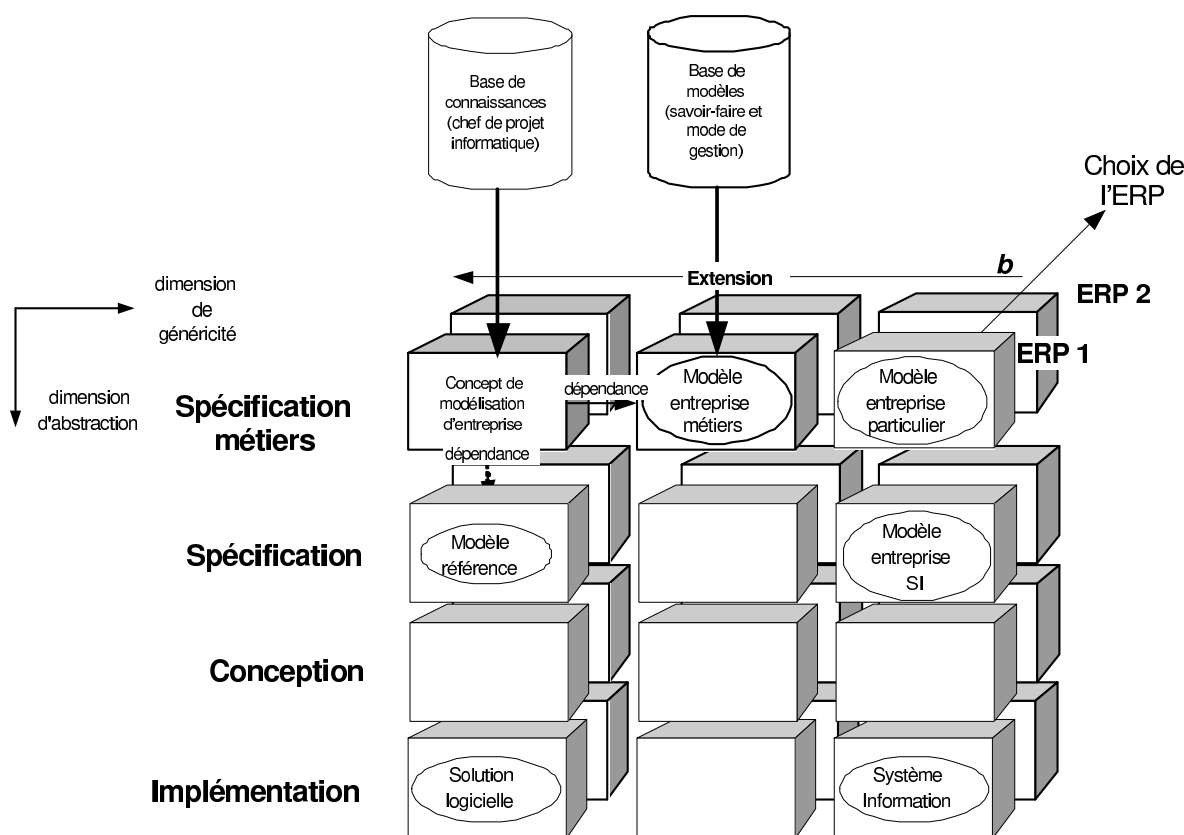


FIG. 9.13 – Conduite de projet « Deuxième phase »

Dans la deuxième phase de cette conduite de projet, nous avons fait le choix d'utiliser le mécanisme d'extension jusqu'au niveau générique de la dimension de généricité. Nous présenterons dans la section 9.6. le fait qu'il soit possible d'atteindre d'autres niveaux avec ce mécanisme.

Dans un second temps, le développement de spécifique doit servir à l'éditeur de la solution logicielle. En effet, l'éditeur doit remettre en cause son modèle métier et le modèle de référence de la solution logicielle. Si lors du déploiement de l'ERP, on s'aperçoit qu'il est nécessaire de réaliser un développement spécifique, cela veut dire qu'il existe un manque dans ces modèles. La fonctionnalité découverte a certainement une utilité pour cette catégorie d'entreprise. Il

s'agit d'une phase classique de retour d'expérience et de capitalisation de connaissances sur chaque projet ERP, même si l'éditeur n'est pas retenu par le client lors de l'étape de sélection.

9.4.2.2 Déploiement

L'outil est choisi, c'est donc le temps de la juste adéquation entre le logiciel et l'organisation. Nous pouvons donc réaliser une deuxième itération de notre conduite de projet dans l'objectif :

- d'analyser les besoins de manière plus détaillée que dans l'étape de sélection,
- d'exprimer ces besoins et de trouver les solutions dans le progiciel en proposant le bon paramétrage des fonctions standard,
- de multiplier les jeux de tests dans un souci de validation, avant la mise en production du Système d'Information,
- de combler les trous fonctionnels, sachant que le choix du client de développer ou non un spécifique, en connaissance de cause, va impacter le coût et le délai, mais également sur la maintenance en exploitation (ou évolutive - mise à jour du produit) de son système.

Quel niveau de granularité atteindre dans la spécification détaillée des besoins de l'entreprise ? Cette question est délicate et elle doit être discutée avec le client lors des différentes analyses. Un élément de réponse se situe dans le choix, par l'entreprise, d'utiliser des processus standard de la solution logicielle. Dans ce contexte, il n'y a pas d'intérêt à détailler ce processus, nous pouvons donc rester à un niveau macroscopique. Par contre, si, pour l'entreprise, le processus choisi est critique dans l'atteinte des performances, c'est-à-dire qu'il touche son cœur de métier, alors il est nécessaire de descendre à un niveau de détail supérieur sur tout ou partie de ce processus. Le modèle doit donc permettre de conjuguer plusieurs niveaux de granularité en fonction de l'importance relative des processus, au gré du client, en prenant en compte, même s'il est aidé par l'équipe projet, des difficultés probables d'appropriation de la modélisation.

Dans le même ordre d'idée, nous devons aussi nous intéresser au niveau de détail des objets de l'entreprise. Dans la précédente étape (sélection de l'ERP), nous parlions d'objets de l'entreprise comme des classes candidates représentant des objets réels tels qu'un bon de livraison, une commande, une fiche produit... Il est possible que le client souhaite adapter ces objets métiers à son organisation. Il est donc nécessaire de proposer au client de pouvoir spécifier, modifier ou paramétrer ces objets métiers. Mais comme pour l'analyse détaillée des processus, nous devons adapter le formalisme utilisé afin qu'il soit compréhensible.

9.5 Mise en œuvre des mécanismes et mise en correspondance des modèles

Afin de mieux appréhender la conduite de projet proposée dans la section précédente, nous allons détailler les différents mécanismes de notre cadre de référence.

Nous tenons à rappeler au lecteur que nous avons identifié trois possibilités, suivant le type de PME, pour la conduite de projet :

1. démarche pour une petite structure : dans ce cas, lors du déploiement de la solution logicielle, nous utilisons un pré-paramétrage correspondant au métier de l'entreprise. Les mécanismes de spécialisation et d'extension sont très limités, ceci est dû principalement aux nombres de journées d'analyse disponibles. Nous sommes dans une situation de configuration des principaux domaines et des processus associés ;

2. démarche pour une structure moyenne : dans ce cas, lors du déploiement, nous avons la possibilité de construire le modèle de l'entreprise. Nous utiliserons ce modèle pour le mettre en correspondance avec le modèle de référence de la solution logicielle. Nous appliquerons les différents mécanismes à ces modèles. Cette mise en correspondance nous permettra d'obtenir le Système d'Information en adéquation avec l'organisation ;
3. démarche pour une grande structure : dans ce cas, nous avons fait l'hypothèse que l'entreprise disposait déjà de son propre modèle d'entreprise avec son propre formalisme. Les mécanismes de spécialisation et d'extension ne s'appliquent pas. Par contre, une activité de conversion (mapping) est nécessaire entre le modèle de l'entreprise et le modèle de référence afin de les mettre en correspondance. Cette activité est nécessaire car les deux modèles ne sont pas basés sur les mêmes constructs de base et les langages de représentation sont peut être différents.

Dans le cadre de l'exemple présenté dans les prochaines sections, nous nous situerons dans le deuxième cas. Nous utiliserons un ensemble de modèles prédéfinis suivant les typologies d'entreprise (savoir faire, mode de gestion, ...).

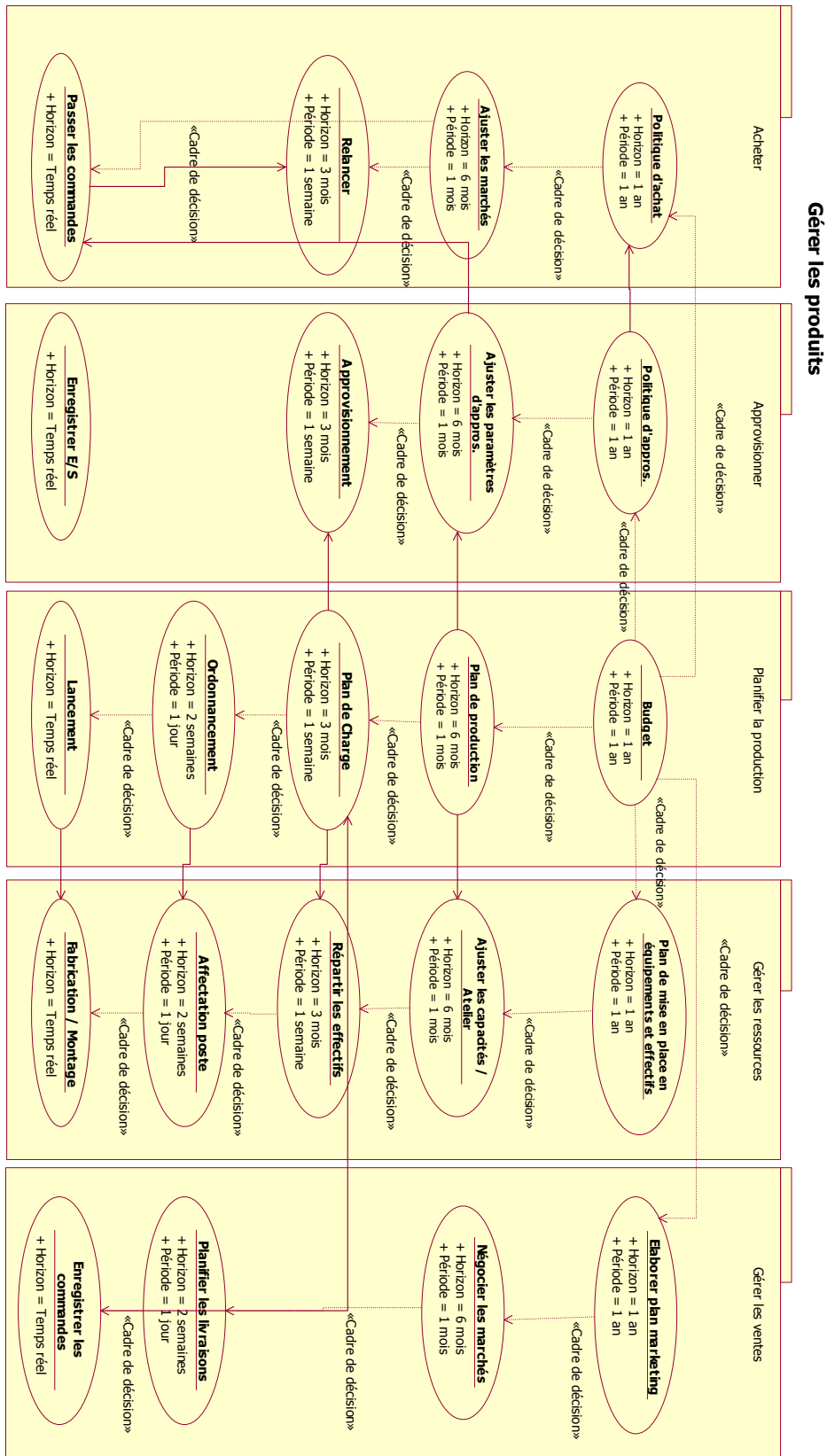


FIG. 9.14 – Grille GRAI à l'aide des cas d'utilisation

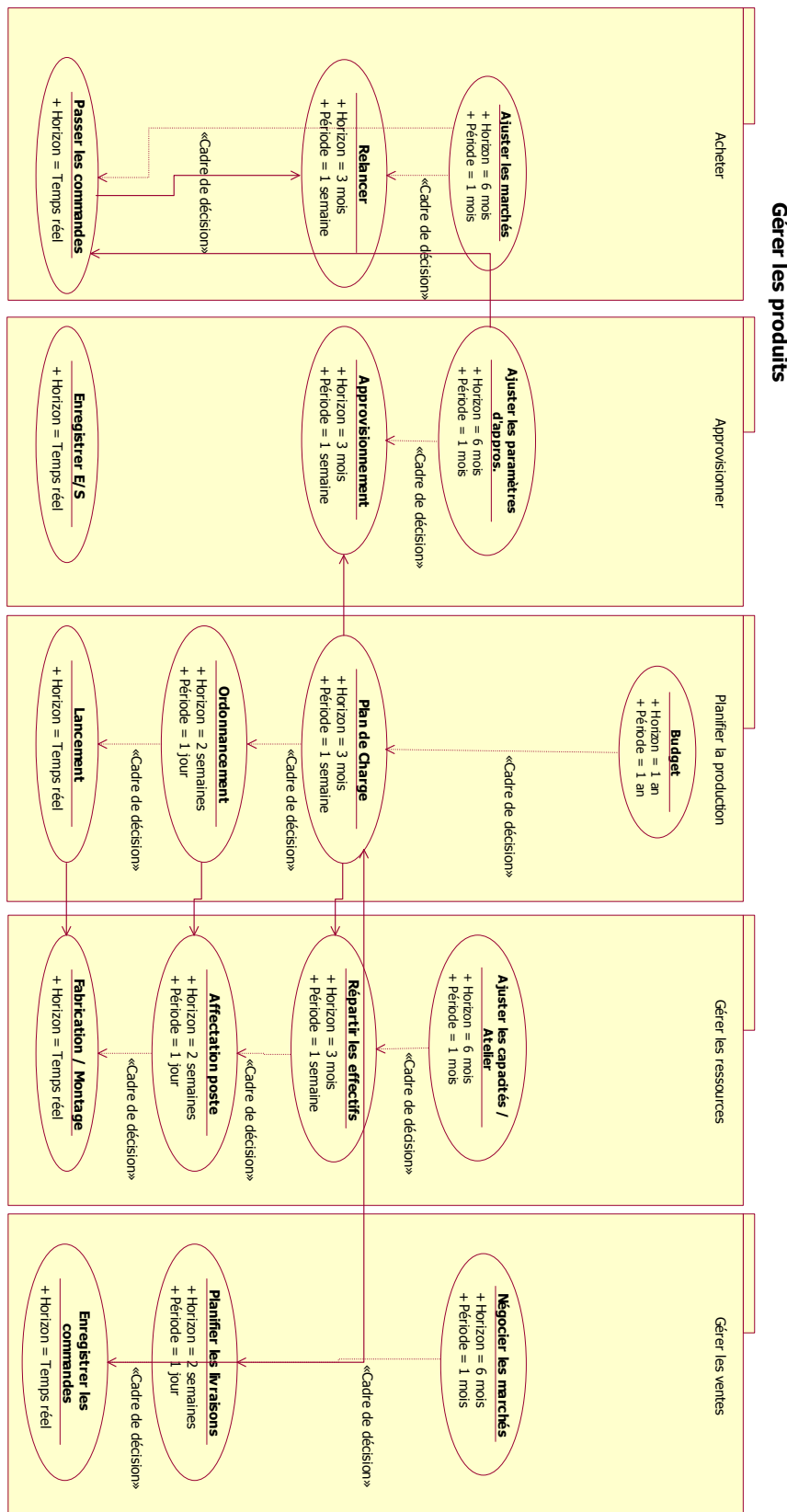


FIG. 9.15 – Grille GRAI particulière à l'aide des cas d'utilisation

Cet exemple rentre déjà dans la typologie, c'est un des points d'entrée du niveau partiel. Afin de lui appliquer le mécanisme de spécialisation, nous développons les étapes suivantes :

- réduction de la grille GRAI pour ne retenir que les centres de décision de l'entreprise,
- positionner ces centres de décision suivant les horizons et les périodes de l'entreprise,
- validation de la grille.

L'entreprise cible fonctionne à la commande. Par conséquent, l'entreprise n'a pas de « politique d'achat », de « politique d'approvisionnement » ou encore de « plan marketing ». De plus, il est rare de mettre en place des « plans de production ». Dès lors, nous obtenons une réduction de la grille GRAI (fig. 9.15) par suppression des cas d'utilisation et des liaisons dépendantes, et cette grille est propre à l'organisation de l'entreprise.

Cette grille particulière permet ensuite de travailler sur le réseau d'activités de chaque centre de décision. Pour cela, nous utilisons les diagrammes d'activités du langage UML. Comme auparavant, nous utilisons des diagrammes d'activités pré-configurés suivant les typologies d'entreprise. Les étapes suivantes du mécanisme de spécialisation sont :

- réduire le réseau d'activités des centres de décision de l'entreprise,
- valider les différents réseaux.

Là encore, procédons à l'aide d'un exemple, prenons le centre de décision « enregistrer commande » du domaine « gérer les ventes ».

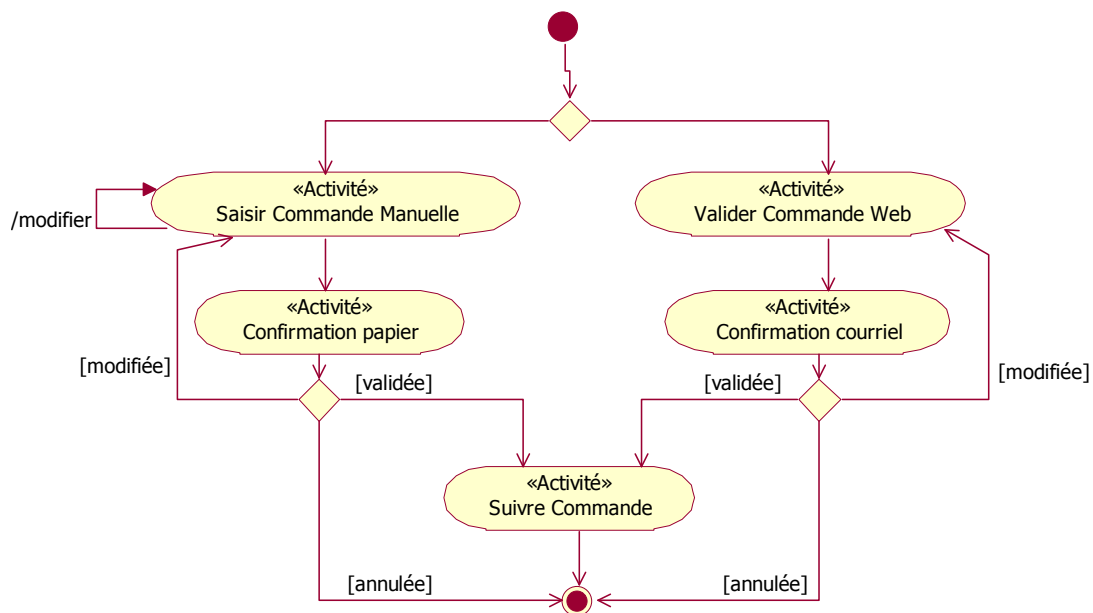


FIG. 9.16 – Réseau d'activité de « enregistrer commande » à l'aide du diagramme d'activité

Ce modèle permet de proposer à l'entreprise cible la possibilité d'enregistrer ces commandes de deux manières : soit par un enregistrement manuel, soit par la validation d'une commande via le Web. Comme pour la grille, nous allons spécialiser ce modèle pour le faire correspondre au vrai processus de l'entreprise. Elle n'a pas de vente par e-business, par conséquent, il est possible de supprimer toute la branche de droite « valider commande Web ».

Dans une troisième phase, et seulement lors de l'étape de déploiement de l'ERP (§ 9.4.2.2), nous pouvons instancier un certain nombre de modèles. Dans cette phase, il s'agit d'instancier les éléments classes des diagrammes UML pour les faire correspondre à l'organisation de l'entreprise, c'est-à-dire :

- identifier les ressources disponibles dans l'entreprise (applicatives, matérielles ou humaines) ;
- affecter les rôles (droits) des personnes (employés de l'entreprise et utilisateurs du Système d'Information) ;
- montrer les principes de codification de l'entreprise pour ces commandes, bon de livraisons, produits, ...
- etc.

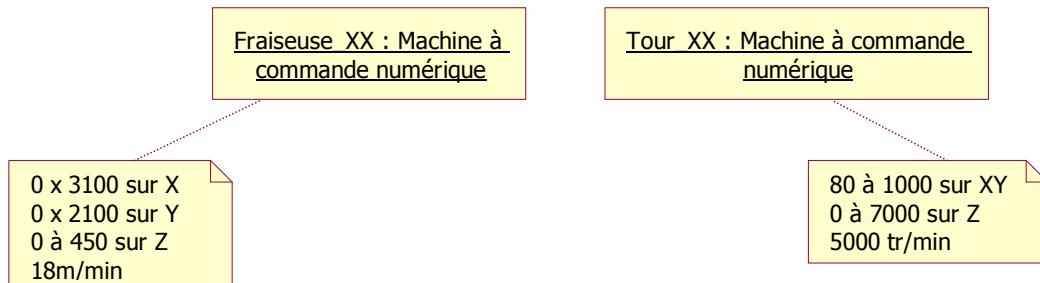


FIG. 9.17 – Liste des machines-outils et description de leurs capacités

À l'aide du diagramme objet, nous pouvons identifier la liste des ressources disponibles dans l'entreprise (fig. 9.17). Nous attachons des notes à ces objets pour exprimer leurs caractéristiques. La figure ci-dessus montre les capacités de deux des machines à commande numérique de l'entreprise.

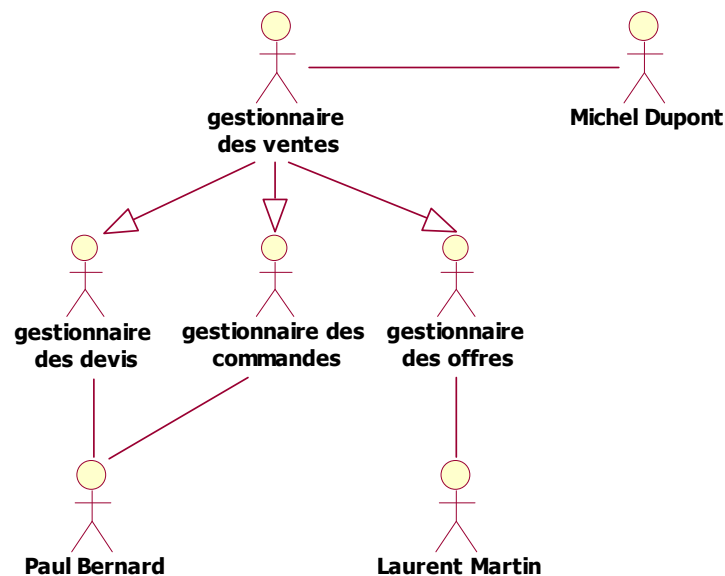


FIG. 9.18 – Affectation des rôles du système à des personnes physiques

Pour affecter les rôles (ou droits) à des personnes physiques de l'entreprise, nous avons plusieurs possibilités :

- à l'aide du diagramme de cas d'utilisation (fig. 9.18), nous pouvons mettre en relation un acteur du système (gestionnaire des ventes, ...) avec un acteur de l'entreprise, une personne physique (M. Paul Bernard)
- nous pouvons aussi utiliser le diagramme objet pour décrire la relation entre un rôle du système et un employé de l'entreprise.

Le diagramme de classe ci-dessus (fig. 9.19) exprime les relations entre les rôles et les personnes affectées à ces rôles. Cette relation permet d'exprimer les droits de chaque personne

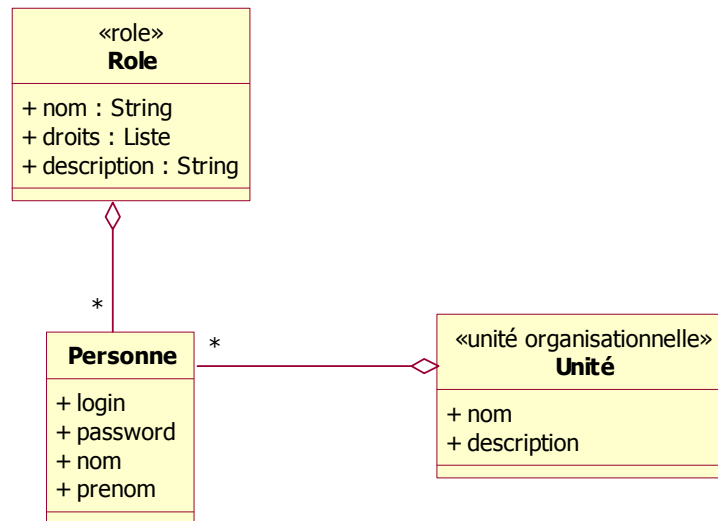


FIG. 9.19 – Relation entre les classes « Rôle », « Personne » et « Unité »

dans l'entreprise. Mais, il décrit aussi les relations entre les personnes et les unités. Cette relation exprime l'appartenance d'une personne à une unité organisation (service des achats, des ventes,...).

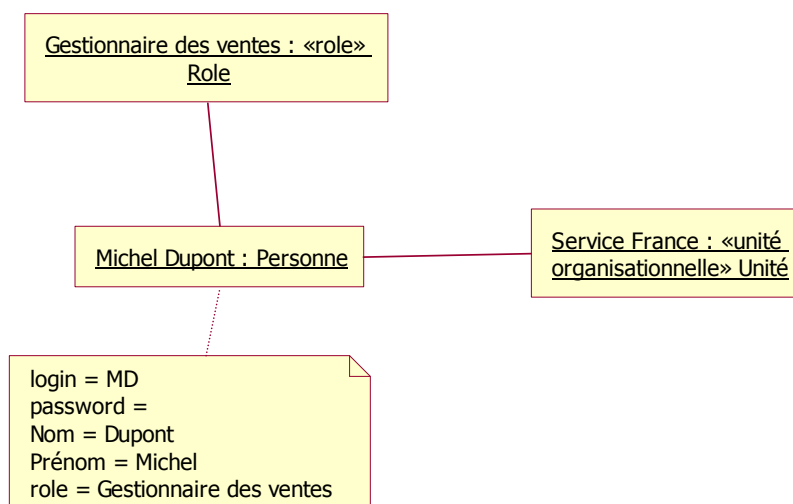


FIG. 9.20 – Instanciation d'un rôle, d'une personne et d'une unité

La figure ci-dessus (fig. 9.20) est une réalisation particulière du diagramme de classes de la figure 9.19. Il est alors nécessaire de reproduire ce modèle pour toutes les personnes de l'entreprise.

La figure ci-dessus (fig. 9.21) présente les codifications appliquées dans cette entreprise présentées sous forme de notes attachées aux classes. Pour les commandes, l'entreprise souhaite appliquer un système :

- à quatre chiffres étendus par un compteur pour la commande,
- aux trois premières lettres du nom du client + un compteur pour les clients.

Tous ces éléments instanciés seront utiles pour le mécanisme de paramétrage (*e*) de la solution logicielle. L'organisation, les employés, les rôles affectés aux employés font partie des exemples d'informations qui vont structurer son Système d'Information. Parfois, ces informations sont déjà archivées et leurs relations déclinées dans un outil en place, il s'agit alors de réussir à les récupérer sous forme automatique si possible.

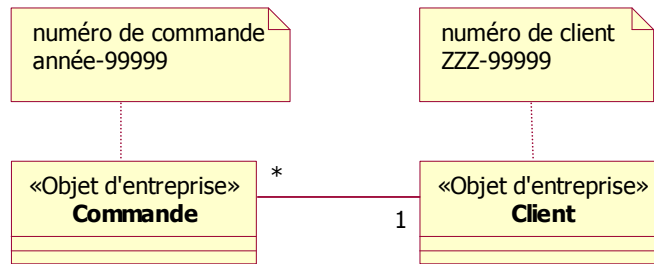


FIG. 9.21 – Description des codifications des objets d’entreprise

9.5.0.3 Mise en correspondance

Avant de parler de la mise en correspondance du modèle de l’entreprise du niveau des spécifications métiers avec le modèle de référence de la solution logicielle, il est nécessaire de rappeler quelques postulats :

- le modèle de l’entreprise et le modèle de référence ont été construits à partir des mêmes concepts de base, et ils sont calqués sur notre méta-modèle (§ 8.1.3) ;
- le modèle de référence de la solution logicielle couvre toute la typologie d’entreprise définie par le niveau partiel de la dimension de généralité.

Puisque la grille GRAI particulière définit le fonctionnement de l’entreprise à un niveau macroscopique, le périmètre fonctionnel de la solution logicielle peut en être déduit. Si des centres de décisions ont été supprimés, des fonctions ne sont pas requises, cela veut dire que des domaines ne seront pas déployés dans le Système d’Information. Nous pouvons par conséquent appliquer le mécanisme de spécialisation et réduire le diagramme de paquetage (fig. 9.22) des différents modules de la solution logicielle. Ce modèle sera utilisé par la suite pour le paramétrage du Système d’Information sur chaque domaine à déployer.

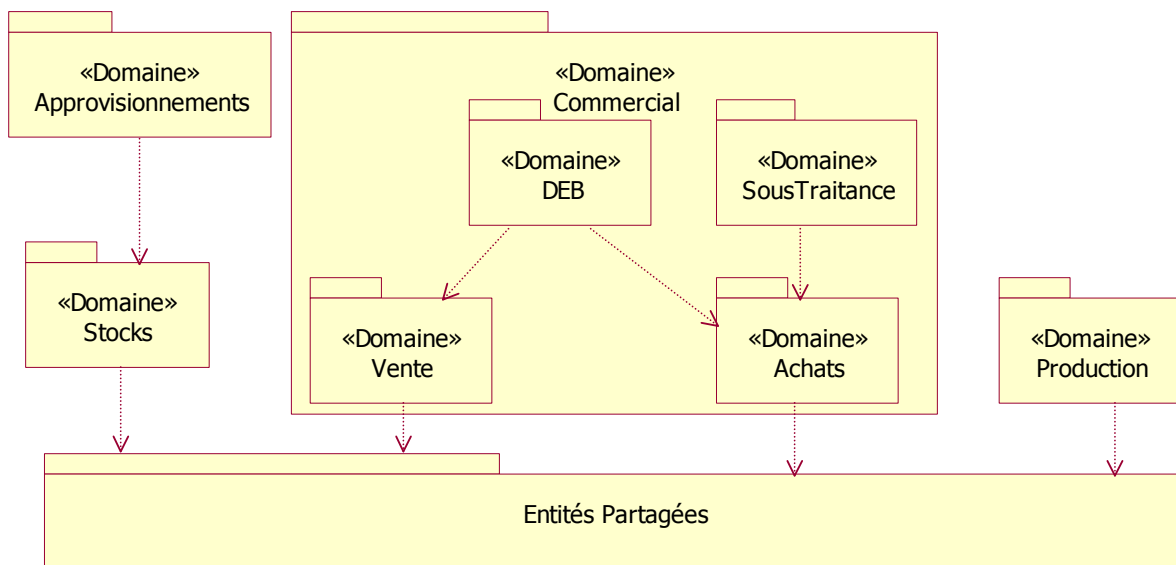


FIG. 9.22 – Domaine fonctionnel de la solution logicielle

Les relations entre les domaines représentent les dépendances entre ces domaines. Ces éléments permettent d’assurer la cohérence des modèles par rapport à des typologies d’entreprise et de détecter d’éventuelles incohérences dans les choix de modules. Par exemple, si une entreprise souhaite faire de la sous-traitance, le domaine « achat » est obligatoire pour gérer celle-ci.

Ensuite, toujours à partir de la grille GRAI, et à l'aide des diagrammes d'activités spécifiant les processus domaines, les processus métiers ou les activités, nous devons mettre en correspondance ces modèles avec les modèles de référence du niveau de spécification. Ces modèles de référence reprennent tous les processus, ou activités informatisables pour la solution logicielle. Nous faisons une distinction entre ces deux modèles, car il est possible que dans le modèle de l'entreprise, nous ayons identifié des activités non informatisables (manuelles par exemple, comme « emballer le produit »). Le Système d'Information ne pourra pas prendre en charge cette activité de cette nature, mais il aura à produire l'information pour gérer l'activité sous-jacente par des ordres, des comptes-rendus, ... Nous obtenons, en appliquant les simplifications du modèle d'entreprise, un modèle de référence particulier ou « modèle d'entreprise du SI ». Ce modèle de spécification est utilisé pour construire les modèles de conception. Dans les modèles de conception, nous retrouvons toutes les classes (processus domaine, processus métier, activités, information, ...) permettant le fonctionnement de la solution logicielle.

Dans l'exemple précédent (fig. 9.22), l'entreprise ne souhaitait pas gérer les commandes via le Web. Par conséquent, l'enchaînement d'activités est adapté, comme décrit sur la figure ci-dessous (fig. 9.23).

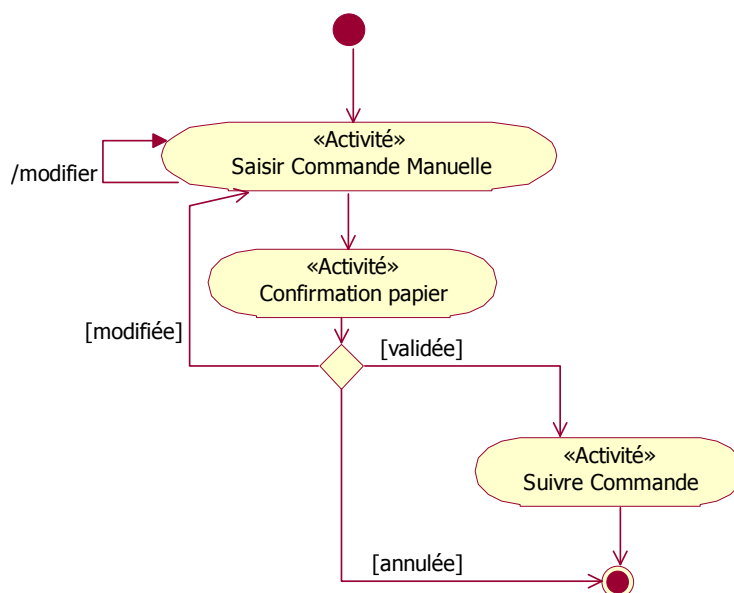


FIG. 9.23 – Enchaînement d'activités pour « enregistrer commande » (modèle d'entreprise)

Comme le modèle d'entreprise générique et le modèle de référence ont été construits sur la même base, le modèle de référence a, dans cet exemple, le même identifiant d'activité que le modèle d'entreprise générique.

Dans ce cas, il s'agit de mettre en correspondance, une par une, les activités des deux modèles. Le résultat de cette mise en correspondance est le modèle de référence particulier.

Enfin, il s'agit d'utiliser les modèles du niveau de spécification pour réduire les modèles du niveau de conception afin de ne garder que les éléments (classes) à déployer au sein de l'entreprise. Nous pouvons spécialiser le modèle de classe décrivant les relations entre les différentes activités du système. Dans notre cas, il se résumerait à la figure suivante (fig. 9.25).

La figure 9.25 présente le résultat, d'un point de vue de la conception, des relations entre le processus métier et les activités qui le compose. Les relations qui ne faisaient pas partie du diagramme d'activités ont été supprimées. Nous obtenons alors le modèle de référence particulier au niveau de la conception.

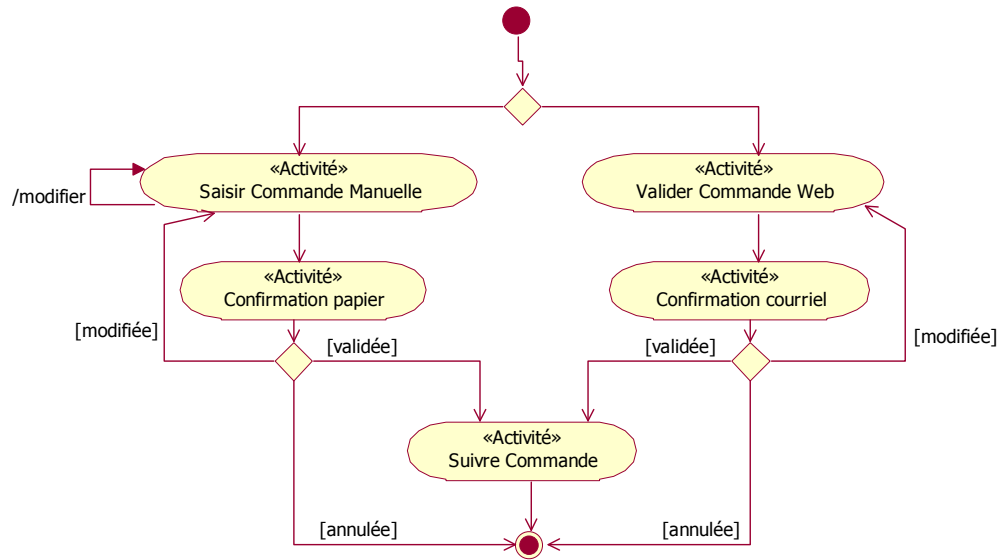


FIG. 9.24 – Enchaînement d'activités pour « enregistrer commande » (modèle de référence)

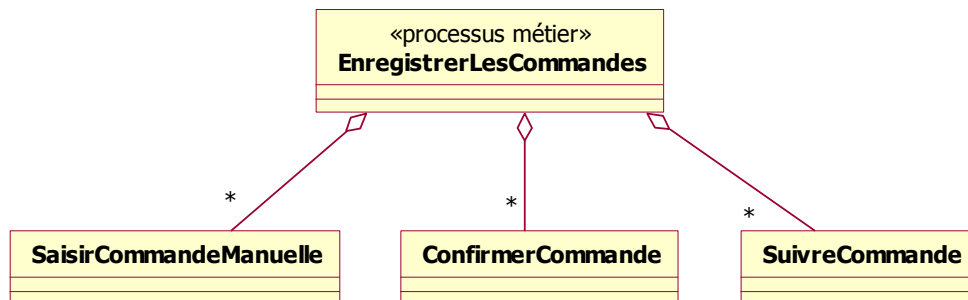


FIG. 9.25 – Enchaînement d'activités pour « enregistrer commande » (niveau conception)

La mise en correspondance du modèle d'entreprise et du modèle de référence, ainsi que l'évolution du modèle de référence suivant la dimension d'abstraction, est un travail long et méticuleux. Pour accélérer ce travail, nous avons étudié l'option d'utiliser un outil de configuration. Nous nous sommes rendus compte qu'il existait un parallèle fort entre la problématique de configuration de produit et la mise en correspondance des deux modèles (Franck Darras et al., 2003a). L'outil de configuration connaîtrait le modèle générique des solutions logicielles (modèle de référence). Le modèle d'entreprise servirait de guide à l'utilisateur lors des différents choix de configuration. Cette automatisation devra faire l'objet de poursuite des travaux de recherche.

9.5.1 Mécanisme d'extension

Le mécanisme d'extension est utilisé pour combler les trous fonctionnels. La découverte d'un manque se fait généralement lors de la réunion de brainstorming dédiée à la validation du modèle. Le développement de spécifique s'effectue en suivant le mécanisme (**d**).

Selon Ben Light (2001), il existe différentes façons de changer et de personnaliser son ERP. Cinq grands types sont identifiés :

- le changement d'une fonctionnalité : il s'agit du changement de l'implémentation de cette fonction. Ce genre de pratique est délicat pour la maintenance parce qu'il est important de maîtriser l'ensemble des connections avec cette nouvelle fonction. Dans le cadre de l'évolution du Système d'Information, deux possibilités sont offertes : soit l'introduire dans la future version, soit l'ignorer ;
- l'ajout d'une fonctionnalité : il s'agit de couvrir un trou fonctionnel détecté lors de l'analyse. Cette nouvelle fonctionnalité sera vraisemblablement ajoutée à la future version ;
- l'automatisation d'un processus ;
- le paramétrage d'interfaces homme-machine, ou de tableaux de bord ;
- les nouveaux tableaux de bords.

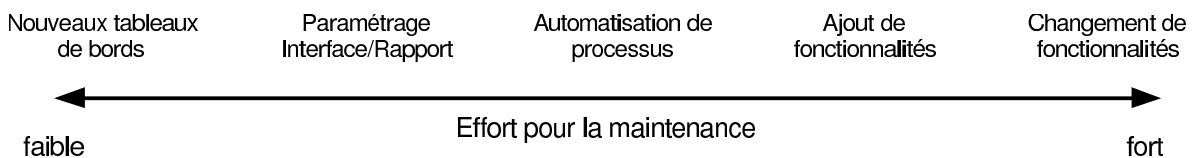


FIG. 9.26 – Effort de maintenance par rapport aux développements de spécificités

Au moyen d'un exemple, nous souhaitons montrer l'apport des concepts objets, et plus précisément du principe d'héritage, pour mettre en œuvre ce mécanisme d'extension sur le modèle de conception, et ainsi faire évoluer une application vers du spécifique sans trop d'effet de bord.

L'exemple concerne la gestion des remises. Il s'agit d'un changement de fonctionnalité. Dans l'application native, l'utilisateur a la possibilité d'appliquer deux remises à un montant. Dans l'entreprise cliente, les utilisateurs ont besoin de trois types de remises. Ce cas n'avait pas été prévu dans la solution logicielle. Il est donc nécessaire de faire évoluer le modèle, et de développer un spécifique pour modifier :

- l'interface homme-machine,
- modifier la règle de calcul processus métier.

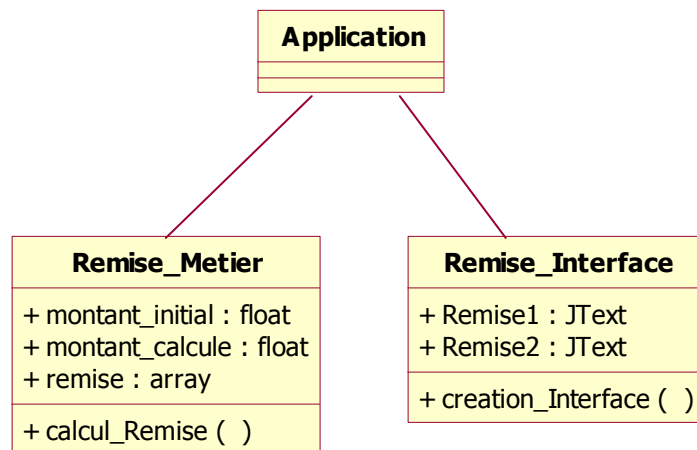


FIG. 9.27 – Modèle conceptuel de l'application (sans spécifique)

Le modèle de cette partie de l'application est présenté sur la figure 9.27.

En utilisant le principe d'héritage, nous pouvons répondre aux besoins de l'entreprise cliente et développer les spécifiques souhaités.

Tout d'abord, il est nécessaire de renommer les classes de l'application et de créer de nouvelles classes portant leurs noms. Avec le principe d'héritage, même si les nouvelles classes sont vides (sans attributs et sans opérations), le comportement de l'application n'est pas modifié. Les nouvelles classes héritent automatiquement des attributs et des opérations des classes supérieures.

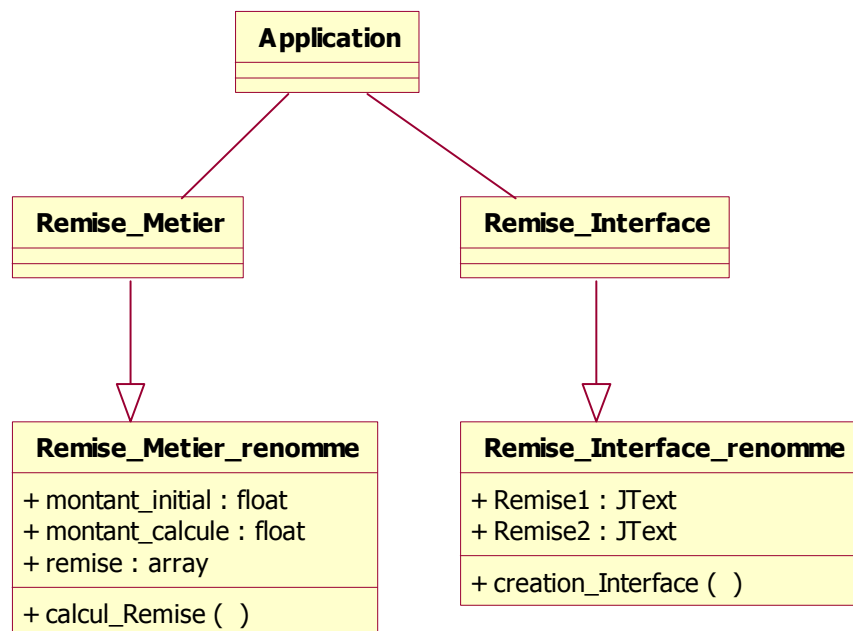


FIG. 9.28 – Évolution du modèle avec le même comportement (sans spécifique)

Puis, dans un deuxième temps, il est nécessaire de surcharger les attributs ou des opérations qui sont impliqués dans le spécifique. Dans notre cas, nous allons juste ajouter un attribut permettant l'insertion du nouveau champ (`Remise_interface`) et surcharger les opérations des

deux classes. Le développement de ces opérations est nécessaire, car la règle de calcul a été modifiée (prise en compte d'une troisième remise) et l'interface homme-machine a été modifiée (nouveau champ de saisie et nouveau mode d'affichage du résultat). Nous obtenons alors le modèle suivant (fig. 9.29) :

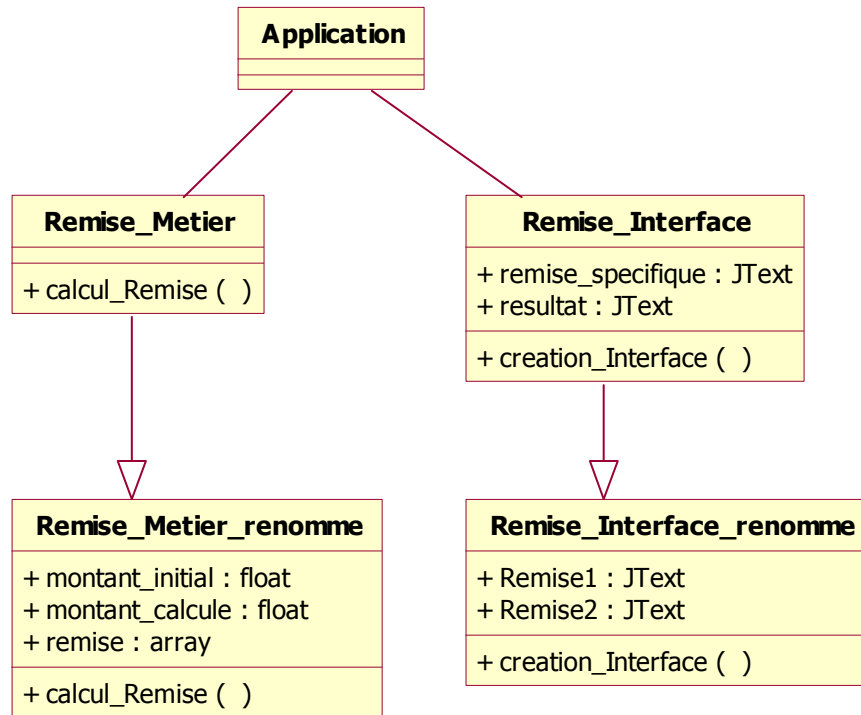


FIG. 9.29 – Évolution du modèle avec un nouveau comportement (avec spécifique)

Dans ce cas, l'application fait appel aux premières opérations des classes spécifiques et non des classes mères (en respectant les conventions de surcharge des attributs ou des opérations).

Le résultat final est présenté avec les deux écrans d'un prototype qui a permis le test de la fonctionnalité modifiée.

Ce principe d'héritage est donc appliqué aux niveaux de la conception et de l'implémentation du cadre de référence. Mais nous pourrions étendre ce principe au niveau des spécifications. Dans le diagramme de cas d'utilisation, nous avons la possibilité de faire hériter un cas d'utilisation d'un cas d'utilisation supérieur. Ce principe, en respectant des règles de modélisation et des conventions de nommage, pourrait être appliqué pour spécifier les processus spécifiques à une entreprise particulière.

9.5.2 Autres mécanismes

Dans le cadre de ce mémoire, nous ne détaillerons pas davantage les mécanismes de développement logiciel (*c*) et les mécanismes de paramétrage (*e*).

Pour le mécanisme de développement logiciel (*c*), nous avons présenté dans les sections précédentes (§ 6.2.1) les différents cycles de développement de projet informatique. Pour le mécanisme de paramétrage, il s'agit de renseigner des paramètres, d'affecter des valeurs liées à l'entreprise cible. Ces paramètres ont été identifiés lors de l'analyse, et définis lors de la particularisation du modèle de référence. Ces paramètres sont propres à l'éditeur, développeur

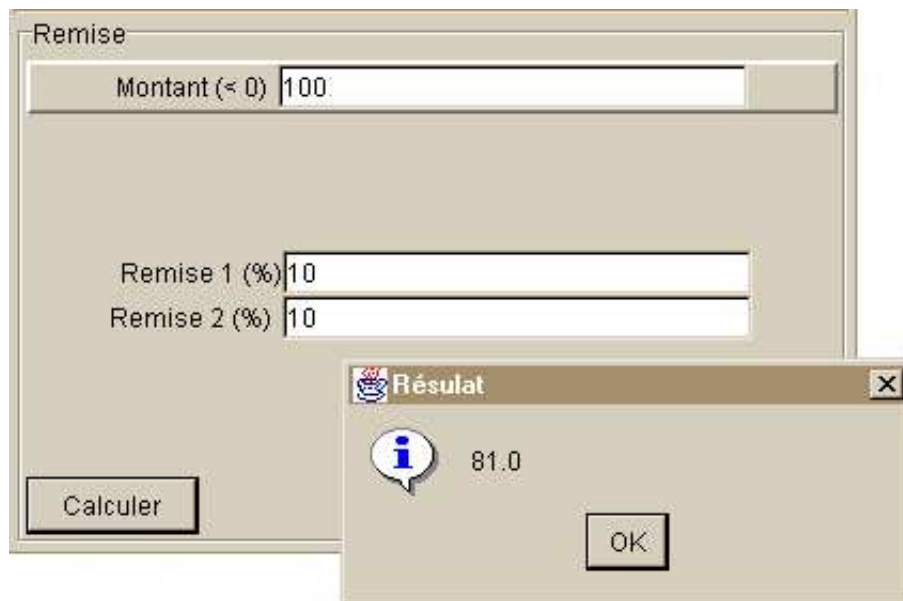


FIG. 9.30 – Exemple de gestion de spécifique (sans)

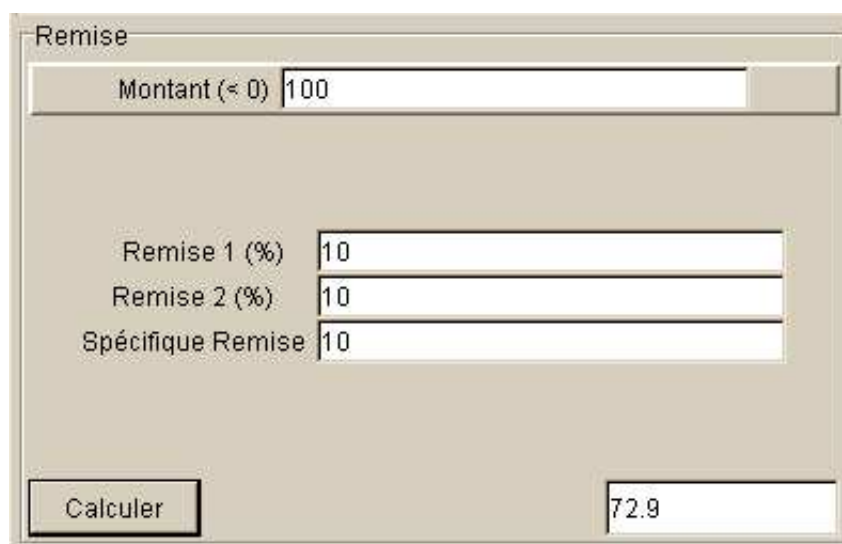


FIG. 9.31 – Exemple de gestion de spécifique (avec)

de solution logicielle qui a un savoir faire satisfaisant sur cette tâche. Ils n'ont pas fait l'objet d'étude dans notre travail de recherche.

9.6 La capitalisation

Il existe un double intérêt dans le développement spécifique. Compte tenu du nombre important de PME/PMI travaillant dans des domaines très variés, Il est très difficile de proposer une solution logicielle générique capable de couvrir plusieurs types d'activités (entreprise manufacturière, entreprise de service, ...).

Pour l'éditeur ou l'intégrateur, le développement de spécifique peut avoir deux facettes. D'un

côté, développer à la demande du client est un avantage concurrentiel par rapport à la concurrence, car très peu d'éditeurs de solution logicielle proposent ce service. De plus, le développement de spécifique permet d'enrichir l'offre pour les futurs clients ou de cibler vers un marché de niche. De l'autre côté, cela peut aussi s'avérer coûteux car s'il y a développement de spécifique, cela implique obligatoirement de maintenir autant de versions qu'il y a de Systèmes d'Information déployés distincts, et de porter une attention particulière lors de l'évolution de version pour la clientèle ayant fait ce choix.

Le développement de spécifique peut arriver à tout moment, il est donc nécessaire de prévoir dans l'architecture de l'ERP et dans le modèle, les mécanismes pour développer des spécifiques sans trop perturber la solution logicielle. La question de la capitalisation des connaissances consignées dans les modèles prend une certaine importance dans ce contexte.

À partir du cadre de référence présenté dans le chapitre précédent (§ 9.3.2), nous pouvons décrire le cheminement particulier dans un cadre de développement de spécifique, afin d'obtenir le Système d'Information adéquat répondant aux besoins du client. En suivant la dimension de généralité, nous pouvons définir trois cheminements particuliers pour la capitalisation :

- développement d'un spécifique pour un client,
- développement de spécifique pour un type d'entreprise (mécanique, électronique, bois...),
- développement de spécifique pour la solution générique.

Le développement d'un spécifique pour un client se situe seulement à un niveau de généralité particulier. Ces modifications ne seront pas reportées pour les autres clients qui ont déployé la solution logicielle. Par contre, il est important de souligner le suivi et la gestion de cette configuration dans un cadre de maintenance et d'évolution par un système futur.

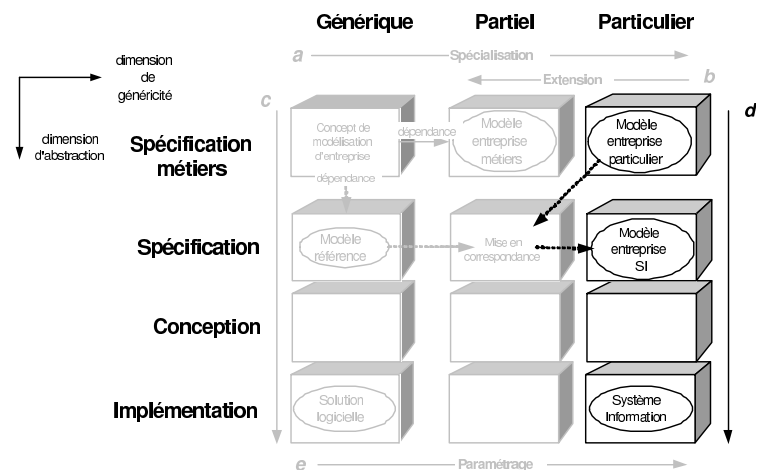


FIG. 9.32 – Cadre de référence - Extension particulière

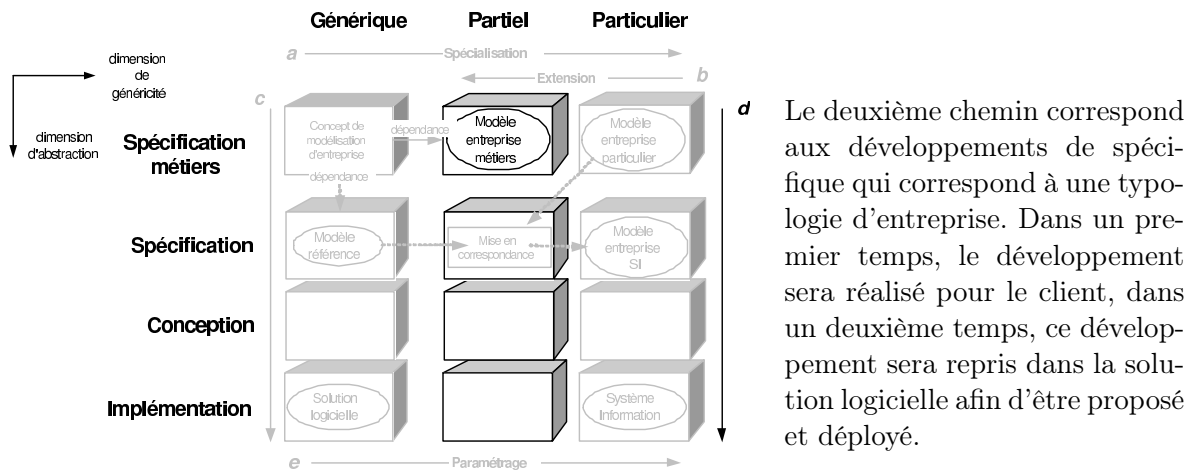


FIG. 9.33 – Cadre de référence - Extension partielle

Le troisième chemin correspond aux développements de spécifique qui peuvent être utilisés à un niveau générique et pour tous les clients potentiels. Dans un premier temps, le développement sera mis en place pour la solution du client, dans un deuxième temps, il sera repris et proposé à l'ensemble des prospects. Dans ce cas, il y a peu de problème lors de l'évolution de la solution, car ces fonctionnalités sont présentes pour toutes les solutions à déployer.

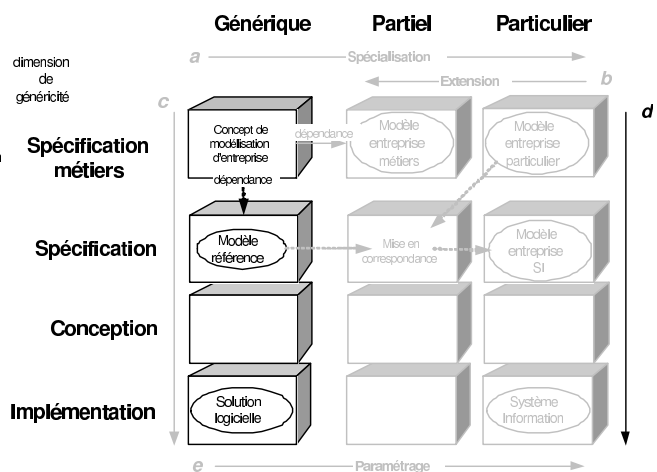


FIG. 9.34 – Cadre de référence - Extension générique

9.7 Conclusions

Dans ce chapitre, nous avons présenté notre cadre de référence pour la gestion de projets des ERP. Dans ce cadre de référence, nous pouvons positionner plusieurs formes de modèles participant au dialogue entre les parties, les modèles de l'entreprise et les modèles de référence, mais aussi la solution logicielle à un niveau générique, et le Système d'Information à un niveau particulier.

Nous avons vu dans les premiers chapitres de la thèse que le monde des PME est très varié par la taille, mais aussi le métier. Nous avons cherché à adapter notre cadre de référence par rapport à des structures de PME. L'utilisation du cadre est différente suivant les ressources disponibles lors du projet de déploiement. La construction des modèles particuliers à partir, soit de modèle partiel pré-paramétré respectant une typologie d'entreprise, soit à partir d'une bibliothèque d'éléments à assembler afin de construire le modèle, est une solution proposée.

À l'aide de notre cadre de référence et en appliquant différents mécanismes, nous avons présenté par l'exemple la mise en correspondance de la vision métier et de la vision informatique, le modèle d'entreprise projeté sur le modèle de référence. Cette projection doit participer à la recherche de grands équilibres, à la juste adéquation entre l'organisation et la solution logicielle déjà évoquée dans ce mémoire.

Dans le cadre de référence, le niveau de spécification métier a une utilité dans la conception et le déploiement d'un Système d'Information, pourquoi ne pas chercher à utiliser le même principe pour l'appliquer à des objectifs différents, tels que la réorganisation de l'entreprise, la mise en place d'indicateur de performance...

Conclusions générales et perspectives

Conclusions générales et perspectives

Dans le cadre de la collaboration avec un partenaire industriel, éditeur et intégrateur d'une solution logicielle dit « Enterprise Resource Planning » (ERP), la problématique traitée dans ce mémoire se trouve à l'intersection des disciplines du génie industriel et du génie logiciel. A partir de ces deux disciplines, notre travail de recherche a consisté en la proposition d'un cadre de référence pour la gestion des projets de conception (côté éditeur) et d'exploitation (côté client de l'éditeur) d'ERP dans des PME/PMI. Ce cadre de référence a été construit à partir de travaux académiques ou de travaux de normalisation en prenant en compte l'expérience acquise par le partenaire industriel.

Nous revenons, dans un premier temps, sur les résultats liés à notre thèse, tant sur les plans académiques qu'industriels. Dans un deuxième temps, nous proposons quelques pistes de réflexion sur les perspectives de poursuite des recherches que nous inspirent les résultats obtenus.

Bilans académiques et industriels

Dans la première partie de notre mémoire, nous nous sommes attachés à définir la place du Système d'Information dans les PME/PMI, et plus particulièrement les solutions logicielles de type ERP (Enterprise Resource Planning).

Dans un premier temps, nous avons situé, à partir d'une analyse bibliographique, la place du Système d'Information au sein des organisations. Nous avons souligné le fait que le déploiement d'un Système d'Information a aussi bien un impact sur l'entreprise et son organisation (maîtrise des flux financiers, informationnels, ressources, ...) que sur les utilisateurs du futur système (appréhension dans l'utilisation de l'outil). Par conséquent, la nécessité d'une juste adéquation entre l'organisation et la solution logicielle devenait l'un des postulats de base.

Dans un deuxième temps, nous nous sommes intéressés à l'évolution des outils de gestion de l'entreprise par rapport aux nouvelles méthodes de gestion. A partir d'un bref rappel historique, nous avons produit une synthèse sur ces méthodes en tissant un lien vers les grandes familles logicielles, telles que les CIM, GPAO, ERP ou encore SCP. Notre partenaire industriel développe et intègre un produit de type ERP. Nous nous sommes donc efforcés de replacer et commenter les différentes références bibliographiques définissant cette famille de produits logiciels. Cette étude, proche d'une étude de marché, fait le constat des tendances du secteur, aussi bien du point de vue des revenus des éditeurs et intégrateurs, que des nouvelles formes d'offre avec l'arrivée de solutions « open-source ».

Ensuite, nous avons étudié les mutations du paysage industriel des PME/PMI. Dans un premier temps, nous avons fait un point sur les changements organisationnels du monde des PME/PMI afin de résister à une pression économique de plus en plus forte. Ces évolutions,

inscrites dans un contexte de mondialisation, ne sont pas sans conséquence sur le devenir des Systèmes d'Information de ces entreprises. Ces différentes mutations vont avoir des impacts sur les architectures physiques et logiques des solutions logicielles, le mode de vente de la solution logicielle auprès du client, et enfin sur la relation client et éditeur/intégrateur dans la gestion de projet.

Dans un deuxième temps, nous nous sommes attachés, à partir des définitions de l'AFITEP (Association Francophone de Management de Projet), à poser les bases de la gestion de projet ERP. Que ce soit du point de vue de la maîtrise d'œuvre (formée de l'éditeur et l'intégrateur) ou du point de vue de la maîtrise d'ouvrage (formée du client et de conseils), nous avons présenté les rôles et les objectifs de chacun, ainsi que les tâches qui leur incombent. Nous avons utilisé cette décomposition en tâches dans notre cadre de référence.

À partir de cette base bibliographique, nous avons produit le cadre de référence pour la gestion de projet, de la conception au déploiement, basé sur les points forts suivants :

- la définition de l'architecture dans le développement d'un ERP,
- la place de l'ingénierie des Systèmes d'Information et l'importance d'un modèle de référence de la solution logicielle,
- le rôle de la modélisation d'entreprise pour une meilleure compréhension de l'organisation de l'entreprise,
- la mise en correspondance du modèle de l'entreprise et du modèle de référence de la solution pour la juste adéquation entre organisation et Système d'Information.

La définition de l'architecture des ERP devient de plus en plus complexe. Nous ne pouvons plus dire que cela est dû à l'émergence d'Internet et des nouvelles technologies de l'information et de la communication. Ces deux points sont devenus une réalité de la vie de tous les jours pour l'entreprise de type PME/PMI, et par conséquent un passage obligatoire. De nouvelles architectures physiques sont apparues car le mode client/serveur a des inconvénients pour les managers (coût de déploiement, coût de maintenance, ...). De plus, les dirigeants souhaitent une évolution permanente de leur Système d'Information par rapport aux stratégies d'entreprise et aux besoins des utilisateurs. Ceci est amplifié par le besoin d'interopérabilité entre des systèmes distants dans le cadre de réseaux d'entreprises. Par conséquent, il est nécessaire de proposer aux entreprises une solution intégrable, ouverte, adaptative, de mise à jour et de mise en œuvre aisées. Une telle architecture remettra en cause certains éléments de définition des ERP car nous aurons :

- une architecture distribuée de la solution logicielle sur le réseau Internet,
- une centralisation logique des données (mais peut être pas physique) et une saisie unique,
- une intégration de modules fonctionnels provenant de différents éditeurs.

Pour mieux maîtriser ces nouvelles architectures, il est nécessaire d'adapter l'ingénierie des Systèmes d'Information. Après avoir présenté brièvement les différents cycles de développement des systèmes, nous nous sommes attardés sur des cadres de modélisation de Systèmes d'Information tels que MERISE, SADT et le framework de Zachman. Nous avons vu que ces cadres de référence sont très riches, et ils proposent de multiples points de vues. Chaque méthode propose son propre formalisme. Mais l'usage grandissant des approches objets est accompagné d'une utilisation toujours plus répandue du langage UML. UML peut s'imposer comme une base intéressante dans la maîtrise de l'architecture des ERP, dans la maîtrise de la complexité et de la complétude des fonctionnalités proposées, mais aussi comme un moyen de communication entre le client utilisateur et l'éditeur/intégrateur.

D'un côté, nous devons prendre en compte le souhait de l'éditeur/intégrateur d'une maîtrise des connaissances sur les différents aspects de l'architecture de son système, de l'autre côté,

nous avons besoin de mieux appréhender et de formaliser l'organisation de l'entreprise. Dans ce contexte, la modélisation d'entreprise trouve toute sa place et se positionne comme un trait d'union.

La modélisation d'entreprise apporte, avec les normes, une vision structurante pour la construction des modèles. Comme pour la construction du modèle de référence de la solution logicielle, la modélisation d'entreprise propose nombre de méthodes et de formalismes apportant chacune une contribution intéressante (sur le point de vue fonctionnel, informationnel, décisionnel, physique, ...). Les entreprises, même les PME/PMI, ont l'obligation, soit à cause d'un partenariat avec un donneur d'ordre, soit avec la mise en œuvre d'une certification qualité, de structurer leur organisation autour de leur processus. Les méthodes et normes de la modélisation d'entreprise apportent deux éléments forts : un cadre conceptuel qui fournit une approche descriptive et discriminante des modèles de ce genre, et une syntaxe et sémantique par les langages de modélisation d'entreprise.

Dans le cadre de nos travaux de thèse, nous proposons de représenter les éléments structurants de la modélisation d'entreprise à l'aide du langage UML. Que ce soit suivant la dimension d'abstraction, ou la dimension de généricité, le langage UML apporte tous les éléments de représentation adaptés à chacun des modèles à l'intersection de ces deux dimensions. Pour cela, nous avons proposé notre méta-modèle. Ce méta-modèle est à la base de la construction des modèles particuliers d'une entreprise. Il fournit les « constructs » de base à la modélisation de notre entreprise cible. Le langage UML nous a fourni la syntaxe pour construire les modèles à partir des éléments graphiques. Afin d'apporter la sémantique à ces éléments, nous avons principalement utilisé la notion de stéréotypes. Au travers d'exemples didactiques, nous avons essayé d'apporter la preuve qu'il était possible d'utiliser le langage UML pour la représentation des différentes dimensions de la modélisation d'entreprise et principalement la dimension des points de vue. Ces travaux nous permettent de lancer une nouvelle piste de recherche sur la définition d'un profil UML pour la modélisation d'entreprise.

Enfin, nos travaux de thèse se sont finalisés par la définition d'un cadre de référence permettant de placer le modèle de référence de la solution logicielle et le modèle de l'entreprise dans le cadre de la gestion de projet. Afin de mettre en correspondance ces deux modèles, nous avons proposé des mécanismes adaptés dans notre cadre de référence. Ce cadre de référence est basé sur le cadre conceptuel de la norme de modélisation d'entreprise, et il a été adapté pour répondre aux besoins de notre partenaire industriel. De plus, nous avons construit ce cadre de référence en héritant des travaux de Butler, Boutin et Morley, sur l'évolution des modèles et leur mise en correspondance dans un projet de déploiement d'ERP.

À l'aide d'un exemple, nous avons défini le fonctionnement du mécanisme de spécialisation. Ce mécanisme permet d'obtenir le modèle particulier de l'entreprise. La construction de ce modèle est basée sur l'utilisation détournée d'une grille GRAI afin d'agencer les centres de décisions de l'entreprise comme des cas d'utilisation UML. Chaque centre de décision est relié à des processus à l'aide du diagramme d'activités. Puis, nous réalisons les modèles de conception pour le paramétrage de la solution logicielle. Le mécanisme de paramétrage est l'étape ultime qui permet de produire le logiciel à intégrer dans le Système d'Information. En utilisant un autre exemple, nous avons présenté le fonctionnement du mécanisme d'extension permettant le développement de spécifique afin de recouvrir les trous fonctionnels de la solution logicielle, et d'obtenir une meilleure adéquation organisation/logiciel.

Ce cadre de référence nécessite d'être expérimenté sur des projets de taille réelle pour assurer sa robustesse et sa complétude. Il a été transféré vers les services R&D de notre partenaire industriel pour la gestion des futurs projets de déploiement d'ERP. Des travaux complémentaires restent à mener sur ce cadre de référence, tels que l'automatisation de ces mécanismes.

Perspectives

Dans les perspectives que nous pouvons proposer à la suite de travaux de thèse, nous plaçons, dans tous les cas, le **modèle** au cœur du problème.

En premier lieu, il est nécessaire de travailler à court terme sur le cadre de référence que nous proposons. Ces travaux auraient pour objectif d'assurer la robustesse et la complétude de nos propositions et du positionnement des modèles et des systèmes. Des travaux dans ce sens sont actuellement menés par l'équipe R&D de notre partenaire industriel. Cette équipe travaille sur l'adaptation du méta-modèle, sa déclinaison en modèle métier suivant les savoir-faire et les modes de gestion des entreprises cibles de l'industriel. Enfin, ces travaux devront mesurer l'impact des modèles particuliers sur la solution logicielle et la mise à disposition d'instruments pour un bon paramétrage.

Dans un second temps, il est nécessaire de profiter de la synergie créée par les travaux sur UEML (Unified Enterprise Modeling Language). Pour cela, il est nécessaire que la communauté de la modélisation d'entreprise, les différents créateurs des méthodes de modélisation, la communauté de l'OMG, et pourquoi pas, les éditeurs d'outils de modélisation pour le déploiement d'ERP (Aris par exemple) se réunissent afin de définir le méta-modèle commun. Certaines critiques présentent déjà ce méta-modèle comme un « esperanto », mais si les éditeurs d'outils de modélisation d'entreprise veulent résister à l'évolution du marché, cela deviendra nécessaire. Comme pour le marché automobile et celui, plus proche, du génie logiciel et des langages de programmation, il y a un recentrage autour des leaders :

- pendant l'essor de l'industrie automobile, il y avait une centaine de constructeurs. Il n'en reste qu'une demie douzaine,
- les éditeurs peuvent utiliser une trentaine de langages de programmation, actuellement trois langages de programmation sont utilisés dans 80 % des projets.

Dans la communauté de la modélisation d'entreprise, la bonne parole des éditeurs d'outils d'aide au déploiement ERP commence à se faire entendre. Pour résister suffisamment longtemps, il est nécessaire de se regrouper autour de concepts communs (plate-forme identique pour l'automobile entre la C2 et la 1007, interopérabilité entre les langages de programmation) afin de parler des mêmes éléments de modélisation (construct) tout en gardant une spécificité de chaque méthode.

Ces travaux sur UEML pourraient permettre de développer, avec l'OMG, un profil UML. Ce profil pourrait être utilisé comme base de la modélisation, aussi bien du côté métier que du côté logiciel. Lors d'une table ronde aux journées de l'ADEPA, les industriels français nous ont confirmé le fait qu'il existait, malheureusement, une séparation entre la modélisation des besoins de l'utilisateur et la modélisation de la solution logicielle proposée. Nous pensons qu'en tendant vers un formalisme unique, et en utilisant les travaux sur UEML (passage entre différentes méthodes), il est possible de faire tomber cette barrière, et ainsi d'obtenir une meilleure communication entre le métier et la direction des Systèmes d'Information.

Enfin, il serait intéressant de travailler à plus long terme sur l'automatisation des mécanismes pour le passage d'un modèle d'entreprise à un modèle de référence. Dans un article présenté au congrès de génie industriel de 2003 (Franck Darras et al., 2003a), nous avons posé les bases de ce concept et réalisé une comparaison avec les outils de configuration de produits. Les recherches sur la configuration de produit étant un pôle d'excellence du laboratoire, la possibilité de mettre en œuvre un outil de configuration justifiait une occasion, parmi d'autres, de réunir nos compétences. Cet outil devra, dans un premier temps, permettre la mise en correspondance des deux modèles. Ceci est plus aisé si ces deux modèles sont construits à partir

d'un formalisme unique, tel qu'UML. Le langage UML propose un certain nombre de langages formels permettant d'extraire des informations précises. Nous pouvons citer XMI (XML Metadata Interchange) permettant d'échanger des modèles UML entre deux outils de modélisation ou encore OCL (Object Constraint Language) permettant de spécifier des contraintes sur les éléments graphiques du langage UML. Dans un deuxième temps, en utilisant les différents mécanismes de l'outil configuration, il devrait être possible de sélectionner les composants les plus appropriés, provenant éventuellement de différents fournisseurs, afin de construire l'application souhaitée. Cette spécificité permettrait de prendre en compte les nouvelles tendances architecturales tel que l'EAI, l'urbanisation ou les services Web. Ainsi, nous pourrions choisir le module gestion financière du fournisseur X, le module ressources humaines du fournisseur Y ou le module de gestion de production du fournisseur Z. Cette piste de recherche fera l'objet d'une nouvelle thèse au laboratoire de génie industriel à la rentrée 2004.

Une perspective à très long terme est dans la définition de la place et du rôle de MDA (Model-Driven Architecture) dans les futurs développements de solution logicielle. Le principe de MDA, décrit en annexe D, est dans la construction de modèle indépendant de la plate-forme de développement. Dans notre cadre de conception, lorsque nous naviguons sur la dimension d'abstraction, nous rajoutons de plus en plus d'informations en rapport à cette plate-forme. Un des principes de MDA est la personnalisation finale du modèle. Cette architecture permet de travailler sur le modèle, les spécifications et de définir au plus tard le choix de la plate-forme de développement. Elle permet aussi un changement de technologie moins coûteux. L'évolution de cette norme est actuellement supportée et suivie par l'OMG. De plus, Softeam, éditeur d'un outil de modélisation UML, tend à orienter les développements de son outil vers cette norme. Jean Bézivin et Xavier Blanc (2002) estiment que la norme peut devenir applicable à partir de 2005 pour une finalisation vers 2007.

Pour terminer, dans le cadre de la thèse, nous nous sommes attachés à effectuer une activité de veille sur les travaux des sociologues et leurs analyses des projets ERP. Avec la participation à des journées telles que TIC-TAC PME, nous sommes restés à l'écoute des critiques sur les défauts et les qualités des projets de mise en œuvre des Systèmes d'Information. Il serait intéressant de continuer ce travail de veille et d'analyser la place prise par le modèle dans les projets de déploiement, mais aussi sur le taux d'« adéquation » perçu par le client, entre son organisation et son Système d'Information.

Annexes

Annexe A

Grille de sélection

Objectifs

La grille de sélection, du CXP ou du CETIM, est considérée comme un outil d'expression du besoin.

La liste de questions doit permettre de :

- structurer, organiser et favoriser le dialogue dans le projet,
- révéler les besoins de l'entreprise,
- constituer un dossier de consultation de différents éditeurs portentiels.

La liste du CXP comprend pas moins de 400 questions. Le tableau A.1 page suivante, montre un extrait de ces questions.

N°	Questions stantard	Contraintes de votre entreprise	Pondération	Réponses
1	<p>Quel est le nom du progiciel étudié ?</p> <p>Quelle est la version étudiée ?</p>			
2	<p>Quel est le nom de la société distributrice ?</p> <p>Personne à contacter</p>			
3	<p>Mode de fonctionnemnet du progiciel :</p> <ul style="list-style-type: none"> - application centrale - client/serveur 			
4	Dans quelle langue le progiciel est-il disponible ?			
5	<p>Gestion et administration des ventes</p> <p>Le progiciel offre-t-il des fonctions dédiées à la gestion des aspects suivants :</p> <ul style="list-style-type: none"> - prospection ? - élaboration de devis ? - contrats, marchés, commandes ouverts ? 			
6	Gestion financière			
7	Gestion des ressources humaines			
8	Gestion de la logistique			
9	<p>Gestion industrielle</p> <p>Le progiciel offre-t-il des fonctions dédiées à la gestion des aspects suivants :</p> <ul style="list-style-type: none"> - production ? - maintenance ? - qualité ? - ordonnancement fin d'atelier ? - projet ? - SGDT ? - prix de revient ? 			

TAB. A.1 – Exemple de questions de la grille de sélection.

Annexe B

Unified Modeling Language

L'approche orientée objet

Historique

D'une approche fonctionnelle à l'objet

Dans l'analyse et la conception de systèmes complexes, l'approche fonctionnelle définit les principes suivants (Pierre Alain Muller, 1999) :

- diviser, décomposer pour comprendre,
- composer, réunir pour construire.

Dans l'approche fonctionnelle, on décrit le système par identification et décomposition en sous système correspondant à des fonctions plus ou moins élémentaires qui participent à la représentation de l'ensemble. Nous réalisons cette décomposition (fig. B.1) jusqu'à atteindre des éléments simples représentables dans les langages de programmation (fonctions ou procédure par exemple).

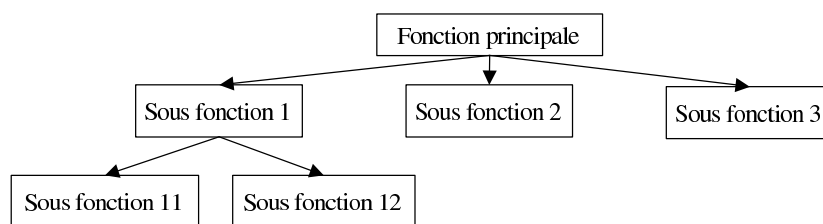


FIG. B.1 – Décomposition fonctionnelle et hiérarchique d'un système

La hiérarchie doit être stable, au sens où une évolution fonctionnelle ne doit pas provoquer des modifications structurelles lourdes.

Évolution des concepts objet

L'approche objet ne peut pas être considérée comme un concept nouveau, étant donné que les premières bases datent des années 65.

Le premier concept à apparaître est le mécanisme d'abstraction unifiant les notions de procédures et de données. Ce mécanisme a été élaboré par Ole Dahl et Kristen Nygaard, deux chercheurs norvégiens, en 1965 (Chabane Oussalah, 1997). Deux éléments sont apparus avec ces travaux, l'un correspond à l'encapsulation et l'autre correspond aux types abstraits de données. L'encapsulation est le mécanisme réunifiant des données et des procédures (traitements) dans la même classe. Quant à lui, le type abstrait de données est lié à la création de nouveaux types de données. Avant, nous avions des types simples (caractère, numérique), maintenant nous pouvons utiliser des types abstraits (chaînes de caractères, collection d'objets...).

Par la suite, ils ajoutent au mécanisme d'abstraction un mécanisme de raffinement de classe, le « préfixage », plus connu sous le nom d'héritage. Ce mécanisme permet, à partir d'une classe générale, de la raffiner pour obtenir une classe plus spécifique. Grâce à la mise en place d'une relation d'héritage entre les deux classes, nous pouvons utiliser à partir de la classe spécifique les propriétés et le comportement de la classe générale. Dans les années 70, les premiers langages, tels que Simula, ont mis en place ces premiers mécanismes.

Enfin, les derniers concepts de l'approche objet sont apparus grâce aux travaux de recherche sur Smalltalk-72 (Alan Kay et Adele Goldberg, 1976). Ce langage a permis d'implémenter des concepts tels que les associations entre classes, l'envoi de message et les précédents concepts implémentés par Simula.

Ces concepts ont permis à l'approche objet de se rapprocher de la modélisation d'éléments réels. Cela permet de vaincre la complexité grandissante des nouveaux systèmes applicatifs. Un système est donc un ensemble d'objets qui coopèrent. La maîtrise de cette complexité est facilitée par le principe d'abstraction d'un objet. Un avantage est apparu avec ces concepts : la modularité des objets. Si deux objets possèdent les mêmes interfaces, ils peuvent être interchangés sans problème.

Définition

Un ensemble de notions et de concepts est apparu autour de l'objet. Ces principaux concepts sont :

- la définition d'une classe est la description abstraite d'une famille d'objets ayant une même structure et un même comportement,
- l'instanciation est un modèle (ou formalisme) qui permet de créer des objets, nommés instances,
- l'héritage permet de résoudre le problème du partage efficace et de la factorisation des connaissances en généralisant ou spécialisant un ensemble d'objets,
- la composition est le lien entre deux objets, dont l'un rentre dans la structure de l'autre,
- le concept de message permet la communication entre les objets par envoi de messages,
- le polymorphisme est la possibilité, pour des objets distincts mais ayant le même protocole, d'activer des méthodes différentes.

L'objet est une unité atomique formée de l'union d'un état et d'un comportement, il est désigné par son identifiant. Ces notions permettent d'appréhender des objets matériels très simplement. Par contre, il est parfois plus difficile de capter les entités abstraites en termes d'objet.

$$\text{Objet} = \text{Identité} + \text{État} + \text{Comportement}$$

L'identité caractérise l'existence propre d'un objet. Elle permet de distinguer tout objet de façon non ambiguë, et cela indépendamment de son état.

L'état regroupe les valeurs instantanées de tous les attributs de l'objet. Il évolue au cours du temps, mais par contre certaines composantes de l'état peuvent être constantes. Toutefois, l'état d'un objet est variable et peut être vu comme la conséquence de ses comportements passés.

Le comportement regroupe toutes les compétences d'un objet et décrit les actions et les réactions de cet objet. Chaque atome de comportement est appelé « opération ». Les opérations d'un objet sont déclenchées à la suite d'une stimulation externe, représentée sous la forme d'un message envoyé vers un autre objet.

Nous ne détaillerons pas plus les concepts de l'approche objet. Nous renverrons le lecteur vers des ouvrages de référence, tel que celui de Bertrand Meyer (Bertrand Meyer, 1988) ou celui de Chabane Oussalah (Chabane Oussalah, 1997).

Niveau d'utilisation

Après une définition rapide des concepts liés à l'approche objet, nous souhaitons montrer les différents contextes d'utilisation des objets.

Quels objets utilise t-on ?

Pour construire une application orientée objet, nous utilisons une typologie d'objets spécifique. Chacun de ces objets apporte à l'application un certain nombre de gains. Nous distinguons les objets suivants :

- les objets de l'interface graphique,
- les objets métiers,
- les objets de l'application.

Les objets de l'interface graphique – IHM (Interface Homme-Machine) – permettent d'offrir aux utilisateurs un certain nombre d'objets (des boutons, des listes...). Les avantages d'une approche objet pour définir une IHM sont dans l'encapsulation et la réutilisabilité de l'élément graphique. Chaque objet graphique possède ses caractéristiques et son comportement, publics ou privés. En outre, il est possible de réutiliser les mêmes objets graphiques afin d'avoir un comportement uniforme de l'application.

Les objets métiers sont une représentation des objets de gestion de l'entreprise (une commande, un article par exemple). En utilisant les différentes associations entre classes, nous pouvons construire un système composé de plusieurs classes en étant proche de l'objet du monde réel.

Jean-Michel Tysebaert (2001) décrit l'ensemble d'objet métiers comme :

- les objets métiers conceptuels : il s'agit des objets du monde managérial. Ils décrivent les informations, les connaissances et des activités que l'organisation a choisi de gérer, voire de capitaliser [...];
- les objets métiers organisationnels : il s'agit d'objets, issus des perceptions de l'utilisateur, analysés et modélisés par un concepteur en vue de la conception d'un système d'information.

Les objets de l'application sont simplement les objets utilisés pour le bon fonctionnement de l'application. Ils ne sont ni liés à la génération de l'interface, ni à la représentation des objets métiers. Il s'agit d'objets de conception ou de programmation.

Évolution des bases de données

Le modèle relationnel a été introduit par E. F. Codd (1969) dans les années 1970. Les premiers objectifs du modèle ont été formulés par E. F. Codd comme suit :

- permettre un haut degré d'indépendance des programmes d'applications et des activités interactives à la représentation interne des données,
- fournir une base solide pour traiter les problèmes de cohérence et de redondance des données,
- permettre le développement de langages de manipulation de données non procéduraux basés sur des théories solides.

Finalement, il a atteint deux autres objectifs non prévus à l'origine :

- être un modèle extensible permettant de modéliser et de manipuler simplement des données tabulaires,
- devenir un standard pour la description et la manipulation des bases de données.

Ces objectifs ont été réalisés grâce à IBM : le modèle entité-relation et le langage SQL ont été normalisés au niveau international en 1986.

Avec l'avènement de l'approche objet et des langages de programmation, plusieurs éditeurs ont tenté de développer des bases de données objets. Actuellement, la gestion de la persistance des objets dans une base de données relationnelle se fait par « mapping ». Le « mapping » est la procédure permettant de réaliser une carte de correspondance représentant les liens entre les attributs d'une classe et les attributs de la table où sont stockés les informations. Cette procédure est généralement automatisée.

Pour mériter le nom de SGBD objet, un système doit d'abord supporter les fonctionnalités d'un SGBD. Il doit assurer :

- la persistance des objets,
- la concurrence d'accès : la base objet doit pouvoir être partagée simultanément par les transactions qui la consultent et la modifient,
- la fiabilité des objets : les objets doivent être restaurés en cas de panne d'un programme dans l'état où ils étaient avant la panne,
- la facilité d'interrogation : il doit être possible de retrouver un objet à partir de la valeur de ses propriétés.

Il doit proposer des fonctionnalités optionnelles :

- la distribution des objets : cette facilité permet de gérer des objets sur différents sites, en particulier sur un serveur ou des clients,
- les modèles de transactions évoluées : il s'agit de supporter des transactions imbriquées, c'est-à-dire elles-mêmes décomposées en sous transactions qui peuvent être totalement reprises.
- les versions d'objets : la gestion des versions permet de revenir à un état antérieur de l'objet avant modification.

Il doit proposer les fonctionnalités inspirées d'une approche orientée objet :

- le support d'objets atomiques et complexes : il s'agit de supporter des objets avec des attributs de références et des collections imbriquées,
- l'identité d'objet,
- l'héritage simple,
- le polymorphisme.

Actuellement, il existe très peu de bases de données objet. De plus, il n'existe qu'une base de données objet commercial. Ce produit est édité par ObjectStore¹.

¹<http://www.objectstore.net/>

Bibliothèques d'objets

Pour terminer, nous souhaitons aborder les bibliothèques d'objets disponibles sur le marché. Il existe différents types de bibliothèques :

- librairie liée au langage de développement,
- composant fournissant un service spécifique.

Premièrement, chaque langage de programmation propose aux développeurs une bibliothèque préétablie. Cette librairie fournit des objets de base permettant de gérer des opérations élémentaires.

Deuxièmement, certaines sociétés informatiques proposent des bibliothèques de composants sur une problématique bien identifiée. Malheureusement, dans la plupart des cas, les composants sont considérés comme des boîtes noires. C'est-à-dire que le composant nous rend bien le service qu'il doit faire, mais on ne connaît pas son fonctionnement interne, la manière dont le composant a été implémenté ou encore son algorithme de fonctionnement. Nous pouvons citer la société ILOG comme fournisseur de composants. Elle propose un ensemble de composants basés sur des problématiques d'optimisation.

Une des perspectives possibles est la mise à disposition par des éditeurs d'ERP de composants résolvant des problématiques de gestion. Nous pouvons donner comme exemple la réalisation d'un calcul des besoins. Actuellement, il n'existe pas de tels produits sur le marché.

Apports de l'approche objet

L'utilisation d'une approche objet est indéniable pour développer de nouveaux projets. Cette approche se justifie par :

- une meilleure réutilisation des objets,
- un moindre coût pour la maintenance ou les évolutions du système.

Par contre, l'approche objet a aussi quelques inconvénients, tels que :

- la reconnaissance et la définition des classes d'objets doit être fondée sur une connaissance affinée des métiers de l'entreprise. Le choix de la décomposition d'un objet métier en un objet logiciel (ensemble de classes) est soumis à des choix d'abstraction très difficiles à maîtriser ;
- le développement d'une application orientée objet n'a d'intérêt que si elle a été conçue avec des objets réutilisables. Il n'y a aucun problème à développer une application procédurale avec un langage objet.

Principe de décomposition

Afin de mieux appréhender la place du modèle, il est nécessaire d'introduire la notion d'abstraction (Andrey Naumenko et Alain Wegmann, 2003), (Jean Bézivin, 2001). D'après l'OMG (Open Management Group), il existe quatre niveaux de modélisation :

0. l'univers du discours,
1. le modèle,
2. le méta-modèle,
3. le méta-méta modèle.

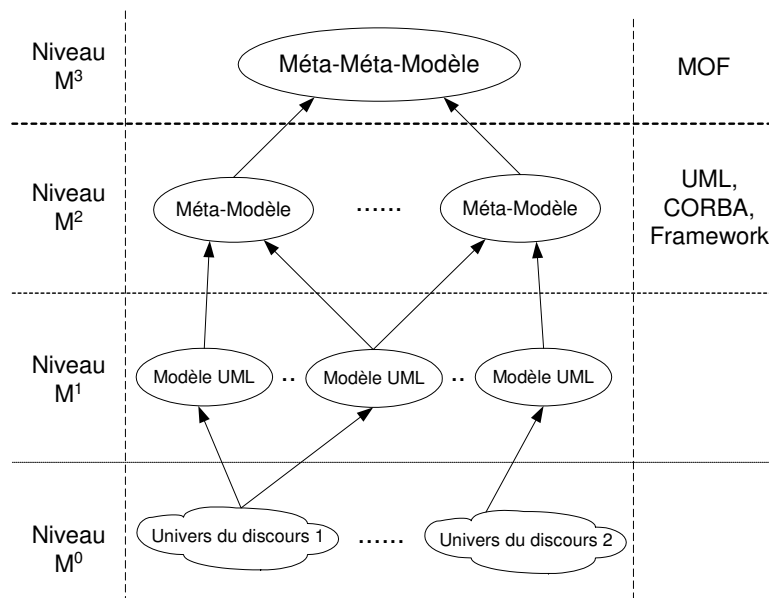


FIG. B.2 – Quatre niveaux de modélisation avec les standards de l'OMG

Chaque niveau (fig. B.2) se traduit par l'emploi de formalismes tels que UML (Unified Modeling Language) ou encore MOF (Meta-Object Facility). Entre chaque niveau, suivant une approche descendante, nous utilisons un principe d'instanciation pour passer de M_3 à M_2 , de M_2 à M_1 , etc. Seul le niveau M_3 n'est pas instancié car il est réflexif, c'est-à-dire que la description de ce méta-méta modèle s'applique à elle-même. Il s'agit donc du dernier niveau d'abstraction.

Entre chaque niveau, suivant une approche ascendante, nous utilisons un principe d'abstraction pour passer de M_0 à M_1 , de M_1 à M_2 , etc. Des auteurs, tels que Jean-Pierre Briffault (2000), Chabane Oussalah (1997) ou encore Pascal Roques et Franck Vallée (2003), définissent un modèle comme une abstraction du monde réel (l'univers du discours). Ce modèle permet d'avoir une vision simplifiée du système en termes compréhensibles afin de simuler, valider le résultat.

De plus Jean-Pierre Briffault (2000) souligne dans sa définition du modèle qu'il est possible d'avoir « plusieurs représentations de la réalité suivant le point de vue de l'observateur et de ses objectifs ». Cette vision est très intéressante dans le cadre de nos travaux, car le modèle n'est peut-être pas le même suivant le point de vue du client que ou de l'éditeur/intégrateur. Nous apportons de plus amples informations, sur ces deux points de vue, dans la dernière partie de ce mémoire.

Le méta-modèle

Le terme *méta* vient du grec et exprime « la succession, la transformation, le changement ». Dans les néologismes scientifiques (métalangue, métamathématique) méta signifie « ce qui dépasse, englobe un objet de pensée, une science » (Dictionnaire Robert).

Un méta-modèle est la définition d'un formalisme permettant par instanciation de créer le modèle. Chabane Oussalah définit le méta-modèle comme suit :

« Un méta-modèle est un langage formel ou semi-formel permettant de modéliser des systèmes particuliers, à savoir des modèles. » (Chabane Oussalah, 1997)

Le méta-modèle doit aussi avoir comme caractéristique le principe de généralité. C'est-à-dire qu'un méta-modèle doit fournir tous les éléments pour représenter un ensemble de modèles distincts.

Le méta-méta modèle

Le méta-méta modèle définit un formalisme instanciable pour obtenir le méta-modèle. Ce formalisme est le MOF (Meta-Object Facility) (Erwan Breton et Jean Bézivin, 2001).

Le MOF constitue donc la fondation de l'architecture de modélisation de l'OMG. Sur cette base ont été développés des méta-modèles comme UML (même si chronologiquement UML précède le MOF).

Approche orientée objet et « pattern de conception » dans les modèles

La spécification des nouveaux systèmes d'informations utilise des modèles mettant en œuvre une approche orientée objet, ainsi l'utilisation d'un langage de modélisation tel que Unified Modeling Language (annexe B) est inéluctable.

Afin d'améliorer l'utilisation des modèles, nous pouvons mettre en place des « patterns de conception ». Nous appelons pattern de conception, « *design pattern* », un ensemble de classes permettant de résoudre un problème particulier. On peut caractériser les patterns selon les critères suivants :

- le problème est classique et connu,
- l'ensemble de classes (ou « design pattern ») résolvant le problème porte un nom unique,
- la solution proposée correspond à la meilleure façon connue de répondre au problème.

On considère que la solution proposée, solution de référence, est la meilleure façon de résoudre le problème, parce que ce problème s'est présenté dans différents domaines. Les experts capitalisent leurs savoirs dans des patterns afin d'en faire bénéficier d'autres développeurs. Donc une généralisation de cette solution a pu être effectuée afin d'obtenir ce pattern.

« Un pattern est une idée qui a été utilisée dans un contexte pratique et qui sera probablement utilisée par d'autres » de Martin Fowler, extrait de (Martin Fowler, 1997).

Les patterns de conception permettent donc :

- d'offrir un catalogue de solutions « clé en main » aux développeurs,
- d'identifier par un nom ces solutions afin que les développeurs puissent bien parler de la même chose,
- de mettre en place les mécanismes nécessaires pour profiter au maximum des concepts de base d'une approche orientée objet.

Les patterns de conception fournissent un ensemble de classes abstraites mettant en œuvre les concepts de l'approche orientée objet. Ils ne sont pas dédiés à un domaine particulier. Dans le cas du modèle suivant (fig. B.3), nous montrons que nous pouvons adapter à plusieurs situations le même modèle, celui-ci ayant fait ses preuves.

Cet exemple (fig. B.3) montre le passage d'un méta-modèle à un modèle concret. Nous avons utilisé ce méta-modèle dans deux situations :

1. le premier cas correspond à une classe « commande » de type collection possédant plusieurs lignes de commande (de 0 à n) de type « membre ».

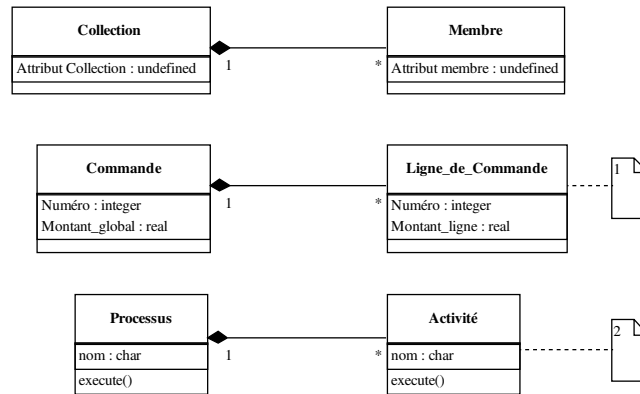


FIG. B.3 – Pattern de conception d’une vision abstraite à un cas concret

- le deuxième cas correspond à une classe « processus » de type collection possédant plusieurs activités (de 0 à n) de type « membre ».

Nous appliquons ce pattern lorsque nous détectons des relations et des spécifications particulières entre la classe dite « collection » et la classe dite « membre ». Le lien de composition entre deux classes implique une relation entre le cycle de vie des deux classes. La classe « membre » ne peut pas exister s’il n’y a pas de classe « collection ». Ceci est vérifié avec nos deux exemples, une ligne de commande n’a pas lieu d’exister si elle n’est pas rattachée à une commande.

L’exemple précédent (fig. B.3) était simple, mais le GOF (Gang of Four) Erich Gamma et al. (1999) ont proposé un ensemble de patterns (modèle de classe), plus complexe, permettant de maîtriser, sans controverse, l’assemblage de ces différentes classes afin d’appliquer la meilleure pratique pour un problème identifié.

Dans l’exemple suivant (fig. B.4), il s’agit de proposer un pattern de conception pour gérer l’assemblage de « composants » dans des « composites ».

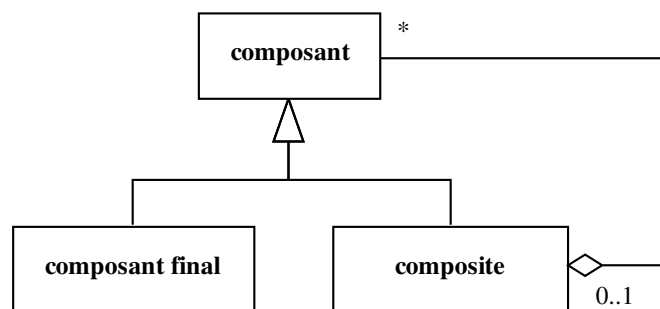


FIG. B.4 – Pattern de conception « composite »

Ce pattern permet de gérer facilement l’ajout d’un nouveau composant sans modifier toute la structure statique du modèle. Dans l’exemple suivant (fig. B.5), il s’agit d’utiliser ce pattern pour construire un produit. D’un côté, nous avons le produit final (le « stylo »), de l’autre, nous avons les composants de ce produit (la « bille » ou « l’encre »). Ce pattern peut aussi bien

être utilisé dans le cadre d'assemblage de produit, mais il peut aussi permettre de décrire une gamme de fabrication (une gamme est composée d'opérations, chaque opération est composée de sous-opérations...).

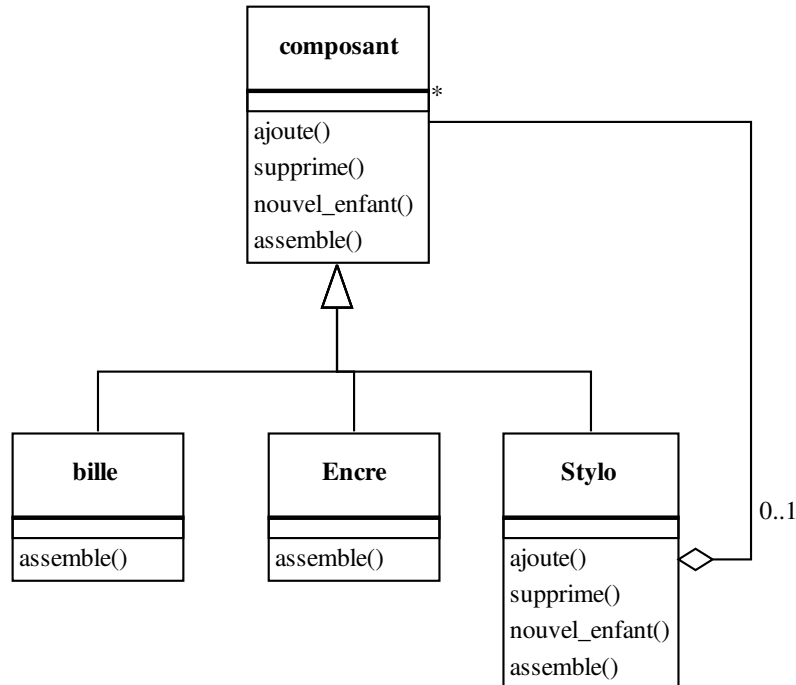


FIG. B.5 – Pattern de conception « composite » dans l'assemblage de produits

Il existe également des patterns métiers, dits « business patterns » qui remplissent le même office, d'un point de vue métier. On considère que les patterns métiers sont spécialisés dans un domaine particulier (réservation dans le domaine du transport), mais assez génériques pour être utilisés dans plusieurs métiers (aéronautique, ferroviaire, transport en commun).

Il existe de plus en plus de patterns d'analyse, de conception ou métiers, nous renvoyons le lecteur vers des ouvrages de référence tels que celui de Erich Gamma et al. (1999) proposant des patterns génériques, ou encore celui de Chris Marshall (1999) proposant des patterns liés aux objets de l'entreprise.

UML

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire les besoins, spécifier et documenter des systèmes et esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue (Pascal Roques et Franck Vallée, 2003).

Évolution d'UML

Historique

UML est principalement une fusion des notations Booch, OMT, OOSE... et des concepts orientés objet. UML réunit plusieurs « écoles » :

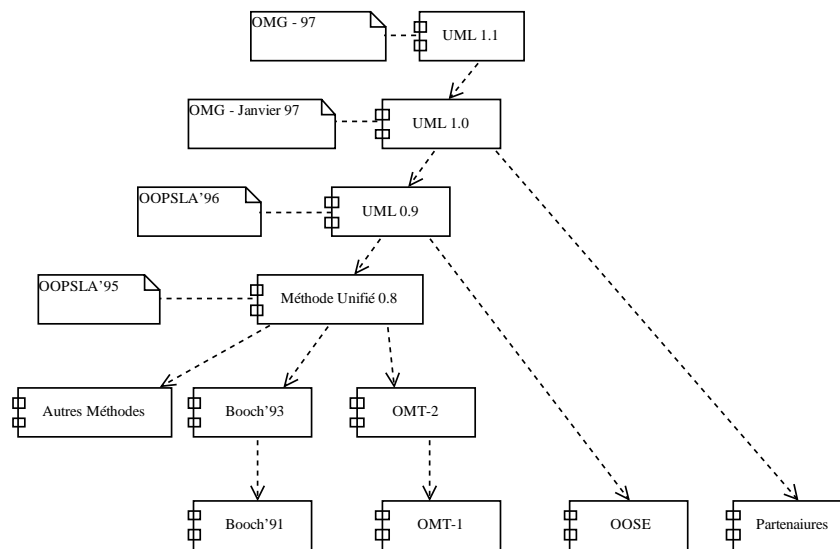
- concepts de modélisation de données (entité-relation),
- concepts de modélisation métier (workflow),
- concepts de modélisation objet.

Au travers des figures (fig. B.6(a) et fig. B.6(b)), nous retrouvons l'évolution de cette norme au cours du temps. UML s'est imposé, au cours du temps, comme le langage de référence en orienté objet dans le monde de l'ingénierie logicielle. D'une profusion de notations, il est aujourd'hui devenu un standard supporté par l'OMG (Open Management Group)¹

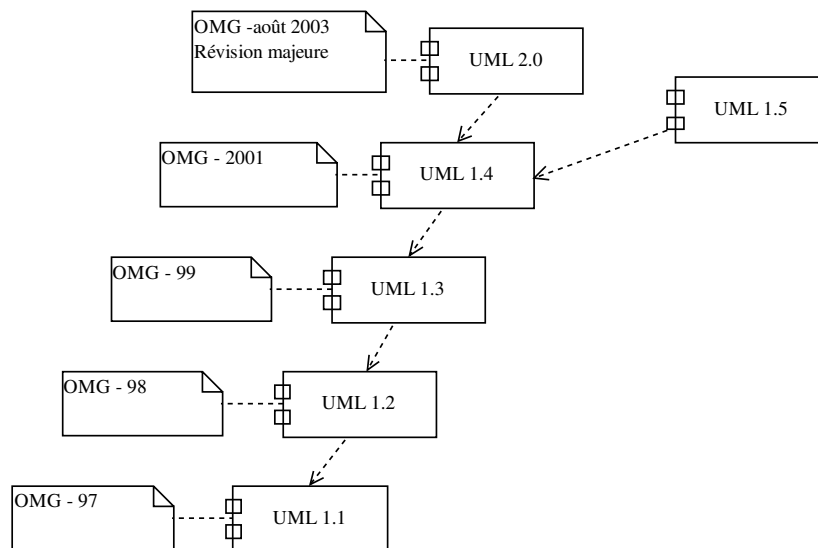
Nous pouvons considérer qu'il y a deux évolutions majeures dans la norme UML. L'une correspond au regroupement des différentes notations et à l'appropriation du standard par l'OMG en 1996. L'autre correspond à une évolution plus récente, le passage de UML 1.4 à UML 2.0.

La première réussite d'UML fut dans la capacité de Rational Software à réunir les trois principaux initiateurs de ce futur standard : G. Booch, J. Rumbaugh et I. Jacobson. Dans le même temps, l'unification de ces trois points de vue a permis de concevoir ce langage de modélisation et de le proposer à l'OMG qui, dans les années 1996, recherchait une méthodologie de modélisation.

¹<http://www.uml.org>



(a) de l'origine à 1996



(b) de 1997 à 2003

FIG. B.6 – Évolution d'UML

Principaux diagrammes

Les modèles sont basés sur des principes d'abstraction et sur un formalisme, c'est-à-dire des règles de représentation à l'aide de neuf diagrammes distincts. On parle alors de langage de modélisation.

Dans le langage de modélisation UML, il existe des diagrammes permettant de représenter toute la complexité d'un système d'ingénierie. Quatre catégories de diagrammes existent :

- représentation des besoins, du domaine et de son environnement :
 - cas d'utilisation (*Uses cases*),
- description de la structure statique du système :
 - diagramme d'objets (*Object diagram*),
 - diagramme de classe (*Class diagram*),
- description dynamique (temporelle) du système :
 - diagramme d'activité (*Activity diagram*),
 - diagramme de collaboration (*Collaboration diagram*),
 - diagramme d'états-transitions (*Statechart diagram*),
 - diagramme de séquence (*Sequence diagram*),
- description de l'architecture du système :
 - diagramme de composants (*Component diagram*),
 - diagramme de déploiement (*Deployment diagram*).

Nouvelle évolution majeure

Une évolution majeure d'UML, voyant le jour récemment, fut le passage de la version 1.4 à la version 2.0. UML était déjà très utilisé dans les projets informatiques ou plus généralement comme langage de modélisation pour des projets d'ingénierie. Le problème est que les concepts de programmation orientée objet (réutilisabilité, approche par composants) ont évolué par rapport à la spécification d'UML 1.4. Les principaux changements sont liés aux diagrammes dynamiques de la norme :

- le diagramme de séquence : la notion de réutilisabilité dans l'enchaînement d'opérations est maintenant prise en compte. Pour cela, il est nécessaire de faire référence à un diagramme de séquence représentant cette simple partie réutilisable ;
- le diagramme de déploiement et le diagramme de composant : les diagrammes de description d'architectures se sont enrichis des concepts des langages de description d'architecture (ADL - Architecture Description Language). Ces langages apportent les éléments de description d'un composant, d'un connecteur... De plus, un lien a été réalisé entre les éléments d'un composant (assemblage de classe) et leur représentation physique (fichiers).

De nouveaux diagrammes sont apparus afin de compléter la spécification et d'affiner la représentation d'un système. Nous retrouvons donc :

- le diagramme temporel (Timing Diagram) : ce diagramme permet de décrire l'évolution d'un élément (classe, activité...) suivant deux axes : ses états et le temps. L'évolution de cet élément est déclenchée par des événements, doit respecter des contraintes ou respecter de simples règles temporelles ;

- le diagramme d’interaction (Interaction Overview Diagram) : ce diagramme est, à la base, une variante du diagramme d’activité. Il permet de représenter les interactions entre les différents diagrammes (diagramme de séquence, diagramme temporel et diagramme de collaboration) composant le modèle.

Cette nouvelle version facilite la mise en place du Model-Driven Architecture (annexe D), améliore l’aspect « composant » en comblant les manques par rapport aux langages de description d’architecture et sur le rôle central du méta-modèle. Dans UML 2.0, le modèle, le méta-modèle et la transformation des modèles ont une place plus importante. Ce recentrage doit permettre de faciliter l’évolution de cette norme. Le développement des outils de modélisation se sera construit uniquement autour du méta-modèle d’UML.

L’évolution des spécifications de cette norme est très intéressante et elle jouera un rôle important dans les prochains développements. Malheureusement, la version finale de ces spécifications a tardé. Par conséquent, dans nos travaux de thèse, nous ne tiendrons pas compte de ces nouveaux diagrammes et nouveaux éléments qui les composent pour modéliser notre système. Nous nous baserons sur la version 1.4 d’UML.

Description du domaine

Un domaine représente la description des grandes fonctions de gestion de l’entreprise. Chaque domaine est composé de processus et d’informations. Ces informations correspondent à la description de la structure statique du domaine, alors que les processus correspondent à la description dynamique.

La description du domaine modélise aussi les relations entre l’utilisateur et le système. Cette description définit deux choses : les acteurs et leurs cas d’utilisation. Selon Pierre Alain Muller (Pierre Alain Muller, 1999), « *un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système* ».

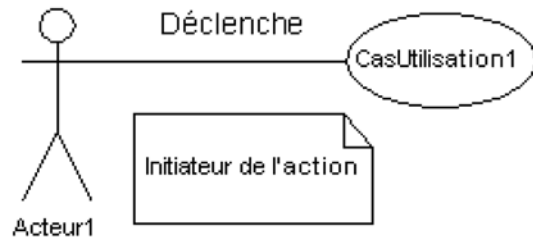
Il existe différents types d’acteurs :

- *les acteurs principaux* : il s’agit des personnes qui utilisent les fonctions principales du système. Dans le cas d’une application, il s’agit des utilisateurs du système,
- *les acteurs secondaires* : il s’agit des personnes qui effectuent des tâches administratives ou de maintenance. Dans notre cas, il s’agit des administrateurs du système,
- *le matériel externe* : il s’agit du matériel incontournable qui fait partie du domaine de l’application et qui doit être utilisé. Dans notre cas, il s’agit donc des ordinateurs ou des différents périphériques mis à notre disposition,
- *les autres systèmes* : il s’agit des systèmes avec lesquels le système doit interagir. Dans notre cas, cela peut être un travail de collaboration avec une autre application.

La description des acteurs doit nous permettre, en les observant ou à partir d’un cahier des charges, de déterminer les cas d’utilisation correspondants. Il est aussi possible de déterminer quels acteurs agissent dans un cas d’utilisation. Un cas d’utilisation représente la relation entre un acteur et une fonctionnalité du système. Dans la représentation de notre modèle, nous pouvons considérer que les processus¹ sont équivalents à des cas d’utilisation.

Un cas d’utilisation est donc une technique d’élaboration des besoins fonctionnels, selon le point de vue d’un acteur ou d’une catégorie d’utilisateurs (regroupement de plusieurs acteurs). Il existe différents formalismes pour la représentation des cas d’utilisation.

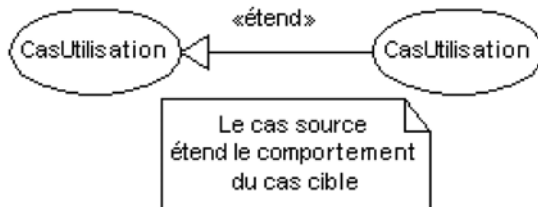
¹« Un processus opérationnel est une succession de tâches qui contribue à la réalisation des objectifs de l’entreprise. De manière générale, un processus peut être défini comme un enchaînement d’activités à exécuter pour atteindre un but donné.» (François Vernadat, 1996)



La relation de communication est le déclenchement par un acteur d'un cas d'utilisation.

Exemple :

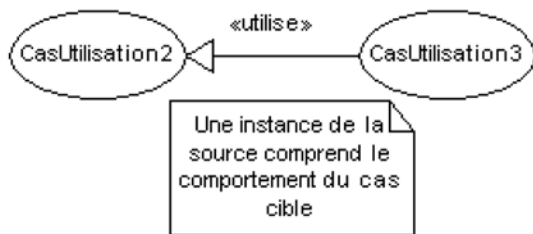
Un client veut émettre un ordre de commande.



La relation d'extension consiste à étendre les possibilités d'un cas d'utilisation par une variante.

Exemple :

Un client veut émettre un ordre de commande via Internet.



La relation d'utilisation consiste à utiliser un autre cas d'utilisation pour en récupérer son comportement.

Exemple :

L'ordre de commande ne peut être validé qu'après identification du client.

Le diagramme (fig. B.7) donne un exemple de mise en œuvre des différentes relations entre cas d'utilisation. Le passage d'un ordre via Internet est une extension d'un ordre de commande « classique », mais dans les deux cas le client doit être identifié.

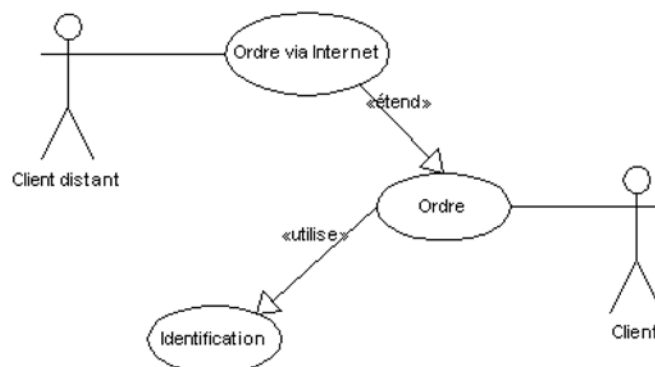


FIG. B.7 – Exemple de mise en œuvre des relations entre cas d'utilisation

Description de la structure statique

La structure statique est la représentation, à l'aide de diagrammes, des classes, de l'ensemble des objets ainsi que les relations entre ces ensembles.

Les **diagrammes de classes** n'expriment rien de particulier sur les liens d'un objet donné, mais décrivent de manière abstraite les liens potentiels entre des classes. Le méta-modèle de la représentation des classes, fig. B.8, montre l'ensemble des classes et liens possibles.

Les **diagrammes d'objets** s'utilisent pour montrer un contexte dans l'utilisation de la classe, mais aussi pour faciliter la compréhension des parties complexes d'un diagramme de classe.

Toutes les informations mémorisées, manipulées, transformées, analysées et partagées sont des informations élémentaires à un domaine. Ces informations ne doivent figurer qu'une seule fois dans la modélisation pour respecter une règle importante de *non-redondance* de l'information.

Ces informations sont nécessaires pour accomplir les objectifs de chaque domaine.

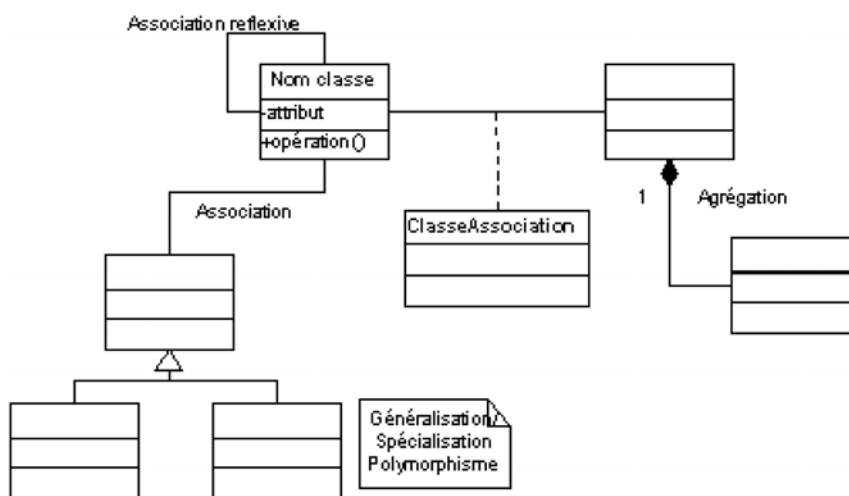


FIG. B.8 – Extrait du méta-modèle de la représentation des classes

Les classes

Les classes ne sont pas des fonctions. Selon Pierre Alain Muller, les classes sont des descriptions abstraites – condensées – d'un ensemble d'objets du domaine du Système d'Information. Les classes sont décrites à l'aide de trois compartiments :

- le nom de la classe : représente ce qu'elle est, et non ce qu'elle fait,
- les attributs : représentent un ensemble d'informations élémentaires sur les instances de la classe,
- les opérations : représentent un ensemble d'opérations applicables aux instances de la classe.

À partir des classes, un ensemble de relations peut les relier. Ces relations sont :

- une association (fig. B.9) est une relation entre deux classes ;
- une agrégation (fig. B.10) est une association non symétrique ;
- une composition (fig. B.11) est une agrégation matérialisant une relation d'inclusion ;
- une généralisation (fig. B.12) est une relation de classification entre un élément plus général et un élément plus spécifique.

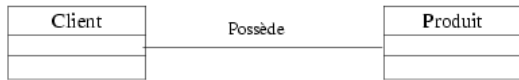


FIG. B.9 – Relation d'association

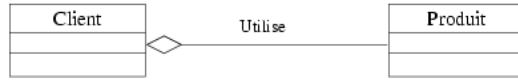


FIG. B.10 – Relation d'agrégation



FIG. B.11 – Relation de composition

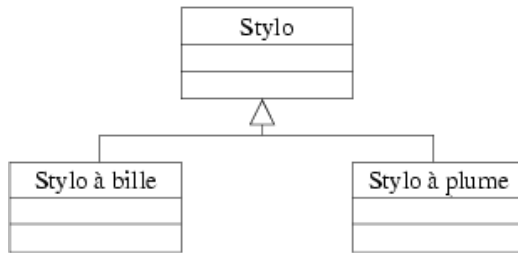


FIG. B.12 – Relation de généralisation

Les associations

Les associations représentent des relations structurelles entre classes d'objets. Un ensemble de descripteurs est lié aux associations, tel que :

- arité des associations : les associations sont la plupart du temps binaires, mais peuvent être aussi ternaires, etc. ;
- nommage des associations : le nom des associations permet de représenter plus précisément l'utilisation de celles-ci ;
- nommage des rôles : un rôle décrit comment une classe voit une autre classe au travers de l'association ;
- multiplicité des associations : chaque rôle porte une indication qui montre combien d'objets peuvent être liés à un autre objet d'une classe ;
- contraintes sur les associations (par exemple, l'indication de la multiplicité de l'association).

Les classes-associations

Selon Pierre Alain Muller, une association peut être représentée par une classe pour ajouter des attributs et des opérations dans l'association. Par exemple, la fig. B.13, une classe « Commande » peut porter sur une classe « Produit » et être associée à une « quantité commandée ».

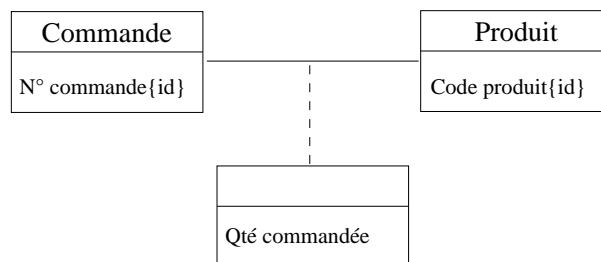


FIG. B.13 – Association porteuse d'attribut

L'agrégation

L'agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité.

La généralisation

La généralisation est employée pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. En fait, le terme généralisation désigne un point de vue porté sur un arbre de classification. L'exemple de (Pierre Alain Muller, 1999), fig. B.12, montre qu'un animal est un concept plus général qu'un chat ou chien, et inversement un chat est un concept plus spécialisé qu'un animal.

La spécialisation

La spécialisation est la vision inverse de la généralisation.

Le polymorphisme

Le polymorphisme décrit la caractéristique d'un objet qui peut prendre plusieurs formes. Ainsi, un objet a la possibilité de déclencher des opérations spécifiques en réponse à un message général. Chaque sous-classe hérite de la spécialisation des opérations de ses sur-classes, mais a la possibilité de modifier localement le comportement de ces opérations pour mieux prendre en compte ses particularités.

Description du modèle dynamique

La description du modèle dynamique permet de représenter les différents processus, activités ou opérations à effectuer pour atteindre les objectifs définis dans les deux premières phases de la conception de notre Système d'Information.

Nous avons à notre disposition un ensemble de quatre diagrammes UML, avec la répartition suivante :

- les diagrammes d'interactions :
 - les diagrammes de collaboration,
 - les diagrammes de séquence.
- les diagrammes d'états :
 - les diagrammes d'états-transitions,
 - les diagrammes d'activité.

Ces diagrammes seront présentés plus en détail dans les paragraphes suivants. Ils permettent de décrire, plus ou moins précisément, les différents événements qui existent entre les utilisateurs et le système (les objets) ainsi que les opérations qui leurs sont affectées.

Les **diagrammes de collaboration** montrent des interactions entre objets en insistant plus particulièrement, par rapport aux diagrammes de séquence, sur la structure statique qui permet la mise en collaboration d'un groupe d'objets. Ainsi, ils permettent une meilleure représentation du contexte des objets. Les diagrammes de collaboration sont une extension des diagrammes d'objets.

Les **diagrammes de séquence** montrent des interactions entre objets selon un point de vue temporel. Le contexte des objets n'est pas représenté de manière explicite, comme dans les diagrammes de collaboration où la représentation se concentre sur l'expression des interactions.

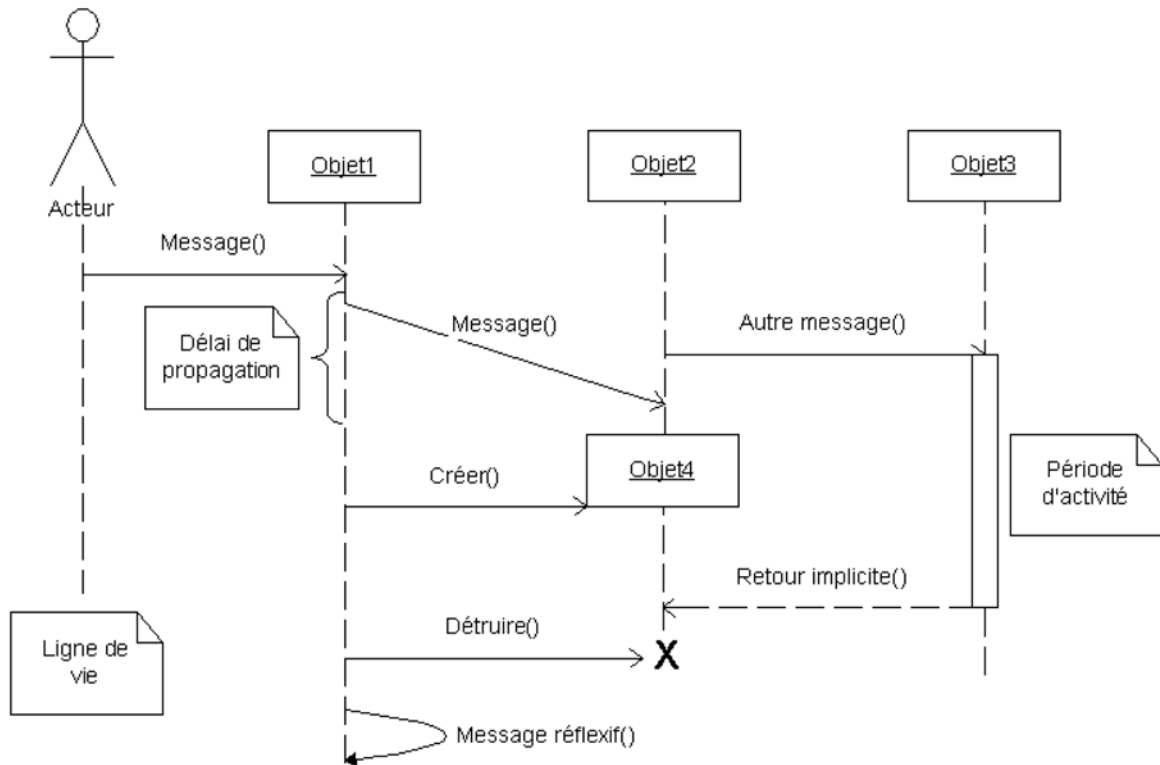


FIG. B.14 – Extrait de la représentation des diagrammes de séquence

Une barre verticale, ligne de vie, est construite à partir d'un acteur ou d'un objet. Les communications, ou messages¹ entre acteurs ou objets sont représentées par des flèches entre les lignes de vies. Par rapport au diagramme de collaboration, ce diagramme permet d'identifier plus facilement l'ordre temporel de l'envoi des messages, en suivant un ordre de lecture de haut en bas. À partir de l'envoi des messages², il est possible de créer ou détruire des objets ou d'avoir une période d'activités pendant laquelle un certain nombre de changements sont effectués.

Les **diagrammes d'états-transitions** (fig. B.15) permettent de représenter des automates. Les automates sont la description du comportement des objets. Le comportement d'un objet est décrit à partir de trois éléments essentiels : les états du système, les transitions et les événements.

Selon Pierre-Michel Riccio (Pierre-Michel Riccio, 1999), « l'état caractérise une période de

¹Un message est un élément de communication entre objets et acteurs qui déclenche une activité dans l'objet destinataire; la réception d'un message correspond à un événement.

²L'envoi de message peut être l'activation de méthodes ou le changement de valeur d'attributs des objets.

stabilité du système (un système est toujours dans un état). La transition matérialise le passage d'un état à un autre. L'événement, qui déclenche le changement d'état, est précisé sous une forme textuelle à proximité des transitions. »

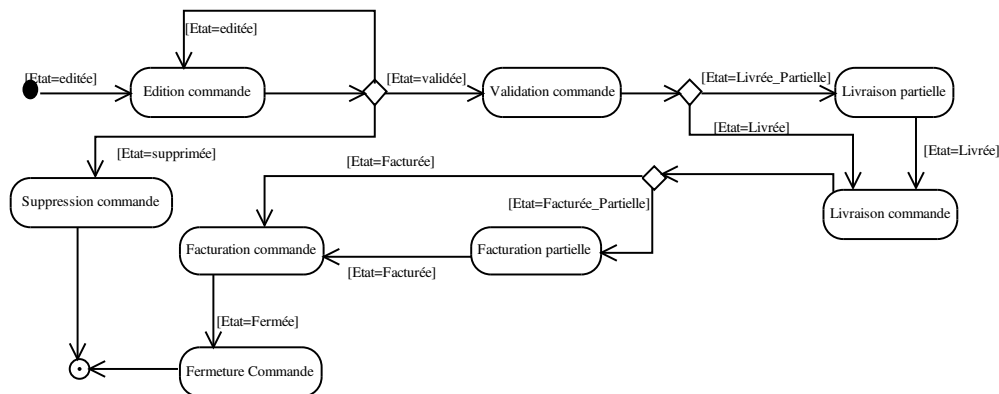


FIG. B.15 – Diagramme d'états-transitions

À chaque état, il est possible de représenter un ensemble d'actions ou d'opérations à effectuer pour résoudre un problème et passer à un autre état, une autre situation.

Les **diagrammes d'activité** (fig. B.16) sont une variante des diagrammes d'états-transitions, organisés par rapport aux actions et principalement destinés à représenter le comportement interne d'une méthode ou d'un cas d'utilisation.

Le modèle dynamique peut se diviser en deux parties :

- l'analyse des scénarios fonctionnels,
- l'analyse de l'état des objets.

L'analyse des scénarios fonctionnels

Elle sert à décrire les éléments constitutifs d'une activité ou d'une opération (processus du modèle des besoins). Cette notion représente une partie de l'interaction entre le système (les objets) et son environnement (les acteurs).

Les scénarios peuvent être représentés à l'aide de plusieurs diagrammes. Ils décrivent plus précisément les différents cas d'utilisation montrant différentes mises en œuvre du système. Nous pouvons donc utiliser :

- des diagrammes de séquence gérant l'organisation des événements et des messages dans le temps,
- des diagrammes d'activité reliant les acteurs à leurs activités,
- les diagrammes de collaboration, très proches des diagrammes de séquence, sont moins faciles d'utilisation dans ce cas, car ils montrent difficilement l'aspect temporel du scénario.

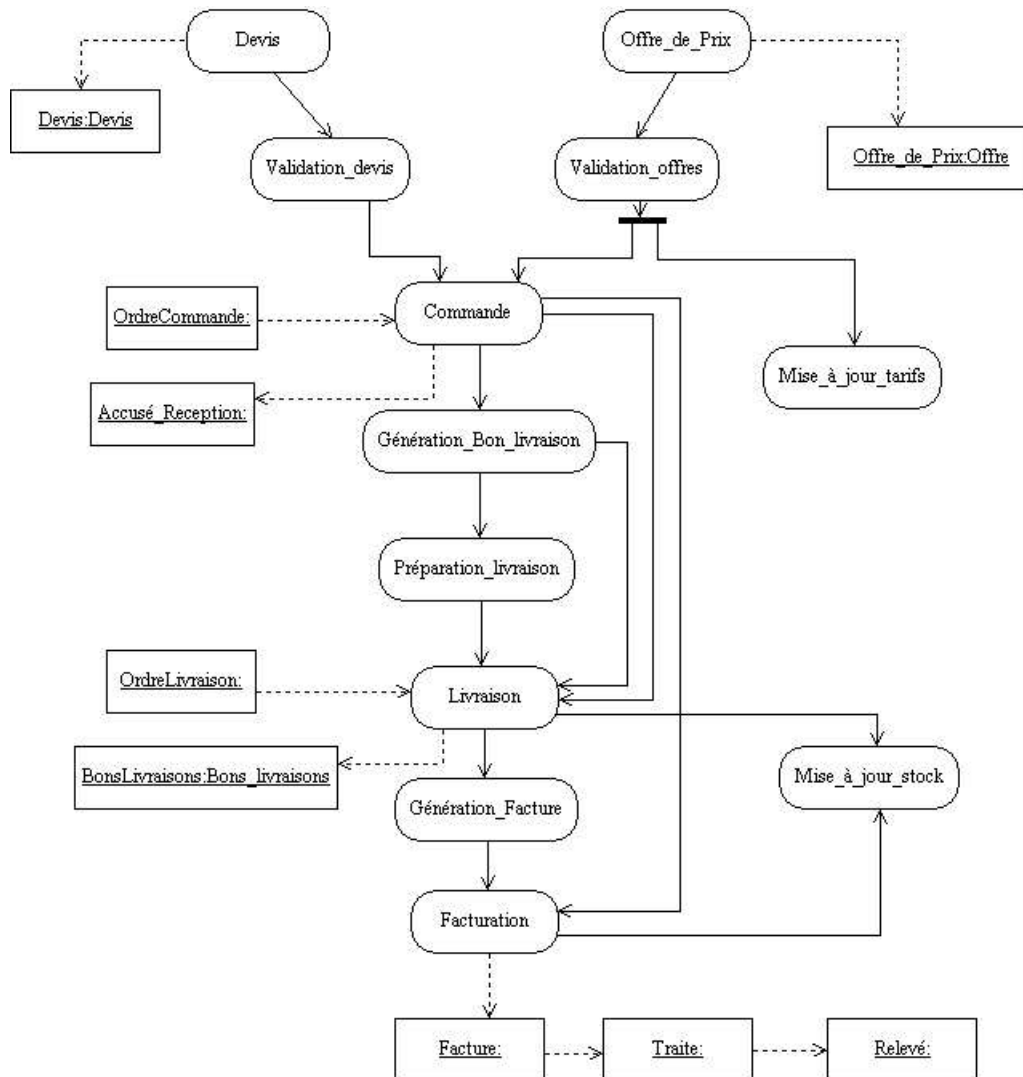


FIG. B.16 – Diagramme d'activités

L'analyse de l'état des objets

Selon Mokrane Bouzeghoub et Arnold Rochfeld (Mokrane Bouzeghoub et Arnold Rochfeld, 2000), elle définit la description de l'ensemble des états pertinents qu'un objet peut avoir, ainsi que l'ensemble des événements qui font passer un objet d'un état à un autre.

La dynamique d'un objet peut être décrite suivant les états successifs d'un ou plusieurs attributs d'un objet, ceci est représenté à l'aide du diagramme d'états-transitions.

Description de l'architecture du Système d'Information

La description de l'architecture du Système d'Information montre comment ses composants interagissent par rapport aux différents logiciels ou matériels donnés.

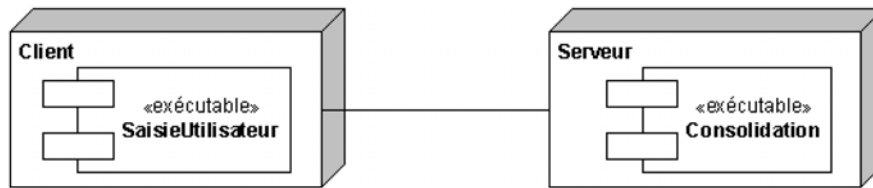


FIG. B.17 – Représentation d'une architecture via les diagrammes UML

L'architecture logicielle

L'architecture logicielle est la représentation des différents composants, les éléments physiques du Système d'Information. Cette représentation se fait à l'aide des modules, des diagrammes de composants, qui peuvent être soit des bibliothèques, de la documentation, des exécutables, des fichiers ou des tables. Ces différents stéréotypes¹ (bibliothèque, documentation, etc.) permettent de dégager une notion de partition des éléments du Système d'Information, déjà véhiculée par les paquetages. Ils permettent aussi une meilleure lecture, compréhension et organisation du modèle de l'architecture du Système d'Information.

L'architecture matérielle

L'architecture matérielle est la représentation de la disposition physique des différents matériels qui entrent dans la composition d'un système.

Chaque ressource matérielle est représentée par un cube, appelé nœud, évoquant la présence physique. Chaque nœud peut être composé d'un module, ou programme, du Système d'Information.

UML pour les systèmes d'ingénierie

Modélisation des systèmes d'ingénierie

Plusieurs raisons permettent de préconiser l'utilisation UML (Unified Modeling Language) pour représenter différents systèmes d'ingénierie.

¹Extension de la sémantique d'un élément du méta-modèle.

En outre, il est adopté et utilisé par la communauté des informaticiens pour le développement de solution logicielle, mais il pénètre également en force le domaine de l'ingénierie comme moyen pratique de spécifier la conception d'un système technique. C'est-à-dire qu'il est aussi bien utilisé par la Maîtrise d'Œuvre, de l'expression du besoin à la conception jusqu'à la génération de code, que par la maîtrise d'ouvrage pour la réalisation et la représentation du cahier des charges par exemple.

Pour terminer, le plus intéressant est dans le fait qu'aucune sémantique métier n'est attachée à UML. Nous pouvons utiliser UML dans des contextes très variés : la représentation d'un système temps réel, la modélisation des éléments composant une architecture système par exemple.

Pour cela, UML propose de définir des profils dédiés à un contexte particulier. La définition d'un profil consiste à créer, spécifier et assembler des classes afin de représenter un méta-modèle (apsec :meta-modele). Par la suite, suivant le principe d'instanciation, nous pourrions instancier ce méta-modèle pour obtenir le modèle spécifique.

L'autre solution, pour utiliser UML dans des contextes variés, est dans l'apport d'une sémantique liée au domaine d'analyse. Chaque élément de la norme UML possède une syntaxe et une sémantique. Mais celle-ci est limitée à la compréhension de l'élément et non liée au contexte de modélisation.

Mais Unified Modeling Language n'est qu'une convention de notation. Cette notation permet de faire dialoguer les parties prenantes à partir d'un formalisme unique et donc compréhensible par tout le monde. À l'opposé, UML doit être complété par la mise en œuvre d'une méthode de conception décrivant l'enchaînement des activités et la fourniture de diagrammes permettant la réalisation de la solution logicielle.

Annexe C

Méthodes de modélisation d'entreprise

IDEF

Dans les années 1970, l'U.S. Air Force a développé, dans le cadre du projet ICAM (Integrated Computer Aided Manufacturing), des systèmes informatisés pour augmenter la productivité des industries manufacturières. Le programme ICAM a identifié le besoin d'une meilleure technique d'analyse et de communication pour les personnes travaillant sur l'amélioration de la productivité.

En conséquence, le programme ICAM a développé une série de techniques connues sous le nom d'IDEF (Icam DEFinition). Ces techniques sont :

1. IDEF0 : utiliser pour obtenir un « modèle fonctionnel ». Le modèle fonctionnel est la représentation structurée des fonctions, activités ou process du système modélisé ou du sujet d'étude.
2. IDEF1 : utiliser pour obtenir un « modèle informationnel ». Le modèle informationnel représente la structure et la sémantique du système modélisé ou du sujet d'étude.
3. IDEF2 : utiliser pour obtenir un « modèle dynamique ». Le modèle dynamique représente les caractéristiques comportementales par rapport au temps du système modélisé ou du sujet d'étude.

Actuellement, les techniques IDEF0 et IDEF1X (IDEF1 eXtend) sont largement répandues dans les secteurs industriels et commerciaux supportant un grand éventail d'entreprises et de domaines d'application.

IDEF0 / SADT

IDEF0 (Integration DEFinition language 0) est basé sur le modèle SADT (Structured Analysis Design Technique). L'originalité de ce modèle est dans la définition d'un langage de modélisation graphique (d'un point de vue de la syntaxe et de la sémantique) et dans la description d'une méthodologie pour le développement du modèle.

Les deux principaux composants du modèle sont :

- Les fonctions : représentées dans un diagramme composé de différents symboles géométriques.
- Les données et les objets en interrelations avec les fonctions : représentés par des flèches.

IDEF0	Function Modeling
IDEF1	Information Modeling
IDEF2	Simulation Modeling
IDEF1X	Data Modeling
IDEF3	Process Description Capture
IDEF4	Object-oriented Design
IDEF5	Ontology Description Capture
IDEF6	Design Rationale Capture
IDEF7	Information System Audit Method
IDEF8	User Interface Modeling
IDEF9	Scenario-driven Info Sys Design Spec
IDEF10	Implementation Architecture Modeling
IDEF11	Information Artifact Modeling
IDEF12	Organisation Modeling
IDEF13	Three Schema Mapping Design
IDEF14	Network Design

TAB. C.1 – Liste des méthodes IDEF

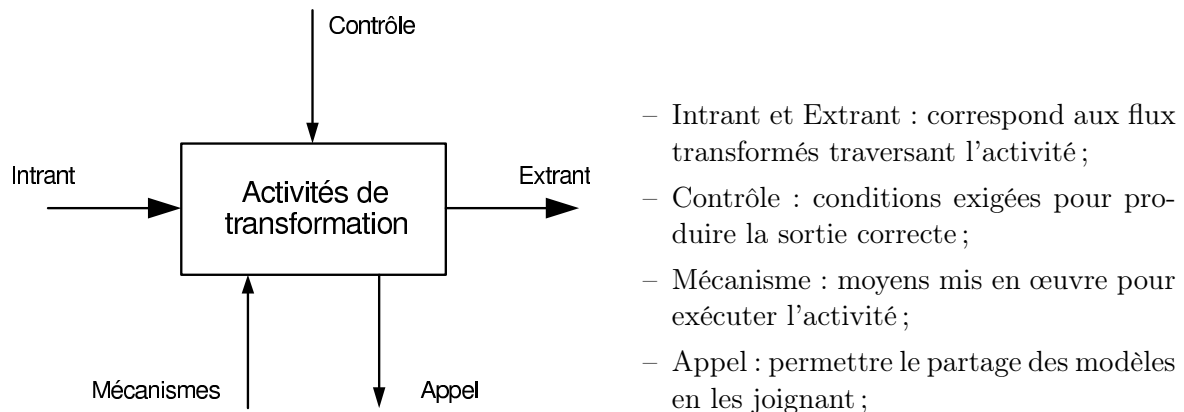


FIG. C.1 – Activité vue par IDEF0

Ce modèle est composé d'un ensemble de nœuds représentant les fonctions et lié par des flèches portant sur les entités servant d'entrées ou de sorties aux fonctions.

Le résultat d'IDEF0 est la représentation des fonctions et du flux informationnel du modèle. Cette représentation s'organise par une suite de diagrammes, textes, et glossaires imbriqués.

IDEF1

La méthode IDEF1 a été conçue pour développer des modèles reflétant l'intégration de l'ensemble des informations de l'entreprise. L'approche d'IDEF1 est la suivante :

1. établir un modèle informationnel,
2. concevoir la base de données du modèle,
3. implémenter et installer la base de données ainsi que les fonctions et procédures associées.

La méthode IDEF1 offre un ensemble de règles et procédures pour la création des modèles informationnels. Elle incorpore les graphiques, le texte et les formes nécessaires pour la meilleure description du modèle.

Il existe deux composants fondamentaux à cette représentation :

- Les diagrammes : les caractéristiques du modèle informationnel, représentées selon un ensemble de règles et de procédures construisant une description significative de l'information.
- Le dictionnaire : chaque élément du modèle est décrit à travers le texte pour obtenir une meilleure définition du modèle informationnel.

IDEF1X

IDEF1X est un langage de modélisation (d'un point de vue sémantique et syntaxique) et associé à des règles et des procédures, nous pouvons l'utiliser pour développer des modèles logiques de données. IDEF1X est donc employé pour produire les modèles informationnels qui représentent la structure sémantique de l'information au sein de l'entreprise.

L'utilisation de cette norme permet la construction des modèles de données qui peuvent servir à supporter la gestion des données comme des ressources, l'intégration des Systèmes d'Information et la construction des bases de données. Il est basé sur le formalisme entité-relation étendu.

Cet outil est employé pour modéliser les données d'une manière standard, cohérente et précise afin de les contrôler. Les premiers objectifs de ce standard sont de :

- fournir des moyens pour comprendre et analyser les données ;
- fournir des moyens communs de représentation et de communication sur la complexité des données ;
- fournir une méthode pour représenter la vue globale des données d'une entreprise.

IDEF2

IDEF2 est un langage de modélisation du comportement d'un système basé sur le principe des files d'attente. C'est une méthode complémentaire à SADT/IDEF0 qui vise à répondre aux lacunes du point de vue analyse des aspects dynamiques d'un système. Cette méthode est basée sur quatre modèles : système physique, flux des entités, contrôle du système et gestion des ressources. IDEF2 aborde donc de façon privilégiée les vues informationnelles et ressources sur des niveaux d'abstraction proche de l'exécution.

IDEF3

La méthode IDEF3 a été proposée en 1992 pour palier aux carences d'IDEF0 en matière de modélisation du comportement de l'entreprise, donc la représentation des processus opérationnels.

IDEF3 modélise, comme IDEF0, les processus sous forme d'un enchaînement d'étapes, appelées unités de comportement, connectées par des boîtes de jonction et des liens. L'avantage, par rapport à l'utilisation d'IDEF0, est la possibilité de décrire un flux de contrôle du processus de manière plus complexe. Par exemple, les boîtes de jonctions permettent de représenter à l'aide de connecteurs logiques (AND, OR ou XOR) les différents flux, les phénomènes de séquencement, de parallélisme ou de synchronisation d'activités.

Avantages – inconvénients de la méthode

Cette suite de méthode IDEF est très employée dans le monde industriel américain. Elle a l'avantage d'un grand nombre de possibilités et de formalisme pour l'ensemble des représentations possibles (IDEF0-14). Son grand nombre de modèles peut permettre de gérer les systèmes complexes, malheureusement ceci peut être aussi un inconvénient, car il est nécessaire de bien les maîtriser.

La suite IDEF n'a pas été conçue en priorité pour la gestion des ressources et organisation de l'entreprise, seul IDEF12 la spécifie. Et à ma connaissance il ne gère pas, non plus, l'aspect décisionnel d'un système comme la méthode GRAI.

IDEF :

<p>Aspect Fonctionnel :</p> <p>IDEF0 : méthode d'analyse fonctionnelle</p> <p>IDEF3 : modélisation du comportement selon le point de vue des processus opérationnels.</p> <p>IDEF0 :</p> <ul style="list-style-type: none"> - activité : une fonction qui transforme des objets d'entrée en objets de sortie ; - interconnectivité des activités : représente des flux de données, des flux de matières ou l'utilisation des ressources. <p>IDEF3 :</p> <ul style="list-style-type: none"> - unité de comportement : est une étape dans un processus ; - boîte de jonction : lien logique entre les différentes unités de comportement ; 	<p>Aspect Informationnel :</p> <p>IDEF1 et IDEF1X servent à décrire les entités de l'entreprise, leurs relations et leurs différents états possibles, c'est-à-dire quels sont les objets traités et comment ils sont gérés.</p> <ul style="list-style-type: none"> - entité : un ensemble d'objets réels ou abstraits, ayant chacun une existence propre ; - relation : une relation est une association entre deux entités, elle n'a pas d'attributs ; - attribut : un attribut est une propriété de l'entité.
--	--

TAB. C.2 – Description du vocabulaire de la suite IDEF

Références : (François Vernadat, 1999) et <http://www.idef.com>

CIMOSA

L'approche CIMOSA (CIM Open System Architecture) permet de diviser le modèle d'entreprise en quatre vues : la vue fonctionnelle, la vue informationnelle, la vue des ressources, la vue organisationnelle.

Démarche de modélisation

La démarche de modélisation en entreprise, en utilisant CIMOSA, passe par les étapes suivantes :

- une analyse des domaines fonctionnels de l'entreprise et leurs relations,
- l'identification des processus maîtres à modéliser,
- l'analyse détaillée des processus maîtres selon les principaux constructs définis dans la méthode (à savoir les événements, sous-processus, activités, aptitudes et vues d'objets),
- la consolidation du modèle au niveau spécification de conception,
- l'adaptation du modèle aux contraintes d'implantation,
- la traduction du modèle dans le langage des systèmes utilisés.

Cette démarche n'a lieu d'être que pour répondre aux différents points présentés justifiant la mise en place d'un modèle d'entreprise.

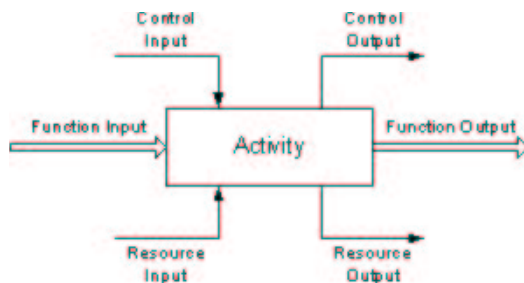
Pour répondre aux exigences des étapes précédentes, nous avons à notre disposition un ensemble de vues nous permettant de décrire le modèle d'entreprise.

Vue Fonctionnelle

CIMOSA considère que toute entreprise se compose :

- d'un grand ensemble de processus communicants chargés de réaliser les objectifs de l'entreprise,
- d'un ensemble d'entités fonctionnelles réalisant les processus opérationnels en fonction de l'état du système.

Pour représenter cette entreprise, le modèle fonctionnel décrit l'enchaînement des différents processus, soit réalisés par des humains, soit par des machines. Ce modèle est hiérarchisé sur trois niveaux : les processus, les activités (fig. C.2) et les opérations.



« Une activité est une étape élémentaire d'un processus. C'est le lieu de l'action et en tant que tel, elle nécessite du temps et des ressources pour l'exécution de chacune de ses *opérations fonctionnelles*. » (François Vernadat, 1999)

FIG. C.2 – Activité vue par CIMOSA

Elle fournit donc les éléments de modélisation (ou constructs) suivants (François Vernadat, 1999) :

- domaine : un élément structurant permettant de regrouper un ensemble de processus en un module indépendant de manière à gérer la complexité du modèle ;
- événement : un événement est un fait instantané qui signale un changement d'état dans le système et nécessitant une action ;
- processus : un processus est une séquence partiellement ordonnée d'étapes, déclenchée par un événement pour atteindre un but donné ;
- activité : une activité est une étape élémentaire d'un processus. Elle nécessite du temps et des ressources.

Vue informationnelle

L'entreprise possède un ensemble de données et d'informations utilisées, enregistrées et traitées pour les besoins des utilisateurs et des applications. Ces données et informations peuvent être organisées et hiérarchisées pour une meilleure compréhension.

Elle fournit donc les éléments de modélisation suivant (François Vernadat, 1999) :

- objet d'entreprise : les objets d'entreprise sont les entités existantes dans l'entreprise et intervenant dans les activités ;
- vue d'objet : une vue d'objet est un état ou manifestation d'un objet d'entreprise à un instant donné.

Vue des ressources

L'entreprise dispose de différents types de ressources, telles que :

- les ressources *humaines* : le commercial, le comptable, le directeur, etc. Chaque ressource possède des compétences et des rôles leur sont attribués dans leurs propres domaines d'action. Les ressources sont organisées à travers la vue organisationnelle de l'entreprise.
- les ressources *matérielles* : il peut s'agir de moyens informatiques et de communications (ordinateur, etc.), de lieux et moyens de stockage (entrepôts, etc.) ou encore de moyens de production (machine-outil, etc.).

Elle fournit donc les éléments de modélisation suivants (François Vernadat, 1999) :

- ressource : une ressource est un objet d'entreprise jouant un rôle de support à l'exécution d'une activité ;
- ensemble d'aptitudes : un ensemble d'aptitudes définit les fonctionnalités potentielles d'une entité fonctionnelle ;
- opération fonctionnelle : une opération fonctionnelle est un granule de fonctionnalités offert par une entité ou requis par une activité.

Vue organisationnelle

D'après Henry Mintzberg (Henry Mintzberg, 1980), l'organisation d'une entreprise se décompose d'une part en une structure organisationnelle et d'autre part dans la coordination des tâches qui composent cette structure.

Avec la vue organisationnelle, nous définissons la structure organisationnelle, la distribution des rôles et du travail en termes de responsabilité et d'autorité entre les différents membres ou divisions de l'entreprise.

Elle fournit donc les éléments de modélisation suivants (François Vernadat, 1999) :

- structure organisationnelle : la structure organisationnelle d'une entreprise correspond à la description des entités organisationnelles et des rapports qu'elles ont entre elles, ainsi qu'aux liaisons hiérarchiques existant entre leurs responsables à différents niveaux ;
- coordination des tâches : la coordination des tâches correspond à un ensemble de mécanismes permettant aux entités organisationnelles de s'informer mutuellement, de se consulter ou de décider ensemble pour gérer les tâches en fonction des objectifs à atteindre et de la complexité de l'organisation.

Avantages – inconvénients de la méthode

CIMOSA est une méthodologie surtout connue au niveau européen, à l'aide entre autre du co-concepteur de cette méthode : François Vernadat.

Son utilisation se situe surtout autour du monde scientifique de la modélisation d'entreprise. Cette méthode permet de gérer la globalité du Système d'Information. Le seul regret peut être dans sa gestion des décisions au niveau de la vue organisationnelle. Mais certains documents discutent de l'implémentation et de la cohabitation des méthodes CIMOSA (Système d'Information et Système Opérant) et GRAI (Système de Décision) d'un point de vue organisationnel et décisionnel.

CIMOSA :

<p>Vue fonctionnelle : sert à décrire la fonctionnalité et le comportement de l'entreprise en termes de processus, d'activité et d'opérations.</p> <ul style="list-style-type: none"> - processus : un processus opérationnel est une succession de tâches qui contribue à la réalisation des objectifs de l'entreprise; - activité : une activité est une étape élémentaire d'un processus; - opération : une opération est une étape élémentaire d'une activité. Elle ne peut être interrompue. 	<p>Vue informationnelle : sert à décrire les entités de l'entreprise, leurs relations et leurs différents états possibles, c'est-à-dire quels sont les objets traités et comment ils sont gérés.</p> <ul style="list-style-type: none"> - entité : un ensemble d'objets réels ou abstraits, ayant chacun une existence propre; - relation : une relation est une association entre deux entités au moins; - attribut : un attribut est une caractéristique descriptive d'une entité ou d'une relation.
<p>Vue des ressources : sert à décrire les moyens nécessaires pour réaliser les fonctions de l'entreprise, leur rôle et leur mode de gestion.</p> <ul style="list-style-type: none"> - ressource : une ressource est un objet d'entreprise jouant un rôle de support à l'exécution d'une activité; - aptitude : un ensemble d'aptitudes définit les fonctionnalités potentielles d'une ressource. 	<p>Vue organisationnelle : sert à décrire la distribution des responsabilités et des autorités dans les prises de décisions.</p> <ul style="list-style-type: none"> - structure organisationnelle : elle décrit les entités organisationnelles et les rapports qu'elles ont entre elles, au sein d'une entreprise.

TAB. C.3 – Description du vocabulaire de CIMOSA

Références : (François Vernadat, 1996), (François Vernadat, 1999) et <http://www.cimosa.de/>

GRAI

La méthode GRAI (Graphes à Résultats et Activités Interreliés) est une méthode basée sur la modélisation d'entreprise ayant pour but la conception ou la reconception des systèmes de production (industriels ou de services). La méthode GRAI se focalise sur la partie décisionnelle d'un système et elle s'applique dans une optique générale d'amélioration des performances (Michel Roboam, 1993).

La méthode GRAI :

- est construite à partir d'un modèle de référence, le modèle GRAI, qui permet de construire le modèle global de l'entreprise,
- s'appuie sur des formalismes graphiques qui représentent les concepts du modèle GRAI et qui facilitent la communication et l'interprétation,
- suit une démarche structurée et participative dans laquelle les acteurs et les étapes sont définis, et propose un ensemble de règles permettant d'optimiser la performance du modèle ou de reconcevoir le système.

Le principal périmètre fonctionnel de la méthode GRAI correspond aux fonctions suivantes :

- planification,
- gestion des approvisionnements,
- gestion des achats,
- gestion des ressources.

Ce périmètre peut s'étendre aux fonctions annexes, telles que le contrôle de la qualité, la conception de produits etc.

Dans la description de la méthode GRAI, nous nous attacherons principalement dans la description de la grille et du réseau GRAI.

Grille GRAI

La grille GRAI est la représentation d'une vision globale et macroscopique de la structure du système de production étudié. Elle situe les centres de décisions les uns par rapport aux autres. Elle permet donc d'identifier :

- les centres de décisions et leurs interactions,
- la structure des centres de décisions,
- les liens décisionnels et informationnels entre les centres de décision,
- les informations externes et internes.

Le graphisme de la grille GRAI respecte les critères de décomposition utilisés par la méthode GRAI :

- l'horizon (H) et la période (P) d'une prise de décision,
- le regroupement des activités décisionnelles sur un niveau donné selon leurs fonctionnalités.

Les colonnes de la grille correspondent aux fonctions du système étudié, tandis que les lignes représentent le rapport horizons/périodes des prises de décisions.

L'intersection entre ces deux dimensions est donc appelée centre de décision. Les centres de décisions sont reliés les uns aux autres par des liens de deux types :

- des informations (flèche simple) : il s'agit d'une liaison informationnelle entre la provenance de l'information (externe ou interne) et le centre de décision ;
- des transmissions d'objectifs (double flèche) : il s'agit d'une liaison décisionnelle entre deux centres.

Nous pouvons synthétiser les différentes composantes de la grille GRAI avec deux exemples. Ces exemples permettent de décrire deux types d'organisation d'entreprise différents : l'un travail à partir des différents plans stratégiques (fig. C.5), l'autre ne travaille qu'à l'affaire (gestion par projet) (fig. C.4). Il est possible d'empiler les grilles GRAI afin de représenter, par exemple, les ressources nécessaires pour un centre de décision.

Réseau GRAI

Le réseau GRAI représente le fonctionnement de tout ou partie d'un centre de décision selon les concepts du modèle GRAI (modèle définissant les activités d'un centre de décision).

Les réseaux GRAI (fig. C.3) sont composés de deux types d'activités :

- une activité d'exécution permettant de passer d'un état à un autre en donnant toujours le même résultat pour les mêmes valeurs d'entrées (faire, calculer, transmettre...),
- une activité de décision permettant de passer d'un état à un autre après avoir pris en compte les différentes valeurs d'entrées (décider, choisir, ajuster, modifier...). L'état final n'est pas automatiquement le même pour des valeurs d'entrées similaires.

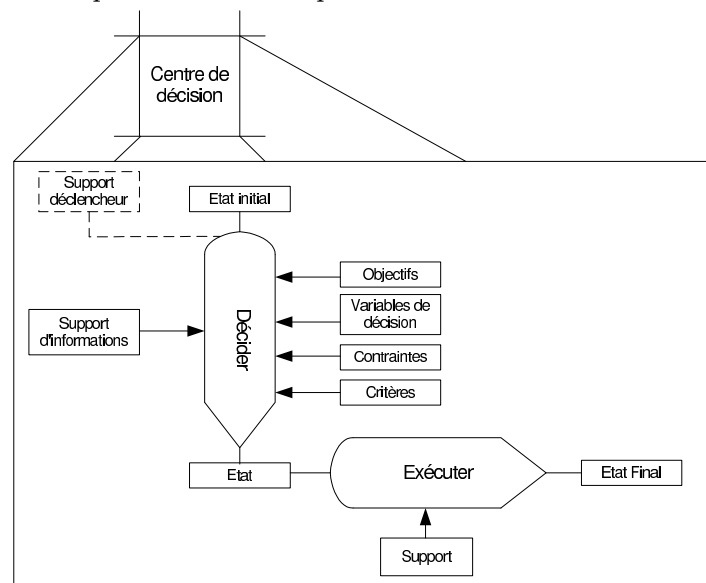


FIG. C.3 – Réseau d'activités

Nous pouvons combiner ces différents éléments afin de décrire l'enchaînement des activités d'un centre de décision. Il existe un et un seul réseau par centre de décision.

Avantages – inconvénients de la méthode

La grille GRAI est très intéressante car elle permet d'avoir une vision schématique et macroscopique du fonctionnement de l'entreprise. De plus, un ensemble de règles permet de piloter

la modélisation de ce système suivant deux principes. Le premier correspond à la conduite du groupe de pilotage dans la conception de la grille (respect du formalisme, du placement des centres de décision et de l'enchaînement des décisions...). Le deuxième correspond à des règles permettant de reconcevoir le système afin d'optimiser l'organisation de l'entreprise et obtenir un enchaînement de décision plus cohérent.

Les principaux inconvénients de cette méthode sont :

- la description des informations qui circulent entre les centres de décisions,
- un point de vue trop macroscopique de l'entreprise. Il est très difficile, avec cette méthode, d'affiner les différentes activités du réseaux.

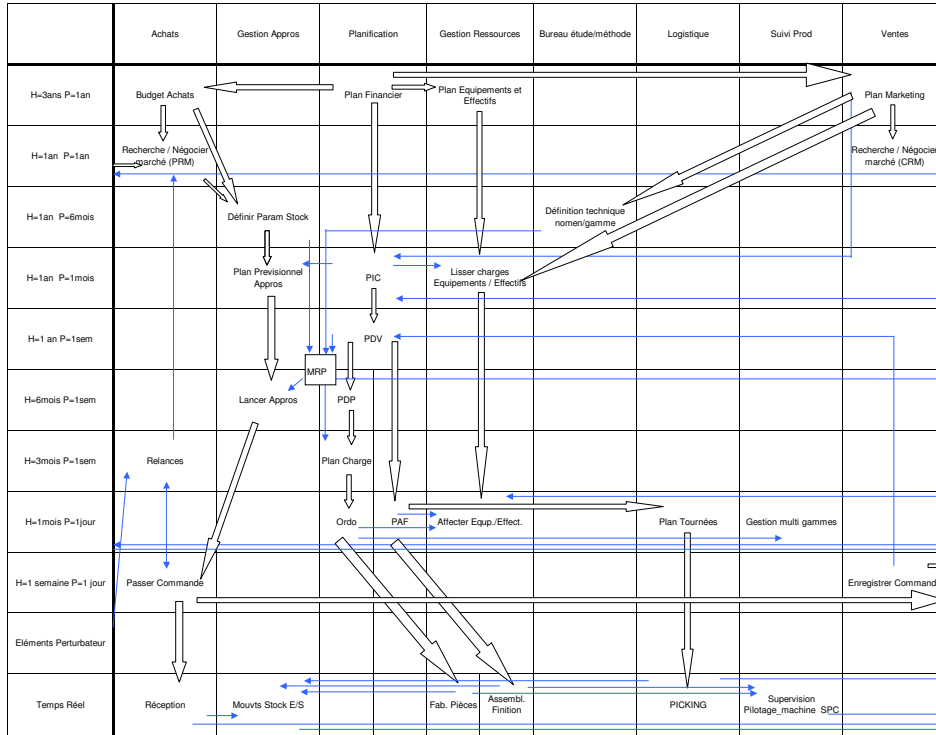


FIG. C.4 – Grille GRAI - Business Plan

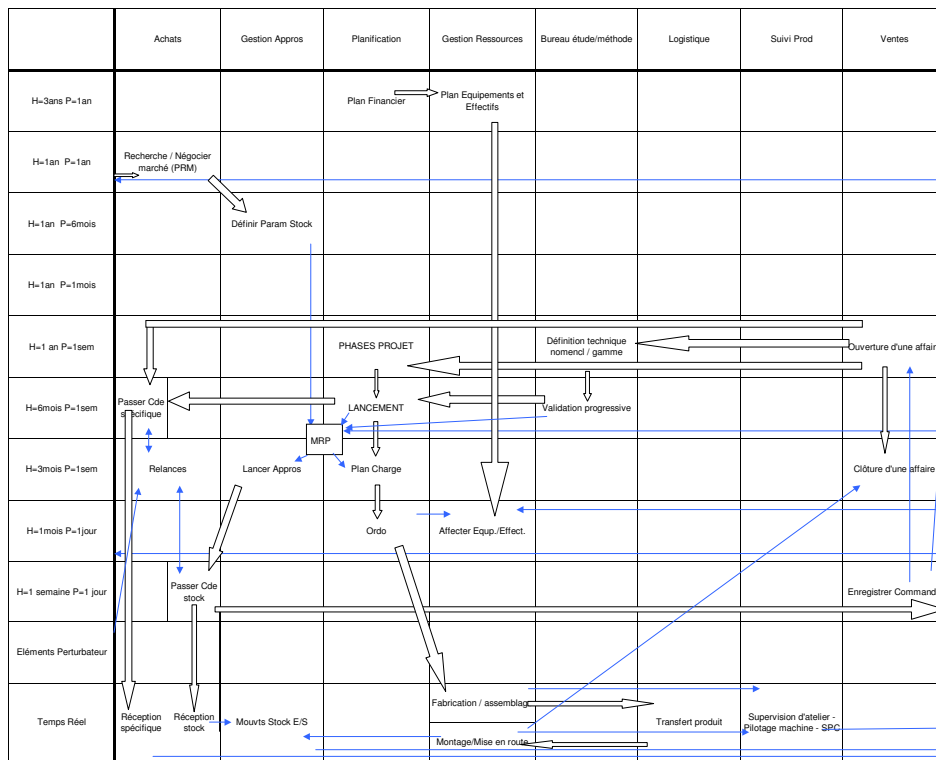


FIG. C.5 – Grille GRAI - Gestion à l'affaire

MECI

MECI est une méthode adaptée à l'analyse et la conception des systèmes complexes que sont les entreprises (Claude Pourcel et Didier Gourc, 2001).

La démarche de modélisation se décompose en trois étapes :

- identifier le système à modéliser : l'identification des processus principaux ainsi que les objets industriels primaires ;
- décomposer le système à modéliser : la décomposition des processus principaux en sous-processus ou activités, bien sûr sans oublier de définir les objets industriels liés aux processus et activités ;
- identifier et caractériser les activités d'un processus : l'identification et la spécification des activités et des objets.

Pour modéliser le système et suivre cette démarche, nous avons à notre disposition un ensemble de formalismes répartis en quatre points de vue ou aspects suivants :

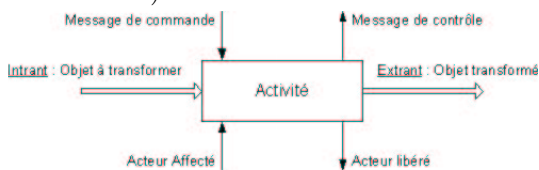
- fonctionnel,
- flux,
- décisionnel,
- organisationnel.

Aspect « fonctionnel »

L'aspect fonctionnel repose sur la mise en évidence des fonctions de l'entreprise dont les objectifs et les relations seront parfaitement identifiés. Notamment sur celles qui relatent les différentes phases du cycle de vie du produit (recherche & développement, commercial, marketing, achat, fabrication). Les éléments de liaison sont les flux d'objets tangibles (matières premières, composants) et symboliques (informations, décisions, monétaires).

Cette vision conduit à une approche métier que nous abordons par le concept d'activité. Elle sera guidée par les principes, tels que :

- la tâche définit la transformation d'un objet industriel ;
- l'exécution d'une tâche requiert une ou plusieurs compétences ;
- la ou les compétences sont fournies par un ou plusieurs acteurs ;
- l'action du ou des acteurs constitue une activité (ou succession d'opérations élémentaires).



Une activité est un ensemble d'actions destinées à modifier les caractéristiques d'un objet industriel.

FIG. C.6 – Activité vue par MECI

Aspect « flux »

Les flux constituent les éléments moteurs d'un système et sont indissociables de la notion de champ. La modulation du débit des flux s'effectue à la suite de décisions prises par les processus de conduite. Un flux doit être considéré comme un lien logique entre deux activités ou deux processus qui permet la circulation des objets industriels.

Aspect « décisionnel »

L'étude des décisions doit être orientée par deux types de considérations :

- la nature de la décision qui traduit l'influence sur les caractéristiques des objets industriels, des tâches qui fixent les conditions de leurs transformations et les compétences des acteurs qui la réalisent ;
- le niveau de décision qui caractérise l'influence plus ou moins directe sur les performances des activités de l'entreprise.

Aspect « organisationnel »

L'organisation d'un système est l'établissement et le maintien de sa structure par la création, la suppression, la modification des organes et des relations vitales entre ces organes. C'est l'environnement qui suscite l'organisation. La structure implique l'établissement de relations spatiales et dynamiques entre certains de ces organes que nous appellerons composants d'organisation.

Avantages – inconvénients de la méthode

Nous n'avons pas assez de recul pour vérifier la validité de cette méthode dans le cadre de nos travaux.

À noter cependant les différents points de vue que nous avons à notre disposition, tel que l'aspect décisionnel. Cet aspect est mal représenté dans CIMOSA et nous devons donc collaborer avec GRAI.

MECI : Méthode adaptée à l'analyse et à la conception de système complexe d'entreprise.

<p><i>Aspect « fonctionnel »</i></p> <ul style="list-style-type: none"> – la tâche et l'objet industriel : l'énoncé d'un but à atteindre et indique sur quoi porte la transformation (l'objet industriel) ; – le concept de compétence : demande de compétence formulée pour l'exécution de la tâche et une offre de la part de l'acteur ; – le concept de ressource et d'acteur ; – le concept d'activité : action destinée à modifier les caractéristiques d'un objet industriel. Le terme d'<i>opération</i> donne une vision plus détaillée que celui d'activité ; – le concept de processus : est une collection d'activités. 	<p><i>Aspect « flux »</i></p> <ul style="list-style-type: none"> – OU ? Définition de l'enchaînement des activités suivi par les objets industriels ; – COMBIEN ? Capacité et charge transportée. Ces facteurs définissent le débit sur une période ; – QUOI ? Permet de prendre en considération l'identité des objets industriels.
<p><i>Aspect « décisionnel »</i></p> <ul style="list-style-type: none"> – nature des décisions : <i>décision d'ingénierie</i> (spécifiant les objets industriels, les tâches, les compétences, les acteurs, les activités, les processus et leur organisation) et <i>décision de conduite de gestion</i> (modulant la circulation des objets industriels entre les différents processus) ; – niveau des décisions : auto-organisation (ou stratégie politique), adaptation, optimisation et régulation (ou opérationnel) ; 	<p><i>Aspect « organisationnel »</i></p> <ul style="list-style-type: none"> – Composant d'organisation : est un ensemble de ressources à utiliser sur une collection d'objets industriels (tangibles ou symboliques) à transformer (intrants) en une collection d'objets transformés (extrants) dans le cadre des objectifs et contraintes fixés par l'environnement du composant considéré.

TAB. C.4 – Description non exhaustive du vocabulaire de MECI

Référence : (Claude Pourcel et Didier Gourc, 2001)

Annexe D

MDA - Model Driven Architecture

Le rôle futur du MDA

L'approche MDA (Model-Driven Architecture) est une approche émergente dans la modélisation d'une application, d'un modèle indépendant jusqu'au code spécifique à une plate-forme. Cette approche n'est pour l'instant qu'aux premières spécifications de la norme supportée par l'OMG. Mais les premières pistes de réflexion pourront jouer un rôle futur dans la modélisation du modèle de référence de notre solution logicielle.

Dans cette section, nous présentons une courte description sur l'approche MDA.

Introduction

Le modèle MDA (Model-Driven Architecture) proposé par l'OMG (Open Management Group) propose un cadre méthodologique et architectural afin de représenter, modéliser et intégrer différents systèmes. Le MDA propose des outils et des méthodes fédérant les langages du middleware (Corba, EJB, SOAP, .NET). Ce lien est possible en utilisant un pont entre des technologies et géré par un modèle commun.

Le modèle MDA (fig. D.1) repose sur les standards supportés par l'OMG tel que UML, MOF (Meta-Object Facility) et CWM (Common Warehouse Metamodel) pour les différents langages (Corba, Web...) afin de proposer les services (gestion de transactions, gestion des événements, gestion de la sécurité...) pour les différents domaines verticaux étudiés par la *Task Force* de l'OMG (finance, service, e-commerce...).

Pour supporter tous ces échanges, l'utilisation de XML (eXtended Markup Language) est incontournable. La modélisation des processus métiers, la représentation des différentes vues informationnelles sont spécifiées en utilisant ce standard.

L'objectif du MDA est de proposer, au travers de l'architecture, en utilisant des composants pour la réalisation et intégrer différentes applications, une séparation entre le modèle et l'implémentation. Cette différenciation doit faciliter :

- la réutilisabilité des composants,
- l'interopérabilité et la portabilité des systèmes,
- la pérennisation de la conception logicielle.

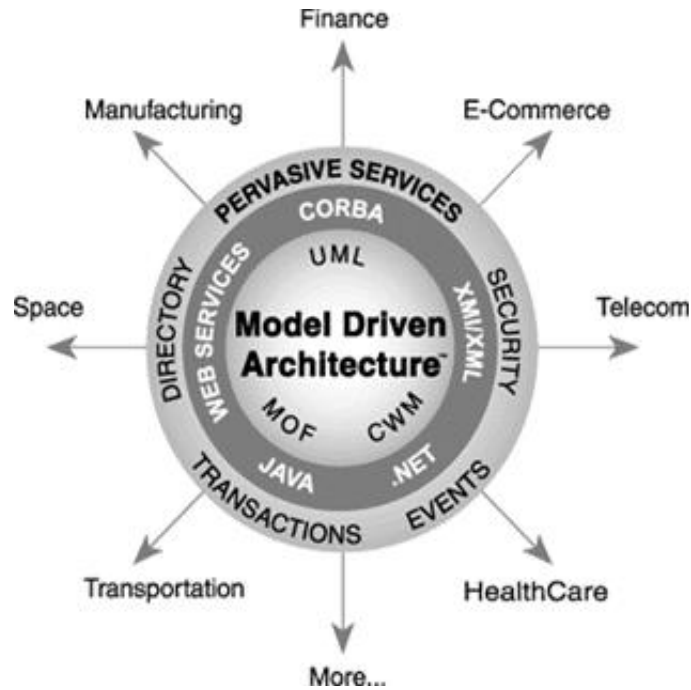


FIG. D.1 – Le modèle d’architecture MDA

Rôle et place des modèles

Afin de représenter un système, le MDA propose une séparation des différents modèles (fig. D.2). Nous retrouvons donc (Xavier Moghrabi, 2003) :

- le CIM (Computation Independent Model) : le modèle métier indépendant représente l’entreprise sans décrire le fonctionnement. Il s’agit plus précisément d’une vision de l’entreprise d’un point de vue externe ;
- le PIM (Platform Independent Model) : le modèle indépendant de la plate-forme décrit le fonctionnement du système sans en détailler l’implémentation technique. Il comprend la description de tous les éléments pérennes et stables ;
- le PSM (Platform Specific Model) : le modèle dépendant de la plate-forme sert essentiellement de base à la génération de code exécutable ;
- le PDM (Platform Description Model) : le modèle de description de la plate-forme est utilisé comme modèle de transformation pour passer du PIM au PSM spécifique à une plate-forme.

Chacun de ces modèles fournit de nouveaux éléments afin de mieux comprendre le système. Pour passer d’un modèle à un autre, différentes règles ont été spécifiées dans cette norme encore en cours de construction et de finalisation (Jean Bézin et Xavier Blanc, 2002).

Indépendance des modèles

Le MDA propose une séparation entre les différents modèles, du CIM au PIM jusqu’au PSM, tout au long du cycle de développement.

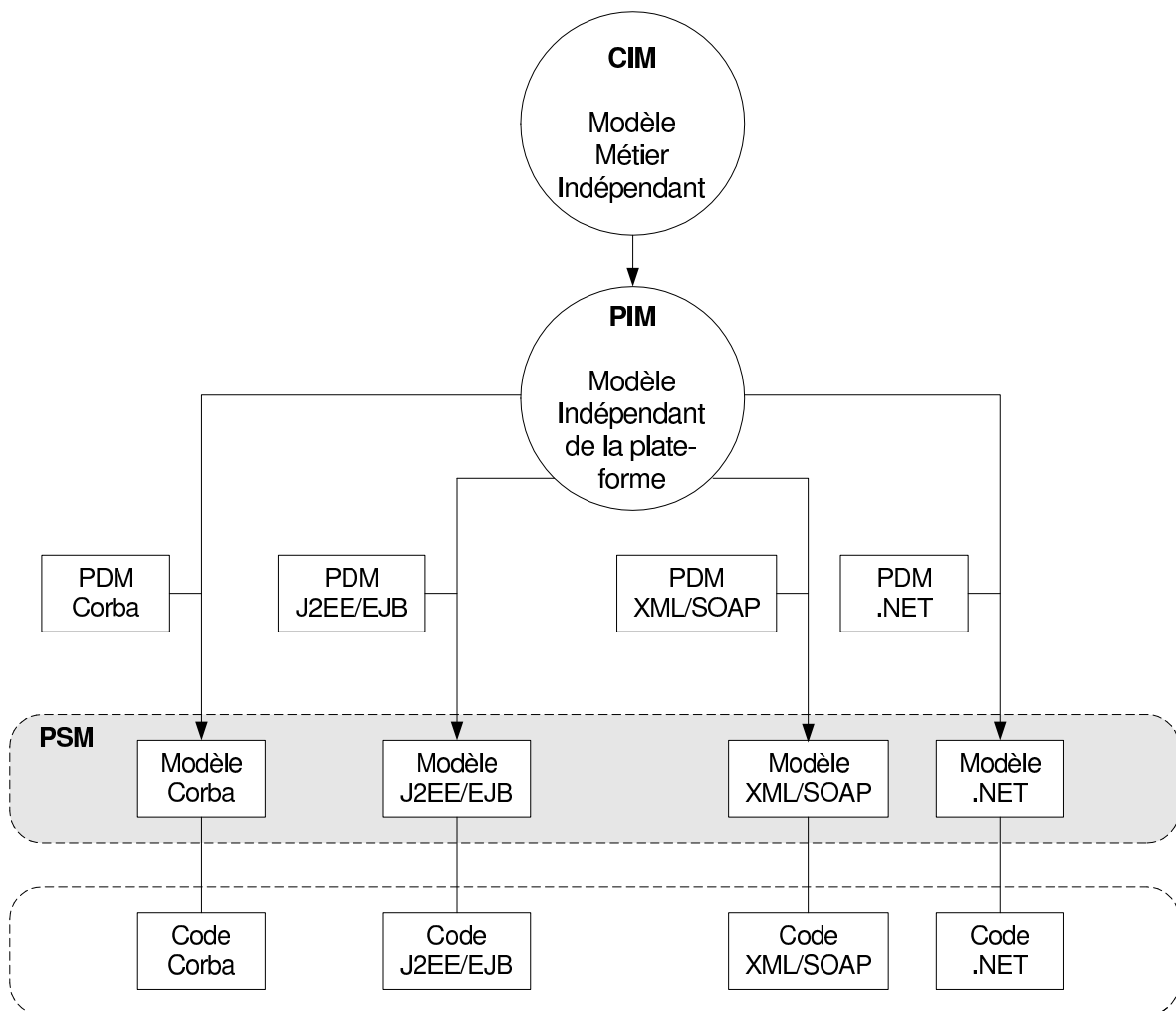


FIG. D.2 – Vision du MDA

Transformation des modèles

Différentes transformations sont nécessaires pour passer d'un modèle à un autre. Nous trouverons donc :

- passage du PIM au PIM : cette transformation est nécessaire quand nous souhaitons modifier le niveau d'abstraction du modèle durant le cycle de développement ;
- passage du PIM au PSM : cette transformation est réalisée lorsque le PIM est assez avancé pour le transformer vers un modèle spécifique à la plate-forme. Pour réaliser cette transformation, nous utilisons les informations stockées dans le PDM ;
- passage du PSM au PSM : cette transformation a pour objectif d'affiner le modèle spécifique à la plate-forme durant l'évolution du cycle de vie du modèle ;
- passage du PSM au PIM : cette transformation a pour objectif d'obtenir un modèle PIM à partir d'un système existant. Cette transformation est basée sur le principe de « reverse engineering ».

Synthèse

L'approche MDA propose des avantages indéniables tels que :

- des systèmes interopérables favorisant l'hétérogénéité des plates-formes,
- le choix de la plate-forme au dernier moment,
- un développement plus rapide et un changement de technologie moins coûteux.

Par contre, aujourd'hui, il n'existe pas encore d'outils supportant cette nouvelle méthode. Ces outils devront permettre, à terme, de bien faire la distinction entre les différents modèles et automatiser les différentes tâches de transformation.

Cette norme est en cours de finalisation. Jean Bézivin et Xavier Blanc espère obtenir une norme applicable dans les années 2005 et une clôture de la norme pour l'année 2007. Il reste donc beaucoup de points encore obscurs.

Bibliographie

- AFNET (2002). *Guide de mise en œuvre des TIC dans les PME*. Communauté de Travail PME.
- Avignon, L., Joguet, D., et Pezziardi, P. (2000). *Intégration d'applications : l'EAI au cœur du e-business*. Eyrolles.
- Berdugo, A., Mahl, R., et Jean, G. (2002). *Guide du management des systèmes d'information*. Hermès-sciences.
- Bitrant, G. R., Haas, E., et Hax, A. C. (1981). Hierarchical Production Planning : A Single Stage System. *Operations Research*, 29(4) : 717–743.
- Blumenthal, S. C. (1969). *Management information systems : a framework for planning and development*. Prentice-Hall.
- Boehm, B. W. (1981). *Software engineering economics*. Prentice Hall.
- Boehm, B. W. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4) : 22–42.
- Botta-Genoulaz, V., Millet, P.-A., et Neubert, G. (2001). The role of enterprise modeling in ERP implementation. In *Industrial Engineering and Production Management*, volume 1, pages 220–231.
- Boutin, P. (2001). *Contribution à la mise en place d'un ERP*. Thèse, École Nationale Supérieure des mines de Saint-Etienne.
- Bouzeghoub, M. et Rochfeld, A. (2000). *OOM - La conception objet des systèmes d'information*. Hermès-sciences.
- Braesch, C. (2002). Le modèle OLYMPIOS. In *Actes de l'école de la modélisation d'entreprise*, École des mines d'Albi-Carmaux.
- Breton, E. et Bézin, J. (2001). Un méta-modèle de gestion par les activités. In *CITE*, page 20, Troyes.
- Briffault, J.-P. (2000). *Systèmes d'information en gestion industrielle*. Hermès-sciences.
- Burlat, P. (2001). *GRECOPME II*. École Nationale Supérieure des mines de Saint-Etienne.
- Butler, Keith A. and Bahrami, A., Esposito, C., et Hebron, R. (2000). Conceptual models for coordinating the design of user work with the design of information systems. In *Date & Knowledge Engineering*, volume 33, pages 191–198.
- Butler, K. A., Esposito, C., et Hebron, R. (1999). Connecting the design of software to the design of work. In *Communications of the ACM*, volume 42, pages 38–46.

- Bézivin, J. (2001). From object composition to model transformation with the MDA. In *Proceedings of TOOLS, USA*, Santa Barbara.
- Bézivin, J. et Blanc, X. (2002). Promesses et interrogations de l'approche MDA. Développeur référence.
- Canonne, R. et Damret, J.-L. (2002). Essai de définition et impacts actuels des ERP : les chiffres clés. In *Propos Raisonables sur les ERP*, page 8. École des mines de Paris.
- Carel, G., Giard, V., et Midler, C. (2001). Management de projet et gestion des ressources humaines. Technical report, Cahier de recherche de l'IAE.
- Chalmeta, R., Campos, C., et Gragel, R. (2001). References architectures for enterprise integration. *Journal of Systems and Software*, 53 : 175–191.
- Chen, D. et Vernadat, F. (2001). Standardisation on enterprise modelling and integration : Achievements, on-going works and future perspectives. In *Proceeding of IFAC 10th Symposium on Information Control in Manufacturing*, Vienna. INCOM.
- Chen, P. P. (1976). The entity-relationship model - toward a unified view of data. *TODS*, 1(1) : 9–36.
- Codd, E. F. (1969). Derivability, redundancy and consistency of relations stored in large data banks. *IBM Research Report, San Jose, California*, RJ599.
- Courtois, T. (2001). La nouvelle GPAO ne fait plus cavalier seul. *CXP Informations*, (301) : 24–25.
- Darras, F., Gaborit, P., et Pingaud, H. (2002). Une analyse du besoin pour le déploiement d'un ERP en PME/PMI. In RAULT, J.-C., editor, *15^e International Conference on Software & Systems Engineering and their Applications*, volume 1, page 9. CNAM.
- Darras, F., Gaborit, P., et Pingaud, H. (2003a). Gestion de la configuration lors de l'intégration d'un ERP. In *5^{ème} Congrès International de Génie Industriel*. Université de Laval - Québec.
- Darras, F., Gaborit, P., et Pingaud, H. (2003b). La place et le rôle de la modélisation dans les projets ERP. In *4^e Conférence Francophone de MODélisation et SIMulation*, pages 128–133. LAAS - Toulouse.
- Darras, F., Gaborit, P., et Pingaud, H. (2003c). Using model of enterprise and software for deployment of ERP software. In *5th International Conference on Enterprise Information Systems*, pages 558–565. École Supérieure d'Électronique de l'Ouest - Angers.
- Davis, G. B., Olson, M. H., Ajenstat, J., et Peaucelle, J.-L. (1985). *Systèmes d'information pour le management*. Economica.
- Deixonne, J.-L. (2001). *Piloter un projet ERP*. Dunod.
- Doumeingts, G. (1984). La méthode GRAI. Thèse d'Etat, Université de Bordeaux.
- Dupont, L. (1998). *La gestion industrielle*. Hermès.
- El Amrani, R. (2003). Vision organisationnelle cible comme facteur de réussite d'un projet ERP : le cas de SAP chez l'entreprise Consto. In *De nouveaux e-usages ? Leur intégration dans les entreprises et la société*, page 8, Grenoble. 8^{ème} colloque de l'AIM.
- ENV 12204 (1996). *Advanced manufacturing technology - Systems architecture - Constructs for enterprise modelling*. CEN, Comité Européen de Normalisation.

- ENV 40003 (2001). *Enterprise Integration - Framework for Enterprise Modelling*. CEN, Comité Européen de Normalisation.
- Ernst & Young (2003). Le ROI des Systèmes d'information des PME : où en est-on aujourd'hui ?
- Fowler, M. (1997). *Analysis Patterns : Reusable objects models*. Addison Wesley.
- France-Lonard, B. (2002). *Les systèmes d'information : Art et pratiques*, chapitre Les systèmes d'information au service de la stratégie, pages 179–186. Éditions d'organisation.
- Gamma, E., Helm, R., Johnson, R., et Vlissides, J. (1999). *Design Patterns : Catalogue de modèles de conception réutilisables*. Vuibert.
- Giard, V. (1991). *Gestion de projets*. Economica.
- Gulla, J. A. et Brasethvik, T. (2002). A model-driven ERP environment with search facilities. *Data & Knowledge Engineering*, 42 : 327–341.
- Hammer, M. et Champy, J. (1993). *Le reengineering*. Dunod.
- Industrial Engineering (2001). *Handbook of Industrial Engineering : Technology and Operations Management*, 3^{ème} édition.
- Jean, G. (2000). *Urbanisation du business et des SI*. Hermès.
- Kadima, H. (2003). *Les Web services - Techniques, démarches et outils*. Dunod - 01 Informatiques.
- Kay, A. et Goldberg, A. (1976). Smalltalk-72 instruction manual. Technical Report SSL-76-6, Xerox Parc, Palo Alto, Californie.
- Kettani, N., Mignet, D., Paré, P., et Rosenthal-Sabroux, C. (1999). *De MERISE à UML*. Eyrolles.
- Kruchten, P. (1995). The 4+1 view model of architecture. *IEEE Software*.
- Kumar, K. et Hillegersberg, J. v. (2000). Enterprise resource planning : Experiences and evolution. *Communications of the ACM*, 43(4) : 22–26.
- Laur, B. (2001). Six progiciels de gestion intégrés pour PME. *Décision Micro & Réseaux*, (487) : 34–40.
- Lauras, M., Lamothe, J., et Pingaud, H. (2003). Un projet de modélisation de la dynamique de coopération au sein d'une chaîne logistique pharmaceutique et cosmétique. In *4^e Conférence Francophone de MODélisation et SIMulation*. LAAS - Toulouse.
- Le Moigne, J.-L. (1973). *Les systèmes d'information dans les organisations*. Systèmes-décisions. Presses Universitaires de France.
- Le Moigne, J.-L. (1977). *La théorie du système général*. Presses Universitaires de France.
- Le Moigne, J.-L. (1992). *La modélisation des systèmes complexes*. Dunod.
- Lequeux, J.-L. (1999). *Manager avec les ERP : Progiciels de gestion intégrés et Internet*. Éditions d'organisation.
- Levi, M. H. et Klapsis, M. P. (1999). FirstSTEP process modeler - a CIMOSA-compliant modeling tool. *Computers in Industry*, 40 : 267–277.

- Light, B. (2001). The maintenance implications of the customisation of ERP software. *Journal of software maintenance*.
- Lissandre, M. (1990). *Maîtriser SADT*. Colin.
- Maaloul, I. et Mezghani, L. (2003). L'implantation des ERP et ingénierie du changement : Les déterminants de la satisfaction des utilisateurs d'un ERP. In *De nouveaux e-usages ? Leur intégration dans les entreprises et la société*, page 13, Grenoble. 8^{ème} colloque de l'AIM.
- Marcinak, R. et Rowe, F. (1997). *Systèmes d'information, dynamique et organisation*. PIQ. Economica.
- Marshall, C. (1999). *Enterprise modeling with UML : Designing successful software through business analysis*. Addison-Wesley.
- Mercé, C. (1987). *Cohérence des Décisions en Planification Hiérarchisée*. Thèse, Université Paul Sabatier, Toulouse.
- Meyer, B. (1988). *Object-oriented software construction*. Prentice Hall.
- Mintzberg, H. (1980). Structure in 5's : A synthesis of the research on organizational design. *Management Science*, 26(3) : 322–341.
- Moghrabi, X. (2003). L'approche model-driven architecture, crédible pour développer un progiciel de gestion intégré.
- Morley, C. (2002). La modélisation des processus : typologie et proposition utilisant uml. In *Processus & Systèmes d'information- Journée ADELI*, page 13. CNAM.
- Morley, C., Hugues, J., et Leblanc, B. (2000). *UML pour l'analyse d'un système d'information*. Dunod.
- Muller, P. A. (1999). *Modélisation objet avec UML*. Eyrolles.
- Mélèse, J. (1990). *Approches systémiques des organisations*. Éditions d'organisations.
- Nanci, D. et Espinasse, B. (1996). *Ingénierie des systèmes d'information : MERISE*. Sybex.
- Naumenko, A. et Wegmann, A. (2003). Two approaches in system modeling and their illustrations with MDA and RM-ODP. In *5th International Conference on Enterprise Information Systems*, pages 398–402. École Supérieure d'Électronique de l'Ouest - Angers.
- Oussalah, C. (1997). *Ingénierie objet : Concepts et techniques*. InterEditions.
- Peaucelle, J.-L. (1999). *Systèmes d'Information : Le point de vue des gestionnaires*. Economica.
- Pingaud, H. (2003). *Solutions pratiques : Logistiques & Supply Chain*, chapitre Quels sont les outils et systèmes d'information de la logistique et de la supply chain, page 36. Édition WEKA.
- Pourcel, C. et Gourc, D. (2001). Démarche de modélisation en entreprise : Présentation de MECI.
- Reix, R. (2002). *Système d'information et management des organisations*. Vuibert, 4^{ème} édition.
- Riccio, P.-M. (1999). Une méthodologie unifiée pour la construction de systèmes de qualité dans le domaine des TIC.

- Roboam, M. (1993). *La méthode GRAI : Principes, outils, démarches et pratique*. Teknea.
- Roques, P. et Vallée, F. (2003). *UML en action*. Technologies objet. Eyrolles, deuxième édition.
- Rosnay, J. d. (1975). *Le microscope : vers une vision globale*. Points essais. Éditions du Seuil.
- Saadoun, M. (2000). *Technologies de l'information et management*. Hermès.
- SAP (1999). *SAP R/3*. Le Macmillan, campus press édition.
- Scheer, A.-W. (1999). *ARIS - Business Process Modeling*. Springer.
- Schneider, M. (2001). *Interopérabilité et intégration des systèmes d'information*, volume 6 of *Ingénierie des systèmes d'information*. Hermès.
- Simon, H. A. (1947). *Administrative Behavior - A Study of Decision-making Processes in Administrative Organization*. Macmillan Company.
- Sprott, D. (2000). Componentizing the enterprise application packages. In *Communications of the ACM*, volume 43, pages 63–69.
- Tardieu, H., Rochfeld, A., et Colletti, R. (1998). *La méthode MERISE : Principes et Outils*. Éditions d'organisations.
- Theroude, F., Braesch, C., et Haurat, A. (2001). OLYMPIOS : Un modèle pour le pilotage de processus. In *Conception, Analyse et gestion des Systèmes Industriels*, pages 249–253. 3^{ème} Conférence Francophone de MODélisation et SIMulation - MOSIM'01.
- Tomas, J.-L. (1999). *ERP et progiciels intégrés : La mutation des systèmes d'information*. InterEditions.
- Tomas, J.-L. (2002). *ERP et progiciels de gestion intégrés : sélection, déploiement et utilisation opérationnelle, les bases du SCM et du CRM*. Dunod.
- Tysebaert, J.-M. (2001). *Les logiciels de gestion hautement intégrés : Préparation par l'ingénierie de métier*. Éditions TECHNIP.
- Vernadat, F. (1996). *Enterprise modeling and integration : Principles and Applications*. Chapman & Hall.
- Vernadat, F. (1999). *Techniques de modélisation en entreprise : Applications aux processus opérationnels*. Economica.
- Vernadat, F. (2001). UEML : Towards a Unified Enterprise Modelling Language. In *Conception, Analyse et gestion des Systèmes Industriels*, pages 3–10. 3^{ème} Conférence Francophone de MODélisation et SIMulation - MOSIM'01.
- Zachman, J. A. (1987). A framework for information systems architecture. In March, S. T., editor, *Entity-Relationship Approach, Proceedings of the Sixth International Conference on Entity-Relationship Approach, New York, USA*, volume 26, pages 276–292. North-Holland.

Proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré

Les travaux présentés dans cette thèse portent sur la proposition d'un cadre de référence afin d'accompagner la gestion de projet lors de la conception et le déploiement d'un ERP pour les PME/PMI. Ces travaux se placent à l'intersection des domaines du génie industriel et du génie logiciel.

Les enjeux et les risques dans la mise en œuvre d'un système d'information sont importants. Afin d'améliorer la qualité des projets suivant des notions de coûts, de délais et de performance, nous apportons de nouveaux éléments dans la conception et le pilotage de ces projets. Nous proposons une méthode d'ingénierie des projets ERP basée sur l'utilisation des modèles d'entreprise et de référence.

Dans une première partie, nous travaillons sur l'utilisation d'un formalisme unique, le langage UML (Unified Modeling Language) pour modéliser l'organisation de l'entreprise et le fonctionnement de la solution logicielle. Le langage UML permet de représenter les points de vue physique, logique ou fonctionnel de ces systèmes. De plus, le langage UML permet, au gré de l'utilisateur, d'avoir différents niveaux de granularité.

Dans une deuxième partie, nous travaillons sur la définition d'un cadre de référence pour la gestion des projets ERP. Dans ce cadre de référence, nous avons placé les modèles d'entreprise et de référence. Nous avons défini des mécanismes pour modifier les modèles et les mettre en correspondance afin d'obtenir le « bon » paramétrage de la solution logicielle. Un paramétrage qui correspond à la juste adéquation de l'organisation et du système d'information.

Mots-clés : modèle d'entreprise, modèle de référence, UML, ERP, PME/PMI

Proposal of a formal framework for the design and use of an Enterprise Resource Planning software.

This research work proposes a formal framework to manage the project of design and exploitation of an ERP in SMEs/SMIs. The study covers two scientific disciplines : industrial engineering and software engineering.

Stakes and risks in the implementation of an information system are substantial. In order to improve the quality of projects in terms of cost, time and performance, we contribute to the progress of knowledge with a set of new elements which facilitate project design and management. We propose a method of engineering of these projects based on the use of enterprise and reference models.

First, we work on the use of a single formalism, the UML language (Unified Modeling Language), to model the company organization and software functionalities. The UML language makes it possible to represent the physical, logical or functional points of view of these systems. Moreover, the UML language allows the user to have various levels of granularity with the representation, with respect to the user's expectations.

Second, we deal with the definition of a formal framework for ERP project management. Within this reference framework, we try to allocate enterprise and reference models. We defined mechanisms to modify the models and to integrate them in order to obtain the right parameter setting for the software solution. The customization corresponds to the right adequacy between the organization and the information system.

A case study is used in order to demonstrate the relevance of our approach for the last two parts of the thesis.

Keywords : enterprise model, reference model, UML, ERP systems, SMEs/SMIs